

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

На правах рукопису

ПУХАЛЕВИЧ АНДРІЙ ВОЛОДИМИРОВИЧ

УДК 004.942:519.25

**МОДЕЛІ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ
ПЕРЕРОБКИ ІНФОРМАЦІЇ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ
ПРОЕКТІВ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

05.13.06 – Інформаційні технології

Дисертація

на здобуття наукового ступеня кандидата технічних наук

Науковий керівник
д.т.н., проф. Приходько С.Б.

Миколаїв – 2017

ЗМІСТ

Перелік умовних позначень	4
Вступ	5
Розділ 1 Сучасний стан інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.	
Постановка завдань дослідження	13
1.1 Аналіз стану інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.....	13
1.2 Дослідження існуючих моделей та методів оцінювання тривалості проектів з розробки програмного забезпечення	16
1.3 Аналіз шляхів вдосконалення моделей тривалості проектів з розробки програмного забезпечення та постановка завдань дослідження.....	26
1.4 Висновки до розділу 1	30
Розділ 2 Побудова негаусівських ймовірнісних моделей тривалості проектів з розробки програмного забезпечення на основі нормалізуючих перетворень	32
2.1 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення.....	33
2.2 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи РС.....	42
2.3 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи mid-range	45
2.4 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи mainframe	48
2.5 Висновки до розділу 2.....	51
Розділ 3 Побудова нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення	53

3.1	Аналіз емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення для побудови нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення.....	54
3.2	Нормалізація випадкових величини із використанням нормалізуючих перетворень.....	56
3.3	Вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення.....	58
3.4	Вибір нормалізуючого перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення	79
3.5	Побудова нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення.....	100
3.6	Висновки до розділу 3.....	121
Розділ 4	Інформаційна технологія переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення	123
4.1	Автоматизована інформаційна система переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення....	124
4.2	Створення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення....	129
4.3	Методика статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.....	147
4.4	Висновки до розділу 4.....	151
Висновки		152
Список використаних джерел		155
Додатки		168
Додаток А	Акти впровадження результатів дисертаційної роботи	169
Додаток Б	Настанова користувача комп'ютерної програми «Оцінювання тривалості розробки програмного забезпечення»	173

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	Application programming interface
COCOMO	Constructive cost model
JRE	Java runtime environment
ISBSG	International software benchmarking standards group
MF	Main frame
MR	Mid-range
PC	Personal computer
PERT	Program evaluation and review technique
SLIM	Software life-cycle model
UML	Unified modeling language
БД	База даних
ОЗП	Оперативний запам'ятовувальний пристрій
ПК	Персональний комп'ютер
ПЗП	Постійний запам'ятовувальний пристрій
СУБД	Система управління базами даних

ВСТУП

Актуальність теми. Процес оцінювання тривалості робіт є одним з основним факторів, які впливають на успішність виконання проектів з розробки програмного забезпечення [1]. Достовірне оцінювання тривалості проектів з розробки програмного забезпечення дозволяє успішно їх завершувати, тоді як порушення запланованого розкладу проекту приводе до фінансових втрат з боку замовника та з боку розробника цих проектів.

Для проектів з розробки програмного забезпечення оцінювання тривалості завжди було одним із найризикованіших аспектів планування. Така ситуація склалася через постійно зростаючу кількість елементів і залежностей, які повинні бути враховані при оцінюванні тривалості проектів з розробки програмного забезпечення [2]. Сучасні програми є надзвичайно складними, вони характеризуються наявністю великої множини компонентів, паралельного виконання, розподілених ресурсів та інших не менш важливих аспектів складності [3], що значно ускладнює оцінювання тривалості проектів розробки програмного забезпечення.

Дані багатьох вчених та розробників показують, що більшість (71% в 2015 році) проектів з розробки програмного забезпечення не виконуються в заданий термін [3-7]. Тобто, існує проблема низької достовірності оцінювання тривалості таких проектів, що призводить до фінансових втрат. Тому не припиняються дослідження з побудови та вдосконалення моделей тривалості проектів з розробки програмного забезпечення (в тому числі побудова математичних моделей з використанням методів теорії ймовірностей, математичної статистики), та зі створення інформаційних технологій переробки інформації для оцінювання тривалості цих проектів.

Теоретичну та практичну базу в галузі інформаційних технологій складають твори таких вчених як В. М. Глушкова, С. А. Лебедева,

Е. Л. Ющенко, З. Л. Рабіновича, Ю. В. Капітонова, А. А. Летичевского, В. Є. Ходакова, О. А. Павлова, Б. Боема, Т. Бернерс-Лі, Г. Мура, В. Серфа та інших.

Значний внесок у розвиток теорії ймовірностей, математичної статистики та статистичної обробки даних внесли такі вчені як А. М. Колмогоров, Л. М. Большев, С. В. Фомін, Ю. М. Тюрін, М. В. Смирнов, О. І. Орлов, Ю. М. Кофанов, С. А. Айвазян, І. М. Коваленко, В. А. Бостанджиян, А. Стьюарт, М. Дж. Кендалл, Г. Хан, Р. Фішер, В. С. Госсет, С. Шапіро, А. Вальд, Дж. Нейман, Н. Дрейпер, К. Пірсон, Х. Крамер, Н. Л. Джонсон та інші.

Оцінюванню тривалості проектів з розробки програмного забезпечення присвячено багато наукових праць та досліджень [1-46 та інші]. Діапазон методів оцінювання тривалості проектів з розробки програмного забезпечення весь час розширюється, були побудовані моделі багатьох видів. Існують різні методи та експертні системи для оцінювання тривалості проектів з розробки програмного забезпечення. За [8] основними інструментами і методами оцінювання тривалості робіт є експертне оцінювання, оцінювання за аналогами, параметричне оцінювання (нелінійні регресійні моделі COCOMO [9], ISBSG [10, 11]), оцінювання за трьома точками (метод PERT [8, 12], що базується на такій ймовірнісній моделі як бета розподіл).

Вказані моделі, які використовуються при створенні сучасних інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення, не завжди добре враховують реальний розподіл емпіричних даних, що знижує достовірність оцінювання тривалості цих проектів. Закон розподілу емпіричних даних тривалості проектів з розробки програмного забезпечення та трудомісткості цих проектів відрізняється від нормального закону розподілу [9, 10, 47-56]. Тому, як правило, вказані моделі не дозволяють впевнено виконувати оцінювання довірчих інтервалів та інтервалів прогнозування тривалості без застосування припущення про нормальність закону розподілу емпіричних даних [57, 58].

Для підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення необхідно побудувати по перше ймовірнісну модель, яка буде більш адекватно апроксимувати розподіл емпіричних даних тривалості робіт в цих проектах; по друге, нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів із використанням іншого нормалізуючого перетворення, яке дозволить звужити довірчий інтервал та інтервал прогнозування нелінійної регресії в порівнянні з перетворенням у вигляді десяткового логарифму, яке використовується при побудові моделей COCOMO та ISBSG.

Таким чином, побудова відповідних негаусівських ймовірнісних моделей та нелінійних регресійних моделей із використанням нормалізуючих перетворень [59, 60] для підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення, і створення на їх основі інформаційної технології переробки інформації є актуальною та має практичну цінність.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційну роботу виконано на кафедрі програмного забезпечення автоматизованих систем Національного університету кораблебудування ім. адмірала Макарова відповідно до планів НДР з ініціативних тем «Оцінювання тривалості програмних проектів на основі негаусовських стохастичних моделей» [52] (номер державної реєстрації 0111U005719), «Побудова нелінійної регресійної моделі тривалості робіт на основі нормалізуючих перетворень для управління часом в програмних проектах» [53] (номер державної реєстрації 0113U0000199). Здобувач був відповідальним виконавцем вказаних науково-дослідних робіт.

Мета і задачі дослідження.

Мета дисертаційної роботи полягає у вирішенні науково-практичного завдання підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення.

Для досягнення поставленої мети в роботі необхідно вирішити наступні задачі:

1. Провести аналіз сучасного стану інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення. Дослідити існуючі моделі та методи оцінювання тривалості проектів з розробки програмного забезпечення, визначити переваги та недоліки цих моделей та методів.
2. Побудувати ймовірнісні моделі тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe.
3. Вибрати нормалізуючі перетворення для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe та побудувати нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі вибраних нормалізуючих перетворень для вказаних платформ.
4. Побудувати рівняння нижньої та верхньої границь довірчих інтервалів нелінійних регресій тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі вибраних нормалізуючих перетворень для платформ PC, mid-range, mainframe.
5. Побудувати рівняння нижньої та верхньої границь інтервалів прогнозування нелінійних регресій тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі вибраних нормалізуючих перетворень для платформ PC, mid-range, mainframe.
6. Розробити архітектуру автоматизованої інформаційної системи та створити інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.
7. Створити методику статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.

Об'єктом дослідження є процес оцінювання тривалості проектів з розробки програмного забезпечення.

Предметом дослідження є математичні моделі тривалості проектів з розробки програмного забезпечення та інформаційна технологія переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.

Методи дослідження. При проведенні досліджень використовувалися методи теорії ймовірностей, математичної статистики, математичного моделювання, інтервального аналізу, регресійного аналізу, об'єктно-орієнтованого програмування.

Методи теорії ймовірностей та математичної статистики застосовані для аналізу та виявлення існуючих закономірностей у емпіричних даних трудомісткості та тривалості проектів з розробки програмного забезпечення. Методи математичного моделювання застосовані для побудови аналітичних моделей законів розподілу вказаних емпіричних даних та для моделювання значень тривалості проектів з розробки програмного забезпечення. Методи регресійного аналізу застосовані для побудови регресійних моделей тривалості проектів з розробки програмного забезпечення. Методи інтервального аналізу застосовані для побудови довірчих інтервалів вказаних регресійних моделей. Методи об'єктно-орієнтованого програмування застосовані для створення програмного забезпечення.

Наукова новизна одержаних результатів.

– *удосконалено* негаусівську ймовірнісну модель тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe *за рахунок* застосування щільності ймовірності Джонсона, *що дозволяє* підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення;

– *отримали подальший розвиток* нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів (моделі ISBSG) для платформ PC, mid-range, mainframe *за рахунок*

застосування нормалізуючого перетворення Джонсона, *що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення;*

– *вперше* побудовано рівняння нижньої та верхньої границь довірчих інтервалів нелінійних регресій тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe *за рахунок* застосування нормалізуючого перетворення Джонсона, *що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення;*

– *вперше* побудовано рівняння нижньої та верхньої границь інтервалів прогнозування нелінійних регресій тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe *за рахунок* застосування нормалізуючого перетворення Джонсона, *що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення.*

Практичне значення одержаних результатів. Результати дисертаційної роботи, висновки та рекомендації, що містяться у роботі, схвалені та використовуються в наступних установах: ТОВ «Макротел» (акт впровадження від 21.05.2016), ТОВ «Вебкодерс» (акт впровадження від 07.11.2016). Завдяки застосуванню інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення створеної з використанням розроблених математичних моделей, які враховують реальний характер розподілу емпіричних даних, підвищилась достовірність оцінювання тривалості проектів з розробки програмного забезпечення. Створені відповідні програмно-інформаційні засоби (інженерна методика та програмне забезпечення) дозволили підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення.

Результати дисертаційної роботи також застосовуються в навчальному процесі на кафедрі програмного забезпечення автоматизованих систем Національного університету кораблебудування ім. адмірала Макарова (акт впровадження від 02.11.2016).

Особистий внесок здобувача. Дисертаційна робота є самостійно виконаною науковою працею. Усі наукові результати отримано здобувачем особисто. У працях, опублікованих у співавторстві, здобувачеві належать: побудова негаусівських ймовірнісних моделей тривалості проектів з розробки програмного забезпечення за рахунок використання нормалізуючого перетворення Джонсона, що дозволяє краще враховувати реальний закон розподілу емпіричних даних тривалості проектів з розробки програмного забезпечення та виконувати оцінювання довірчих інтервалів математичного сподівання тривалості робіт в цих проектах [47-52, 78, 79]; побудова нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range, mainframe за рахунок використання нормалізуючого перетворення Джонсона, що дозволяє підвищити достовірність оцінювання тривалості завдяки покращенню апроксимації закону розподілу емпіричних даних трудомісткості і емпіричних даних тривалості проектів з розробки програмного забезпечення, та завдяки побудові довірчих інтервалів нелінійної регресії і інтервалів прогнозування нелінійної регресії [53-56, 80-84]; створення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення із застосуванням моделей, побудованих з використанням нормалізуючого перетворення Джонсона [61, 85] ; проектування модулю оцінювання та моделювання тривалості, розробка і тестування програми [62].

Апробація результатів дисертації. Основні положення й результати дисертаційної роботи доповідалися на 17 конференціях та семінарах:

- XII Науково-практичній конференції «Інформаційні технології в освіті та управлінні» (м. Нова Каховка, 2010);
- VI, VII Міжнародній науково-практичній конференції «Управління проектами: стан та перспективи» (м. Миколаїв, 2010, 2011);
- Форумі молодих науковців НУК «Макаровські читання» (м. Миколаїв, 2011, 2012, 2013, 2014);

– IX, X Міжнародній науково-практичній конференції «Сучасні інформаційні технології в економіці та управлінні підприємствами, програмами та проектами» (м. Алушта, 2011, 2012);

– Міжнародній науково-практичній конференції «Комп'ютерні науки: освіта, наука, практика» (м. Миколаїв, 2012);

– Міжнародній науково-практичній конференції «Математичне моделювання процесів в економіці та управлінні інноваційними проектами» (м. Алушта, 2013);

– Науковому семінарі кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування ім. адмірала Макарова (м. Миколаїв, 2013);

– II Науково-практичному семінарі "Управління проектами: наука, практика, освіта" (м. Миколаїв, 2013);

– XII Міжнародній науково-технічній конференції «Сучасні проблеми радіоелектроніки, телекомунікацій та комп'ютерної інженерії» (с.м.т. Славське, 2014);

– Міжнародній науковій конференції студентів, аспірантів та молодих вчених «Теоретичні та прикладні аспекти кібернетики» (м. Київ, 2014);

– XVIII науково-практичній міжнародній конференції «Інформаційні технології в освіті та управлінні» (м. Херсон, 2016);

– Всеукраїнській науково-практичній конференції «Прикладна геометрія та інформаційні технології в моделюванні об'єктів, явищ і процесів» (м. Миколаїв, 2016).

Публікації. За матеріалами дисертації опубліковано 18 наукових праць, з них 9 статей у фахових наукових виданнях України, 7 з яких – у виданнях, які включені до міжнародних наукометричних баз ([48-51, 54] – у Google Академія; [55] – у CrossRef Metadata Search, РІНЦ, Google Академія, [56] – у Index Copernicus, Google Академія); 8 тез у збірниках праць вітчизняних та міжнародних конференцій; одне свідоцтво про реєстрацію авторського права [62].

РОЗДІЛ 1

СУЧАСНИЙ СТАН ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПЕРЕРОБКИ ІНФОРМАЦІЇ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ
ПРОЕКТІВ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.
ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Аналіз стану інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

Результати досліджень в галузі створення програмного забезпечення показують, що при використанні існуючих інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення більшість проектів з розробки програмного забезпечення не виконуються в заданий термін [3-7], що приведе до фінансових втрат з боку замовника та з боку розробника цих проектів (таблиця 1.1 [4]).

Таблиця 1.1 – Дані по успішності виконання проектів з розробки програмного забезпечення [4]

	1996	1998	2000	2002	2004	2006	2009	2011	2012	2013	2014	2015
Успішні	27%	26%	28%	34%	29%	35%	32%	29%	27%	31%	28%	29%
Затримки	33%	46%	49%	51%	53%	46%	44%	49%	56%	50%	55%	52%
Невдалі	40%	28%	23%	15%	18%	19%	24%	22%	17%	19%	17%	19%

Одним з основних факторів, який впливає на успішність виконання проектів з розробки програмного забезпечення та дозволяє вчасно завершувати

такі проекти, є достовірність оцінювання тривалості робіт при управлінні часом проекту.

Оцінювання тривалості робіт – це процес приблизного визначення кількості робочих періодів, необхідних для завершення окремих робіт при передбачуваних ресурсах [8]. Для проектів з розробки програмного забезпечення оцінювання тривалості робіт завжди було одним із найскладніших та найризикованіших аспектів планування. Така ситуація склалася через постійно зростаючу кількість елементів і залежностей, які повинні бути враховані при оцінюванні тривалості проектів з розробки програмного забезпечення [2].

Процес оцінювання тривалості робіт є частиною управління часом проекту. Управління часом проекту включає в себе процеси, що забезпечують своєчасне завершення проекту [8], в тому числі і оцінювання тривалості робіт. Загальна схема процесів управління часом показана на рис. 1.1 [8].



Рис. 1.1. Процеси управління часом проектів [8]

Ці процеси взаємопов'язані один з одним, кожен процес відбувається в кожному проекті щонайменше один раз і виконується в одній або декількох фазах проекту, якщо проект розбито на фази. Вихідні дані, отримані в процесі оцінювання тривалості робіт проекту є вхідними даними для іншого процесу управління часом – розробки розкладу проекту [8].

У якісної оцінки тривалості робіт при управлінні часом в проектах з розробки програмного забезпечення, повинні бути присутні наступні ознаки [3]:

- оцінка є зрозумілою і підтриманою менеджером проекту та командою розробників;
- оцінку схвалено всіма зацікавленими особами як реально можливою до виконання;
- оцінка оснований на чіткій моделі з основою, яка заслуговує довіри;
- оцінка оснований на базі даних подібного проекту (з подібними бізнес-процесами, подібними технологіями, подібним зовнішнім середовищем, подібними людьми і подібними вимогами);
- оцінка визначена наскільки детально, що ключові області ризику є зрозумілими і ймовірність успіху була об'єктивно оцінена.

Виконання якісного оцінювання історично було одним з найбільш складних завдань при розробці програмного забезпечення. Декілька причин цих складнощів:

- нестача історично сформованої бази оцінок;
- розробка програмного забезпечення оснований на множині взаємозв'язаних факторів, які прямо впливають на продуктивність і зв'язки яких повністю не виявлені.

Програмне забезпечення проектів, програмних продуктів та ІТ-середовища стає дедалі все більш складним, що ускладнює завдання оцінювання тривалості проектів з розробки програмного забезпечення. Таким чином, пошук моделей, які дозволять підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення, та розробка на їх основі інформаційної технології переробки інформації є актуальною та має практичну цінність.

1.2 Дослідження існуючих моделей та методів оцінювання тривалості проектів з розробки програмного забезпечення

Оцінюванню тривалості проектів з розробки програмного забезпечення присвячено багато наукових праць та досліджень. Діапазон методів оцінювання тривалості проектів з розробки програмного забезпечення весь час розширюється, були побудовані моделі різних видів. Існують різні методи та експертні системи для визначення тривалості проектів з розробки програмного забезпечення.

Основними інструментами і методами оцінювання тривалості робіт, які використовуються в сучасних інформаційних технологіях переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення є [8]:

1. Експертне оцінювання.
2. Оцінювання за аналогами.
3. Параметричне оцінювання.
4. Оцінювання за трьома точками (метод PERT).

Крім наведених методів існують також методи, які враховують особливості проектів з розробки програмного забезпечення. На рис. 1.2 представлено класифікацію існуючих моделей [2].

Розглянемо ці моделі та методи більш детально.

Методи, що базуються на експертному оцінюванні

Методи, що базуються на експертному оцінюванні, основані на суб'єктивному рішенні людини-експерта або групи експертів, є найбільш широко використовуваними методами для оцінювання тривалості проектів з розробки програмного забезпечення [13, 17]. На відміну від параметричних методів оцінювання, експертні методи спираються на інтуїцію і досвід, накопичений людиною-експертом [18]. Наприклад, у відповідності до методу експертного оцінювання Delphi (створеному в Rand Corporation в 1966р), на початковому етапі кожний член групи експертів самостійно оцінює тривалість

робіт, без обговорення з іншими членами групи. Після цього відбувається збір та сортування оцінок, отриманих на початковому етапі. На другому етапі проходить обговорення та уточнення отриманих оцінок. Після кожного етапу оцінювання, деякі елементи і деталі будуть враховані, в той час як інші будуть відкинуті. Цей процес ітеративно повторюється, доки не буде досягнуто спільного узгодження щодо тривалості проекту [13, 19].



Рис. 1.2. Класифікація методів для оцінки тривалості проектів з розробки програмного забезпечення [2]

Запропонована Боемом модифікація методу Delphi, що використовується для оцінювання проектів з розробки програмного забезпечення (називається Wideband Delphi [9, 19, 20]) відрізняється від оригінального методу Delphi прискореною процедурою вироблення рішення. Модифікація методу Delphi показала себе більш ефективною: попередньо проводиться нарада координатора і експертів, на якій обговорюються проблеми оцінювання, а після оцінювання експерти не дають ніяких пояснень своїх оцінок. Замість цього після кожного оцінювання координатор проводить зібрання, на якому обговорюються деталі оцінок, по яким були виявлені найбільші розбіжності [21].

Методи, що базуються на експертній оцінці мають свої недоліки, серед яких [21]:

- підстави для отримання оцінок не є явними;
- складно знаходити висококваліфікованих експертів для кожного нового проекту;
- зв'язок між розміром системи і оцінюваними характеристиками – нелінійний [9]. Тому адекватна експертна оцінка може бути отримана лише в тому випадку, коли поточний проект і попередні є приблизно однакового розміру;
- політика керівників, направлена на скорочення витрат, як правило, ставить під сумнів реальний досвід попередніх проектів і привносить частку «сліпого оптимізму» [3].

Незважаючи на те що експертні оцінки можуть приводити до гарних результатів при відносно невисоких витратах, їхнє застосування лежить більше в інтуїтивній, ніж у науковій площині, а тому не завжди сприяє підвищенню прозорості процесу прийняття рішень. Для того щоб охопити динамічні взаємодії навіть в рамках простих програмних систем для простих випадків, був би необхідний людський інтелект рівня інтелекту шахістів світового рівня [3]. Використання минулого досвіду не має на увазі відмови від застосування математичних методів на користь інтуїції професіоналів, скоріше навіть навпаки: у більшості сучасних методів оцінки вартості розробки програмного забезпечення із залученням емпіричних даних використовується адаптація параметричних моделей [22].

Навчально-орієнтовані методи, оцінювання за аналогами

Навчально-орієнтовані методи намагаються визначити аналогічні проекти з розробки програмного забезпечення і отримати оцінку тривалості з попереднього досвіду та відмінності між попереднім і новим проектом [13, 14, 64]. Перевагою таких методів, в порівнянні з експертним оцінюванням, є те, що в цьому випадку оцінки базуються на реальних подібних проектах, а не на загальному досвіді експертів. Недоліком цих методів є те, що не дуже

зрозуміло, яким чином порівнювати два проекти, які ключові параметри повинні відстежуватися, а які можуть бути пущені. Визначення ключових параметрів є непростим завданням через дуже особливий характер проектів з розробки програмного забезпечення [13, 14].

Оцінювання за аналогами передбачає використання таких параметрів як тривалість, бюджет, розмір, вага і складність попередніх подібних проектів в якості основи для оцінювання тих же параметрів або вимірювань майбутнього проекту [8]. При оцінюванні тривалості даний метод спирається на фактичну тривалість попередніх подібних проектів в якості основи для оцінювання тривалості поточного проекту. Цей підхід, що дозволяє оцінювати загальну величину, іноді адаптується в залежності від відомих відмінностей у складності проекту.

Найчастіше оцінювання тривалості за аналогами використовується для оцінювання тривалості проекту, коли обсяг детальної інформації про проект обмежений, наприклад на його ранніх фазах. При оцінюванні за аналогами використовується історична інформація та експертне оцінювання.

Як правило, оцінювання за аналогами обходиться дешевше і займає менше часу, ніж інші методи, але при цьому отримані оцінки зазвичай виявляються і менш точними. Оцінювання за аналогами може застосовуватися до всього проекту або до його частин, а також може використовуватися разом з іншими методами оцінювання. Оцінювання за аналогам виявляється найбільш надійним в тих випадках, коли попередні операції схожі по суті, а не тільки за формою, а члени команди проекту, що готують оцінки, мають необхідний досвід [8].

Алгоритмічні методи

Алгоритмічні методи використовують ітераційний підхід, який базується на математичних моделях та формулах [13]. При цьому вхідними даними є розмір програмного забезпечення проекту (розрахований у функціональних точках чи рядках програмного коду) та такі параметри, як апаратне забезпечення, програмне забезпечення, рівень командного досвіду, рівень

досвіду менеджерів і метод, що використовується для розробки програмного забезпечення. Алгоритм оцінювання дозволяє за вхідними даними визначити оцінку тривалості та її точність. Алгоритмічні методи ітеративно запускаються кілька разів, щоб уточнити значення вхідних даних та щоб підвищити точність оцінювання. Обмеження цього класу методів проявляється, коли алгоритм використовується з неточними чи з неперевіреними даними. Більшість алгоритмічних методів дозволяють оцінювати тривалість, трудомісткість і навіть загальну вартість проектів [13]. До алгоритмічних методів відносяться COSOMO [9] і COSOMO 2.0 [16], SLIM [23], нейроні мережі, метод критичного шляху, метод критичного ланцюга, метод PERT.

Оцінювання тривалості за трьома точками (метод PERT)

Точність оцінок тривалості робіт може бути покращена за допомогою розгляду невизначеностей оцінок та ризиків [8, 12]. Дана концепція основана на Методі оцінки та аналізу програм (PERT). Для оцінки діапазону тривалості робіт PERT використовує три оцінки:

- найбільш вірогідна (D_m). Тривалість визначається з урахуванням попереднього виділення ресурсів, їх продуктивності, реалістичної оцінки їх доступності для виконання, залежно від інших учасників і затримок;

- оптимістична (D_o). Тривалість ґрунтується на аналізі найбільш сприятливого сценарію розвитку;

- песимістична (D_p). Тривалість ґрунтується на аналізі найбільш несприятливого сценарію розвитку.

Аналіз PERT дозволяє визначити очікувану (D_e) тривалість за допомогою обчислення середнього зваженого цих трьох оцінок:

$$D_e = \frac{D_o + 4D_m + D_p}{6}.$$

Метод PERT є реалізованим в більшості інструментальних засобів для управління проектами. Його недоліком є те, що формула оцінки середнього

значення базується на бета-розподілі, хоча реальні дані можуть відрізнятися від цього розподілу. Крім того, метод PERT використовує експертні оцінки, тому він має загальні недоліки експертного оцінювання. Експерти можуть бути занадто оптимістичними в оцінках чи можуть не мати необхідного досвіду.

Методи, що базуються на математично-статистичних моделях тривалості.

Методи, що базуються на математично-статистичних моделях тривалості є особливо корисними, коли вони базуються на великій кількості історичних даних для аналізу [13]. До цих моделей відносяться лінійна і множинна регресії та їх модифікації [10, 11, 24, 25].

Модель Путнема SLIM [23] (Software Life-cycle Model, модель життєвого циклу програмного забезпечення) була запропонована в 1978 р. Лоуренсом Путнемом (Lawrence Putnam) і стала першою нелінійною моделлю, що використовує емпіричні дані й знайшла практичне застосування при оцінюванні програмного забезпечення. Модель SLIM створена для програмного забезпечення розміром більш ніж 70 тисяч рядків коду і базується на припущенні, що витрати на розробку програмного забезпечення розподіляються згідно кривих Нордена-Рейлі, які є графіками функції, що показує розподіл робочої сили в часі [28]

Модель COCOMO [9] (Constructive Cost Model, модель витрат розробки) – це алгоритмічний метод оцінювання вартості та тривалості проектів з розробки програмного забезпечення, який базується на регресійних моделях. Параметри регресійних моделей були визначені за даними, зібраними за низкою проектів. Регресійні моделі COCOMO були побудовані для різних типів проектів:

- органічні (organic) – 2-50 тисяч рядків програмного коду (програми, призначені для проміжних розрахунків, для потреб програміста, для наукових обчислень);

- напів-розділені (semi-detached) – 50-300 тисяч рядків програмного коду (прикладні системи, компілятори);

- вбудовані (embedded) – більше 300 тисяч рядків коду (системи контролю повітряного руху, мережі АТМ, військові системи).

З врахуванням того, що 1 людино-місяць – це 152 людино-години, регресійні моделі СОСОМО для оцінювання тривалості робіт в залежності від трудомісткості проектів з розробки програмного забезпечення можуть бути представлені наступним чином:

$$D = 0,371 \cdot E^{0,38} \text{ (для органічного типу проектів),} \quad (1.1)$$

$$D = 0,431 \cdot E^{0,35} \text{ (для напів-розділеного типу проектів),} \quad (1.2)$$

$$D = 0,501 \cdot E^{0,32} \text{ (для вбудованого типу проектів),} \quad (1.3)$$

де D – тривалість проекту (в місяцях);

E – трудомісткість проекту (в людино-годинах).

Побудовані за (1.1), (1.2) та (1.3) моделі показані на рис. 1.3.

Модель ISBSG [10, 11] – нелінійна регресійна модель для оцінювання тривалості проектів з розробки програмного забезпечення. Як фактор в цих регресійних моделях було використано трудомісткість проектів з розробки програмного забезпечення:

$$D = 0,662 \cdot E^{0,328}, \quad (1.4)$$

де D – тривалість проекту (в місяцях);

E – трудомісткість проекту (в людино-годинах).

Побудована за (1.4) нелінійна регресійна модель ISBSG показана на рис. 1.4.

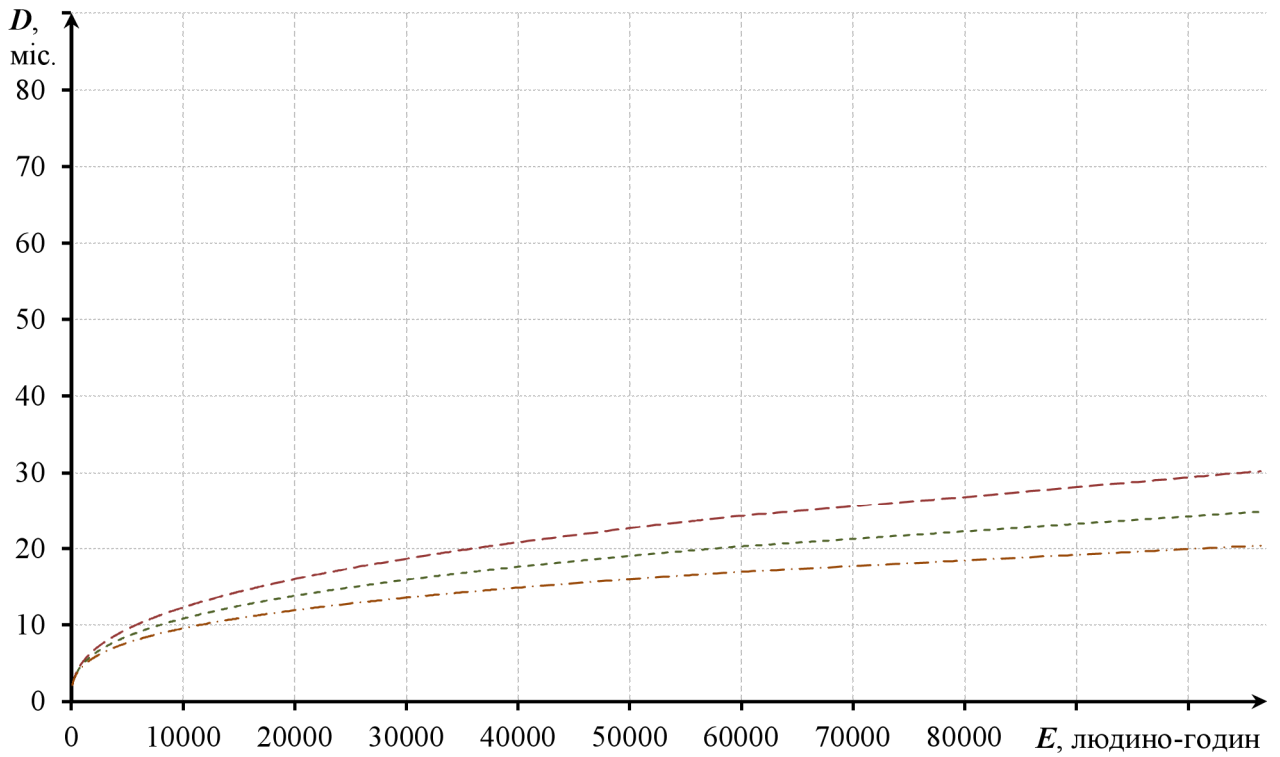


Рис. 1.3. Регресійні моделі COCOMO:

- — органічний (organic) тип проектів;
- — напів-розділений (semi-detached) тип проектів;
- — вбудований (embedded) тип проектів

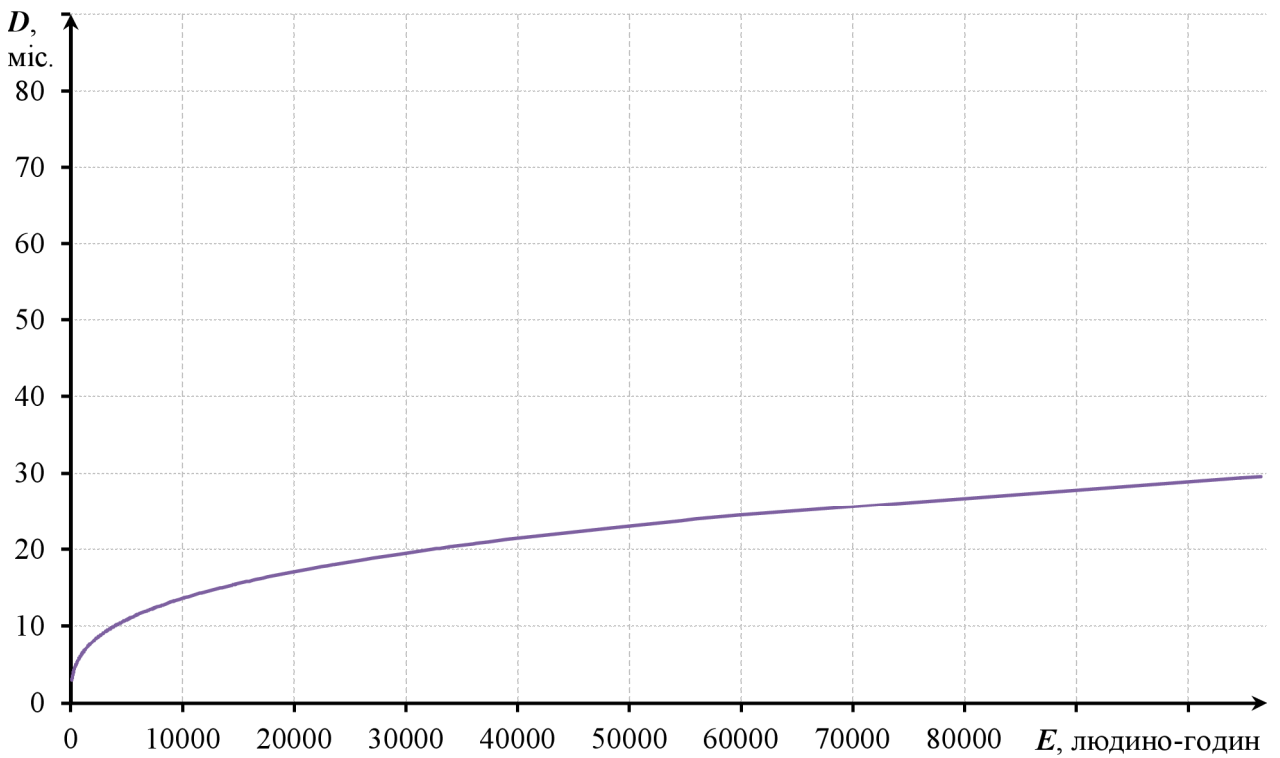


Рис. 1.4. Регресійна модель ISBSG

Закон розподілу емпіричних даних тривалості проектів з розробки програмного забезпечення та трудомісткості цих проектів відрізняється від нормального закону розподілу [9, 10, 47-56]. Тому при побудові нелінійних регресійних моделей COCOMO та ISBSG виконувалась нормалізація емпіричних даних тривалості проектів з розробки програмного забезпечення та трудомісткості цих проектів. Для нормалізації було використано перетворення у вигляді десяткового логарифму. Але вказане перетворення не завжди дозволяє добре нормалізувати емпіричні дані, що приводе до необхідності використовувати інші перетворення. Крім того, регресійні моделі COCOMO та ISBSG не дозволяють виконувати оцінювання довірчих інтервалів тривалості робіт.

Змішані методи

Змішані методи були створені для того, щоб подолати зростаючий розмір і складність проектів з розробки програмного забезпечення. Змішані методи поєднують в собі використання алгоритмічних, статистичних, математичних методів та методів експертного оцінювання [13].

Таблиця 1.2 узагальнює існуючі методи оцінювання тривалості проектів з розробки програмного забезпечення разом з їхніми перевагами і недоліками.

Таблиця 1.2 – Переваги та недоліки методів та моделей оцінювання тривалості проектів з розробки програмного забезпечення

Назва	Переваги	Недоліки
1	2	3
Методи, що базуються на експертній оцінці	Ці методи найбільш гнучкі і можуть бути легко пристосованими від проекту до проекту, щоб покращити якість оцінок часу виконання.	Суб'єктивність методу. Залежить від досвіду експертів.

Продовження таблиці 1.2.

Назва	Переваги	Недоліки
1	2	3
Навчально-орієнтовані методи	Базуються на реальних проектах, що були виконані раніше.	Необхідність визначення ключових змінних є складним та довгим процесом через специфічні особливості кожного проекту з розробки програмного забезпечення.
Алгоритмічні методи	Дозволяють покращити оцінки завдяки ітеративному виконанню алгоритму. Може бути легко адаптований до змін вхідних значень.	Якщо вхідні дані не були верифіковані і калібровані, то оцінки можуть бути дуже неточними.
Методи, що базуються на математично-статистичних моделях	Легкі в реалізації. Мають дуже сильне наукове підґрунтя.	Потребують значний набір історичних даних. Не завжди добре враховують реальний розподіл даних
Змішані методи	Включають ключові аспекти з усіх інших методів	Є незрілими, не досить розвинені, відсутня достатня формалізація.
Оцінка за трьома точками	Дозволяє враховувати невизначеності оцінок та ризиків завдяки використанню інтервальної оцінки.	Суб'єктивність методу. Залежить від досвіду експертів. Базується на бета-розподілі, який не завжди добре враховує реальний розподіл даних.

1.3 Аналіз шляхів вдосконалення моделей тривалості проектів з розробки програмного забезпечення та постановка завдань дослідження

Аналіз моделей та методів для оцінювання тривалості, які використовуються в сучасних інформаційних технологіях переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення показав, що найчастіше використовуються ймовірнісні моделі тривалості проектів з розробки програмного забезпечення (наприклад бета-розподіл в методі PERT [8]) та нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів, такі як COCOMO [9], ISBSG [10, 11]. Ці моделі мають недоліки, які можуть знижувати достовірність оцінювання тривалості проектів з розробки програмного забезпечення.

Закон розподілу емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення відрізняється від нормального закону розподілу [9, 10, 47-56]. Тому вказані ймовірнісні та регресійні моделі не дозволяють виконувати оцінювання довірчих інтервалів тривалості проектів з розробки програмного забезпечення. Часто в таких випадках закон розподілу вважається нормальним, або використовується непараметричний підхід [66]. Використання непараметричного підходу для оцінювання довірчих інтервалів чи виконання інтервального оцінювання в припущенні про нормальний закон розподілу емпіричних даних [66] буде знижувати достовірність оцінювання тривалості проектів з розробки програмного забезпечення.

Через те, що розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення та трудомісткості цих проектів відрізняється від нормального закону розподілу, і тому неможливо побудувати адекватні лінійні регресійні моделі, виникає необхідність побудови нелінійних регресійних моделей з використанням нормалізуючих перетворень. При побудові існуючих

нелінійних регресійних моделей COCOMO [9] та ISBSG [10, 11] для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення та емпіричних даних трудомісткості цих проектів було застосоване перетворення у вигляді десяткового логарифму, яке не завжди добре дозволяє нормалізувати вказані емпіричні дані, що може призводити до зниження достовірності оцінювання тривалості проектів з розробки програмного забезпечення.

В [51] показано, що існуючі ймовірнісні моделі (наприклад, бета-розподіл) не завжди адекватно описують розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення, що може призводити до зниження достовірності оцінювання тривалості. Крім того, як було вже сказано, вказані ймовірнісні моделі не дозволяють виконувати оцінювання довірчих інтервалів тривалості проектів з розробки програмного забезпечення.

Для того щоб підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення, потрібно побудувати моделі тривалості, які будуть краще враховувати розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення та трудомісткості, а також дозволять виконувати оцінювання довірчих інтервалів тривалості проектів з розробки програмного забезпечення, тому що інтервальна оцінка дозволить враховувати невизначеності точкових оцінок.

Так як закон розподілу емпіричних даних тривалості проектів з розробки програмного забезпечення та трудомісткості цих проектів відрізняється від нормального закону розподілу, то виникає проблема знаходження довірчих інтервалів тривалості проектів з розробки програмного забезпечення. Для вирішення вказаної проблеми при побудові ймовірнісної моделі тривалості проектів з розробки програмного забезпечення можна використати метод оцінювання довірчих інтервалів статистичних моментів на основі нормалізуючих перетворень [69, 70]. Для вирішення цієї ж проблеми при побудові нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів можна

використати метод побудови рівнянь нелінійної регресії та метод побудови довірчого інтервалу регресії на основі нормалізуючих перетворень [59, 60].

Використання наведених в [59, 60, 69, 70] методів на основі нормалізуючих перетворень дасть змогу побудувати адекватніші моделі тривалості проектів з розробки програмного забезпечення за рахунок кращого врахування розподілу емпіричних даних тривалості та трудомісткості (врахування асиметрії, ексцесу, несиметричності розподілу емпіричних даних). Також це дасть змогу виконувати оцінювання довірчих інтервалів тривалості проектів з розробки програмного забезпечення.

Для побудови математичних моделей тривалості проектів з розробки програмного забезпечення можна використати дані по завершених проектах з розробки програмного забезпечення з бази даних International Software Benchmarking Standards Group (ISBSG) [10, 11]. Вибір вказаної бази даних пов'язаний з досить великою кількістю проектів (790 проектів з розробки програмного забезпечення), наявністю про якість статистичних даних для кожного проекту (що дозволяє використовувати дані лише про проекти з найвищою якістю статистичних даних), а також з наявністю даних про платформу, тривалість та трудомісткість кожного проекту (що дозволяє будувати як ймовірнісні моделі тривалості, так і регресійні моделі тривалості в залежності від трудомісткості для платформ PC, mid-range та mainframe). До бази даних ISBSG входять проекти зі створення та вдосконалення програмного забезпечення на мовах програмування C, C++, JAVA, Visual Basic, COBOL/COBOL II, ПЛ/1, Access, Focus, Powerbuilder (інформація про кваліфікацію розробників програмного забезпечення в базі даних відсутня). Вказане програмне забезпечення призначалося для використання в системах підтримки прийняття рішень, управління операціями та інших інформаційних системах.

Щоб підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення, потрібно вирішити наступні задачі:

- побудувати ймовірнісні моделі тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe, які будуть враховувати розподіл емпіричних даних тривалості і дозволять виконувати оцінювання довірчого інтервалу точкових оцінок математичного сподівання тривалості;

- вибрати нормалізуюче перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe;

- вибрати нормалізуюче перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe;

- побудувати нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів, які дозволять виконувати визначення довірчих інтервалів та інтервалів прогнозування тривалості, використовуючи вибране нормалізуюче перетворення для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення;

- розробити архітектуру автоматизованої інформаційної системи та створити інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення;

- створити інженерну методіку статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.

1.4 Висновки до розділу 1

В даному розділі було проаналізовано сучасний стан інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.

1. Показано, що більшість проектів з розробки програмного забезпечення не виконуються в заданий термін.
2. Визначено, що одним з основних факторів, який впливає на успішність виконання проектів з розробки програмного забезпечення та дозволяє вчасно завершувати такі проекти, є достовірність оцінювання тривалості цих проектів. Достовірність оцінювання тривалості проектів з розробки програмного забезпечення в значній мірі залежить від моделей та методів оцінювання тривалості.
3. Встановлено, що основними інструментами і методами оцінювання тривалості проектів з розробки програмного забезпечення є експертне оцінювання (методи Delphi та Wideband Delphi), оцінювання за аналогами, параметричне оцінювання (моделі та методи COCOMO, ISBSG), оцінювання за трьома точками (метод PERT), а також змішані методи.
4. Показано, що існуючі моделі не досить добре враховують розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення, а також не дозволяють виконувати оцінювання довірчих інтервалів статистичних моментів тривалості проектів з розробки програмного забезпечення. Це призводить до зниження достовірності оцінювання тривалості проектів з розробки програмного забезпечення, тому виникає необхідність вдосконалення існуючих і побудови нових моделей тривалості цих проектів.
5. Для оцінювання тривалості проектів з розробки програмного забезпечення найчастіше використовуються ймовірнісні моделі тривалості робіт цих проектів та регресійні моделі тривалості проектів з розробки програмного

забезпечення в залежності від трудомісткості цих проектів (COCOMO, ISBSG).

6. Для підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення потрібно побудувати ймовірнісну модель тривалості проектів з розробки програмного забезпечення, яка буде враховувати розподіл емпіричних даних тривалості і дозволить виконувати оцінювання довірчих інтервалів математичного сподівання тривалості проектів з розробки програмного забезпечення.
7. Для підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення з використанням регресійних моделей і вдосконалення існуючих нелінійних регресійних моделей ISBSG, побудованих з використанням перетворення у вигляді десяткового логарифму, потрібно побудувати нелінійні регресійні моделі, довірчі інтервали та інтервали прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів за рахунок використання нормалізуючих перетворень, які краще нормалізують відповідні емпіричні дані, ніж десятковий логарифм.
8. Потрібно розробити архітектуру автоматизованої інформаційної системи та створити інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.
9. Необхідно створити інженерну методику статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.

Результати досліджень даного розділу опубліковано в роботах [47-56, 78, 79].

РОЗДІЛ 2

ПОБУДОВА НЕГАУСІВСЬКИХ ЙМОВІРНІСНИХ МОДЕЛЕЙ ТРИВАЛОСТІ ПРОЕКТІВ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ НОРМАЛІЗУЮЧИХ ПЕРЕТВОРЕНЬ

На даний час часто в якості ймовірнісної моделі тривалості проектів з розробки програмного забезпечення використовується бета-розподіл. В [51] показано, що ця ймовірнісна модель не завжди адекватно апроксимує реальний розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення, що може знижувати достовірність оцінювання тривалості цих проектів.

Також, через те що закон розподілу емпіричних даних тривалості проектів з розробки програмного забезпечення відрізняється від нормального закону розподілу [9, 10, 47-56], неможливо виконувати оцінювання довірчих інтервалів точкових оцінок математичного сподівання тривалості цих проектів. Знаходження інтервальних оцінок статистичних моментів випадкової величини можливе лише в тих випадках, коли закон розподілу емпіричних даних є нормальним чи експоненціальним. У випадках коли закон розподілу емпіричних даних відрізняється від нормального закону розподілу, можна використати нормалізуючі перетворення, які дозволяють здійснювати перехід до випадкової величини з нормальним законом розподілу.

2.1 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення

Розглянемо детальніше основну ймовірнісну модель, яка використовується в існуючих методах оцінювання тривалості проектів з розробки програмного забезпечення, визначимо її недоліки та шляхи їх усунення.

В існуючих методах оцінювання тривалості проектів з розробки програмного забезпечення використовується така ймовірнісна модель як бета-розподіл. Щільність ймовірності бета-розподілу задається як

$$f_{\beta}(x) = \beta(a, b)^{-1} x^{a-1} (1-x)^{b-1}, \quad 0 < x < 1, \quad (2.1)$$

де $\beta(a, b) = \int_0^1 x^{a-1} (1-x)^{b-1} dx$ – бета функція Ейлера;

a, b – параметри розподілу, $a > 0, b > 0$.

У випадку якщо для апроксимації емпіричних даних необхідно розширити чи змістити область визначення щільності ймовірності (2.1), то щільність ймовірності бета-розподілу задається як

$$f_{\beta}(x) = \frac{1}{\lambda} \beta(a, b)^{-1} \left[\frac{x - \varphi}{\lambda} \right]^{a-1} \left[1 - \frac{x - \varphi}{\lambda} \right]^{b-1}, \quad \varphi < x < \varphi + \lambda, \quad (2.2)$$

де $\beta(a, b) = \int_0^1 x^{a-1} (1-x)^{b-1} dx$ – бета функція Ейлера;

a, b, λ, φ – параметри розподілу, $a > 0, b > 0, \lambda > 0$.

Графіки щільності ймовірності бета-розподілу для різних значень a, b при $\lambda=1, \varphi=0$ показані на рис. 2.1.



Рис. 2.1. Щільність ймовірності бета-розподілу для різних значень a, b при $\lambda=1, \varphi=0$

Недоліком бета-розподілу є те, що він не завжди адекватно апроксимує розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення [51]. Крім того, ця ймовірнісна модель не дозволяє виконувати оцінювання довірчих інтервалів точкових оцінок математичного сподівання тривалості проектів з розробки програмного забезпечення. Тому виникає необхідність використовувати інші ймовірнісні моделі.

Для того щоб краще апроксимувати емпіричних даних тривалості проектів з розробки програмного забезпечення, а також щоб отримати можливість інтервального оцінювання тривалості проектів з розробки програмного забезпечення, можна використати таку ймовірнісну модель як розподіл Джонсона. Джонсон запропонував 3 сім'ї розподілів. Для розподілу Джонсона сім'ї S_L функція щільності ймовірності задається як [68]

$$f_L(x) = \frac{\eta}{\sqrt{2\pi}(x-\varphi)} e^{-\frac{\eta^2}{2} \left[\frac{\gamma - \eta \ln \lambda}{\eta} + \ln(x-\varphi) \right]^2}, \quad x > \varphi, \quad (2.3)$$

де $\gamma, \eta, \lambda, \varphi$ – параметри розподілу;

$\eta > 0; -\infty < \gamma < \infty; \lambda > 0; -\infty < \varphi < \infty$.

Функція щільності ймовірності для сім'ї S_L завжди унімодальна, асиметрія завжди додатна, а ексцес більший за 3.

Для розподілу Джонсона сім'ї S_B функція щільності ймовірності задається формулою

$$f_B(x) = \frac{\eta\lambda}{\sqrt{2\pi}(x-\varphi)(\lambda+\varphi-x)} e^{-\frac{1}{2} \left[\gamma + \eta \ln \left(\frac{x-\varphi}{\lambda+\varphi-x} \right) \right]^2}, \quad \varphi < x < \varphi + \lambda, \quad (2.4)$$

де $\gamma, \eta, \lambda, \varphi$ – параметри розподілу;

$\eta > 0; -\infty < \gamma < \infty; \lambda > 0; -\infty < \varphi < \infty$.

Функції щільності ймовірності сім'ї S_B можуть бути як унімодальними, так і бімодальними. Необхідні та достатні умови для бімодальності полягають у тому, що $\eta < 1/\sqrt{2}$, $|\gamma| < \eta^{-1} \sqrt{1-2\eta^2} - 2\eta \operatorname{arth} \sqrt{1-2\eta^2}$.

Для розподілу Джонсона сім'ї S_U функція щільності ймовірності визначається як

$$f_U(x) = \frac{\eta}{\sqrt{2\pi \left\{ (x-\varphi)^2 + \lambda^2 \right\}}} e^{-\frac{1}{2} \left[\gamma + \eta \ln \left(\tilde{x} + \sqrt{\tilde{x}^2 + 1} \right) \right]^2}, \quad -\infty < x < \infty, \quad (2.5)$$

де $\gamma, \eta, \lambda, \varphi$ – параметри розподілу;

$\tilde{x} = (x-\varphi)/\lambda$;

$\eta > 0; -\infty < \gamma < \infty; \lambda > 0; -\infty < \varphi < \infty$.

Сім'я розподілу Джонсона підбирається за значеннями оцінок асиметрії у квадраті A^2 та ексцесу ε [69-75]. Для цього спочатку обчислюються вибіркоче середнє \bar{x} та незміщена вибіркова дисперсія S^2 , а потім за ними обчислюються коефіцієнт асиметрії A та коефіцієнт ексцесу ε , після чого підбирається сім'я розподілу Джонсона.

Вибіркове середнє випадкової величини x обчислюється як

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (2.6)$$

де n - це кількість емпіричних значень випадкової величини x ;

x_i - i -значення випадкової величини x .

Незміщена вибіркова дисперсія випадкової величини x обчислюється як

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad (2.7)$$

де n - це кількість емпіричних значень випадкової величини x ;

x_i - i -значення випадкової величини x ;

\bar{x} - вибіркоче середнє випадкової величини x , що обчислюється за (2.6).

Тоді, коефіцієнт асиметрії випадкової величини x обчислюється як

$$A = \frac{n}{(n-1)(n-2)} \sum_{i=1}^n (x_i - \bar{x})^3 \frac{1}{(S^2)^{3/2}}, \quad (2.8)$$

де n - це кількість емпіричних значень випадкової величини x ;

x_i - i -значення випадкової величини x ;

\bar{x} - вибіркоче середнє випадкової величини x , що обчислюється за (2.6);

S^2 - незміщена вибіркова дисперсія величини x , що обчислюється за (2.7).

Коефіцієнт ексцесу випадкової величини x обчислюється як

$$\varepsilon = \frac{(n^2 - 2n + 3) \sum_{i=1}^n (x_i - \bar{x})^4 + (2n - 3) \frac{3}{n} \left[\sum_{i=1}^n (x_i - \bar{x})^2 \right]^2}{(n-1)(n-2)(n-3)(S^2)^2}, \quad (2.9)$$

де n - це кількість емпіричних значень випадкової величини x ;

x_i - i -значення випадкової величини x ;

\bar{x} - вибіркове середнє випадкової величини x , що обчислюється за (2.6);

S^2 - незміщена вибіркова дисперсія величини x , що обчислюється за (2.7).

Після знаходження коефіцієнта асиметрії A та коефіцієнта ексцесу ε підбирається сім'я розподілу Джонсона. На сьогоднішній день для вибору конкретної сім'ї розподілу Джонсона використовують або відому діаграму в площині ексцес – асиметрія в квадраті $\varepsilon - A^2$ [69-75], або аналітичну залежність $\varepsilon(A^2)$ [76].

В площині $\varepsilon - A^2$ присутні дві лінії, які й визначають конкретну сім'ю розподілів Джонсона (рис. 2.2). Якщо точка з координатами (A^2, ε) знаходиться біля лінії S_L , то можна використовувати розподіл з сім'ї S_L . Якщо точка з координатами (A^2, ε) знаходиться вище лінії S_L , то можна використовувати розподіл з сім'ї S_U , а якщо нижче лінії S_L до лінії критичної області – то розподіл з сім'ї S_B . Якщо ж точка потрапляє у критичну область, то використовувати для апроксимації сім'ї розподілів Джонсона не можна.

При використанні аналітичних залежностей верхня границя критичної області є прямою лінією і може бути задана наступним рівнянням [76]:

$$\varepsilon = A^2 + 1 \quad (2.10)$$

Діаграми, наведені в [72, 73, 74], наводяться для діапазону $A^2 \in [0; 4]$, а існуюча лінійна апроксимація залежності $\varepsilon(A^2)$ для лінії S_L , яка наведена в [76], дає непогані результати в діапазоні $A^2 \in [0; 5]$:

$$\varepsilon = 3 \cdot (1 + 0,641A^2) \quad (2.11)$$

Для вибірок емпіричних даних, в яких оцінки асиметрії в квадраті A^2 значно перевищують значення 5, для лінії S_L , необхідно використовувати наступну нелінійну залежність [77]:

$$\begin{aligned} \varepsilon(A^2) = & 7,2315 \cdot 10^{-6} A^8 - 6,9860 \cdot 10^{-4} A^6 + 4,5460 \cdot 10^{-2} A^4 + \\ & + 1,7979 A^2 + 2,9891 \end{aligned} \quad (2.12)$$

Залежності (2.10) та (2.12), відображені в площині ексцес – асиметрія в квадраті на рис. 2.2.

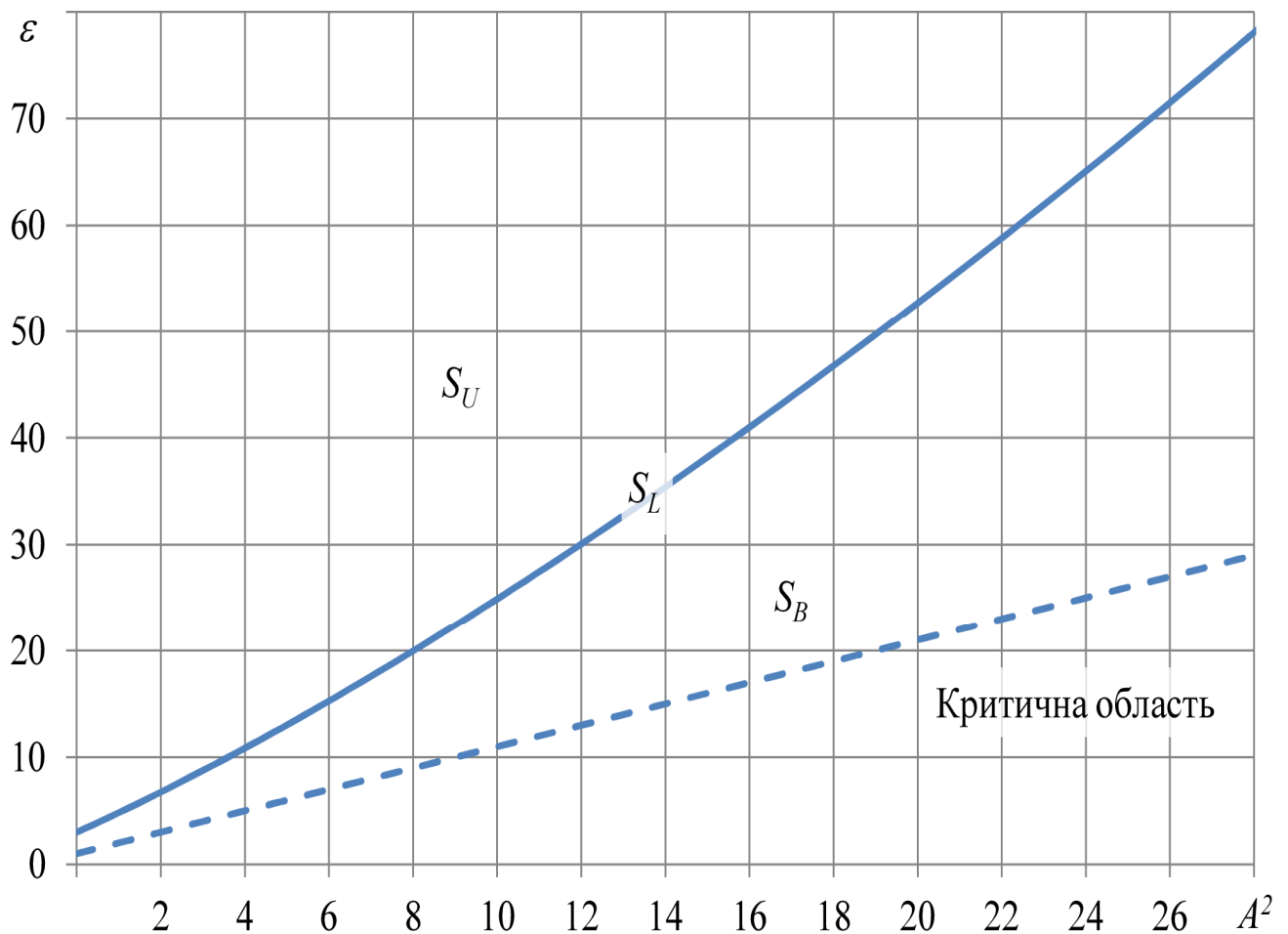


Рис. 2.2. Залежність ексцесу від асиметрії в квадраті для різних сімей розподілу

Джонсона:

- — - залежність (2.10);
- - залежність (2.12)

Параметри розподілів (2.1), (2.2), (2.3), (2.4), (2.5) знаходять за емпіричним розподілом в результаті рішення наступної задачі математичного програмування:

$$\hat{\theta} = \arg \min_{\theta} \left\{ \sum_{j=1}^m [y(x_j) - f(x_j, \theta)]^2 \right\}, \quad (2.13)$$

де $\hat{\theta}$ – оцінка вектору невідомих параметрів;

θ – вектор невідомих параметрів ($\theta = \{a, b\}$ для розподілу (2.1), $\theta = \{a, b, \lambda, \phi\}$ для розподілу (2.2), $\theta = \{\gamma, \eta, \lambda, \phi\}$ для розподілів (2.3), (2.4), (2.5));

x_j – значення x в середині j -го підінтервалу;

$y(x_j)$ – значення ординати гістограми при значенні x_j ;

$f(x_j, \theta)$ – функція щільності ймовірності відповідної сім'ї при значенні x_j ;

m – кількість підінтервалів гістограми.

Параметри розподілів (2.3), (2.4), (2.5) також можна знайти в результаті рішення наступної задачі математичного програмування [70]:

$$\hat{\theta} = \arg \min_{\theta} \left\{ A_z^2 + (\varepsilon_z - 3)^2 + \bar{z}^2 + (S_z^2 - 1)^2 \right\}, \quad (2.14)$$

де $\hat{\theta}$ – оцінка вектору невідомих параметрів;

θ – вектор невідомих параметрів, $\theta = \{\gamma, \eta, \lambda, \phi\}$;

A_z – коефіцієнт асиметрії величини z , що обчислюється за (2.8);

ε_z – коефіцієнт ексцесу величини z , що обчислюється за (2.9);

\bar{z} – вибіркове середнє величини z , $\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$;

S_z^2 – незміщена вибіркова дисперсія величини z , $S_z^2 = \frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})^2$;

z_i – i -значення нормалізованої випадкової величини z , значення якої обчислюються за значеннями випадкової величини x як

$$z = \gamma + \eta q(x, \varphi, \lambda), \quad (2.15)$$

де q – функція, яка залежить від вибраної сім'ї розподілу Джонсона:

$$q(x, \varphi, \lambda) = \ln\left(\frac{x - \varphi}{\lambda}\right), \quad x > \varphi \text{ (сім'я } S_L); \quad (2.16)$$

$$q(x, \varphi, \lambda) = \ln\left(\frac{x - \varphi}{\lambda + \varphi - x}\right), \quad \varphi < x < \varphi + \lambda \text{ (сім'я } S_B); \quad (2.17)$$

$$q(x, \varphi, \lambda) = \operatorname{Arsh}\left(\frac{x - \varphi}{\lambda}\right), \quad -\infty < x < +\infty \text{ (сім'я } S_U), \quad (2.18)$$

де $\gamma, \eta, \lambda, \varphi$ – параметри перетворень;

$$\operatorname{Arsh}\left(\frac{x - \varphi}{\lambda}\right) = \ln\left[\frac{x - \varphi}{\lambda} + \sqrt{\left(\frac{x - \varphi}{\lambda}\right)^2 + 1}\right].$$

Перевірку адекватності ймовірнісних моделей та їх порівняння можна зробити, використавши критерій Пірсона χ^2 .

Для розподілів Джонсона (2.3), (2.4), (2.5) можна також визначити довірчий інтервал точкової оцінки математичного сподівання випадкової величини x , використовуючи для цього значення нормалізованої випадкової величини z , отримані з використанням відповідного для вибраної сім'ї нормалізуючого перетворення Джонсона (2.15).

Для нормальної генеральної сукупності $(1 - \alpha)\%$ довірчий інтервал точкової оцінки математичного сподівання визначається як [67]

$$[\alpha_1(z)] = \left[\bar{z} - t_{\alpha/2, n-1} S_z / \sqrt{n}, \bar{z} + t_{\alpha/2, n-1} S_z / \sqrt{n} \right], \quad (2.19)$$

де n - це кількість емпіричних значень;

$$\bar{z} - \text{вибіркове середнє величини } z, \quad \bar{z} = \frac{1}{n} \sum_{i=1}^n z_i;$$

S_z – середнє квадратичне відхилення величини z , $S_z = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})^2}$;

$t_{\alpha/2, n-1}$ – значення квантілю t -розподілу Стюдента, визначається за таблицею верхніх $100\alpha\%$ точок t -розподілу Стюдента за рівнем значимості $\alpha/2$ та кількістю ступенів вільності $n-1$.

Для того, щоб використовуючи (2.19) визначити довірчий інтервал точкової оцінки математичного сподівання випадкової величини x , спочатку знаходимо значення точкової оцінки математичного сподівання \bar{x} випадкової величини x : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$.

Тоді для випадкової величини x з розподілом Джонсона $(1-\alpha)\%$ довірчий інтервал точкової оцінки математичного сподівання визначається як

$$[\alpha_1(x)] = [z_{\bar{x}} - t_{\alpha/2, n-1} S_z / \sqrt{n}, z_{\bar{x}} + t_{\alpha/2, n-1} S_z / \sqrt{n}] \quad (2.20)$$

де n - це кількість емпіричних значень,

де $z_{\bar{x}}$ – вибіркве середнє величини x після нормалізації перетворенням Джонсона (2.15) відповідної сім'ї, $z_{\bar{x}} = \gamma + \eta q(\bar{x}, \phi, \lambda)$,

S_z – середнє квадратичне відхилення величини z , $S_z = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})^2}$;

$t_{\alpha/2, n-1}$ – значення квантілю t -розподілу Стюдента, визначається за таблицею верхніх $100\alpha\%$ точок t -розподілу Стюдента за рівнем значимості $\alpha/2$ та кількістю ступенів вільності $n-1$.

2.2 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи PC

Побудуємо негаусівську ймовірнісну модель тривалості проектів з розробки програмного забезпечення для платформи PC, використовуючи підхід, показаний в підрозділі 2.1.

Для побудови негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи PC було використано випадкову величину D – емпіричні значення тривалості 52 проектів з розробки програмного забезпечення для платформи PC з бази даних International Software Benchmarking Standards Group (ISBSG) [10, 11]. Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 1,41$, ексцес $\hat{\epsilon}_D = 5,68$; математичне сподівання $\bar{D} = 9,26$, середньоквадратичне відхилення $S_D = 5,59$. Значення критерію Пірсона складає $\chi^2 = 2,81 \cdot 10^1$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7 - 2 - 1 = 4$, $\alpha = 0,05$).

Значення параметрів щільності ймовірності бета-розподілу (2.2) для апроксимації випадкової величини D , знайдені в результаті рішення задачі (2.13), були такими: $a = 1,81891$, $b = 55101$, $\varphi = -1,9248$ і $\lambda = 223770$. Значення критерію Пірсона складає $\chi^2 = 2,78$ (критичне значення $\chi_{кр}^2 = 5,99$, $\nu = 7 - 4 - 1 = 2$, $\alpha = 0,05$). Гістограма випадкової величини D та її теоретичний розподіл у вигляді щільності ймовірності бета-розподілу $f(D)$ зі знайденими параметрами наведені на рис. 2.3.

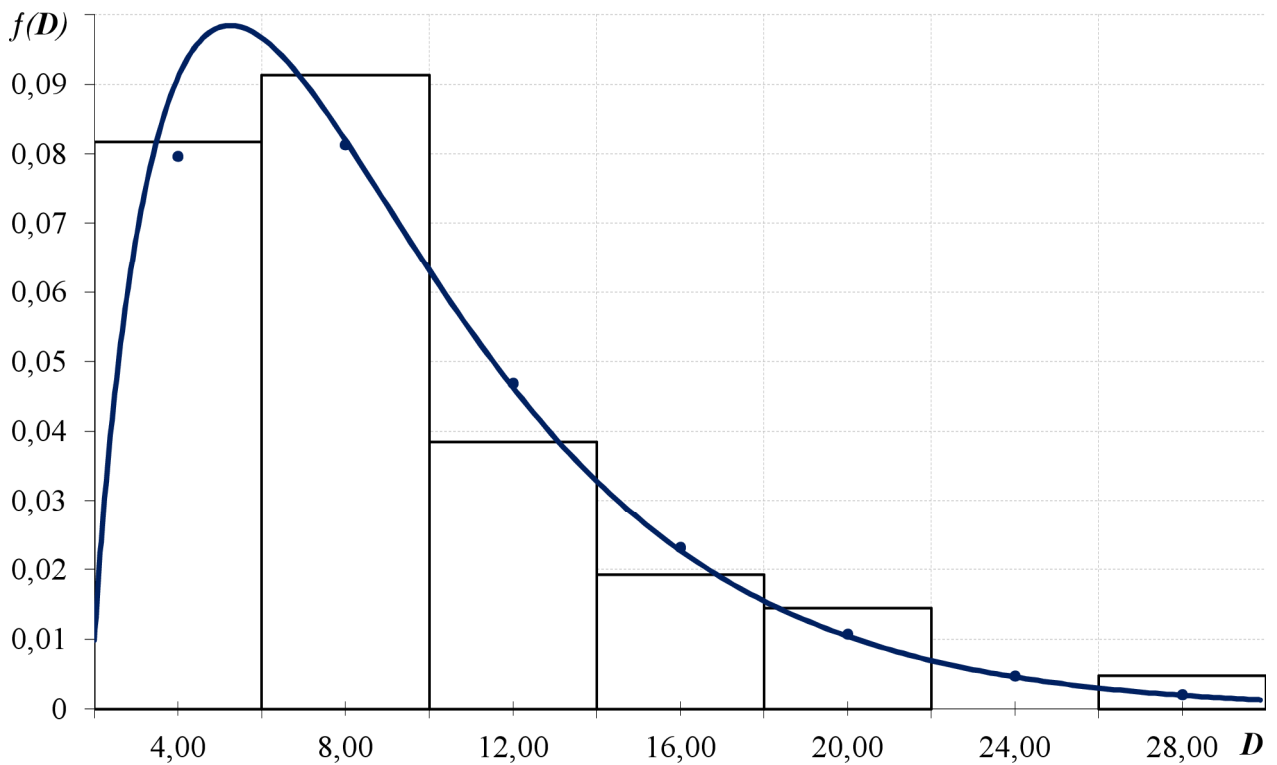


Рис. 2.3. Негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення для платформи РС (бета-розподіл):

□ – емпіричний розподіл величини D ;

—●— – щільність ймовірності бета-розподілу зі знайденими параметрами

По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для побудови негаусівської ймовірнісної моделі випадкової величини D було вибрано розподіл Джонсона сім'ї S_B (2.4). Значення параметрів щільності ймовірності розподілу Джонсона (2.4) для апроксимації випадкової величини D , знайдені в результаті рішення задачі (2.14), були такими: $\gamma = 2,38983$, $\eta = 1,30278$, $\varphi = 0,43283$ і $\lambda = 54,7471$. Значення критерію Пірсона складає $\chi^2 = 2,76$ (критичне значення $\chi_{кр}^2 = 5,99$, $\nu = 7-4-1=2$, $\alpha = 0,05$). Гістограма випадкової величини D та її теоретичний розподіл у вигляді щільності ймовірності розподілу Джонсона $f(D)$ сім'ї S_B (2.4) зі знайденими параметрами наведені на рис. 2.4.

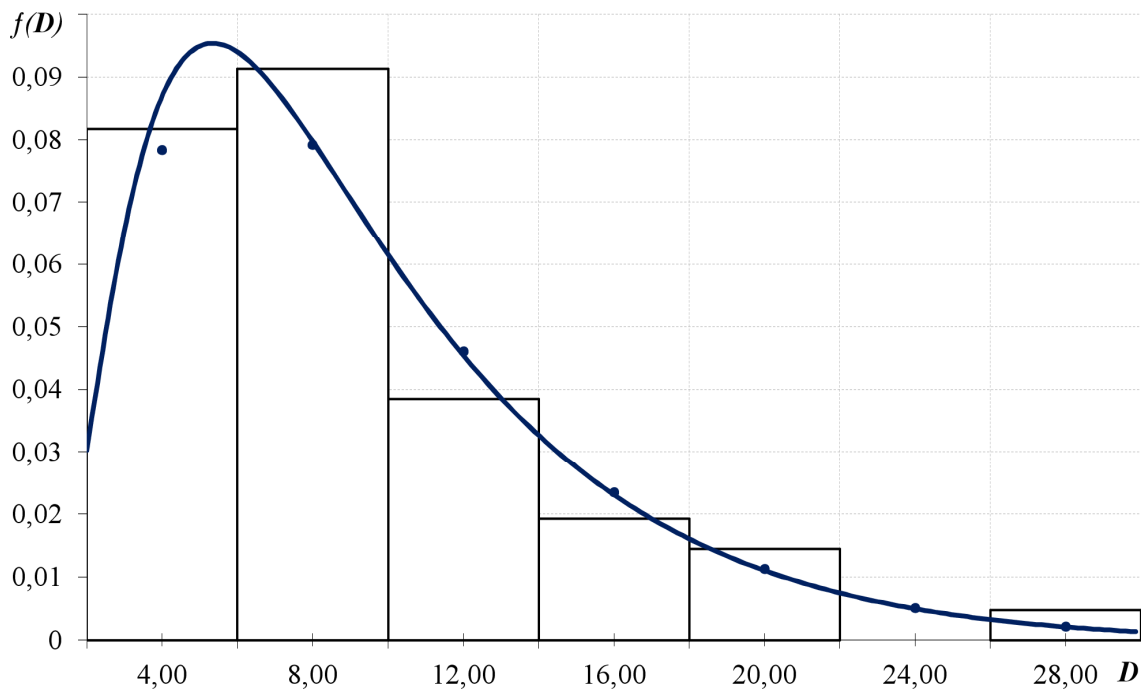


Рис. 2.4. Негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення для платформи РС (розподіл Джонсона):

□ – емпіричний розподіл величини D ;

—●— – щільність ймовірності розподілу Джонсона сім'ї S_B

Для визначення довірчого інтервалу точкової оцінки математичного сподівання випадкової величини D , використовуючи перетворення Джонсона сім'ї S_B (2.17) з вказаними вище параметрами, були отримані значення нормалізованої випадкової величини z . Оцінки ймовірнісних характеристик величини z такі: математичне сподівання $\bar{z} = 0$, середньоквадратичне відхилення $S_z = 1,01$, асиметрія $\hat{A}_z = 0,00$, ексцес $\hat{\varepsilon}_z = 3,12$. Значення критерію Пірсона складає $\chi^2 = 7,85$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$).

Обчислене за (2.20) для випадкової величини D значення 95% довірчого інтервалу точкової оцінки математичного сподівання $[\alpha_1(D)]$ було $[\hat{\alpha}_1(D)] = [7,77, 10,97]$.

Якщо вважати, що випадкова величина D має нормальний розподіл, як це зазвичай робиться при непараметричному оцінюванні [66], то для цього випадку непараметричне оцінювання приведе до збільшення довжини довірчого інтервалу точкової оцінки математичного сподівання на 3% в

порівнянні з наведеним підходом на основі використання перетворення Джонсона.

2.3 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи mid-range

Побудуємо негаусівську ймовірнісну модель тривалості проектів з розробки програмного забезпечення для платформи mid-range, використовуючи підхід, показаний в підрозділі 2.1.

Для побудови негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи mid-range було використано випадкову величину D – емпіричні значення тривалості 81 проекту з розробки програмного забезпечення для платформи mid-range з бази даних International Software Benchmarking Standards Group (ISBSG) [10, 11]. Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 3,830$, ексцес $\hat{\varepsilon}_D = 23,90$; математичне сподівання $\bar{D} = 11,34$, середньоквадратичне відхилення $S_D = 10,45$. Значення критерію Пірсона складає $\chi^2 = 8,61 \cdot 10^5$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$).

Значення параметрів щільності ймовірності бета-розподілу (2.2) для апроксимації випадкової величини D , знайдені в результаті рішення задачі (2.13), були такими: $a = 0,4381$, $b = 2,5797$, $\phi = 1,9999$ і $\lambda = 78,131$. Значення критерію Пірсона складає $\chi^2 = 12,69$ (критичне значення $\chi_{кр}^2 = 5,99$, $\nu = 7-4-1=2$, $\alpha = 0,05$), тобто за цим критерієм дана ймовірнісна модель не дозволяє адекватно апроксимувати розподіл емпіричних даних тривалості проектів з

розробки програмного забезпечення для платформи mid-range. Гістограма випадкової величини D та її теоретичний розподіл у вигляді щільності ймовірності бета-розподілу $f(D)$ зі знайденими параметрами наведені на рис. 2.5.

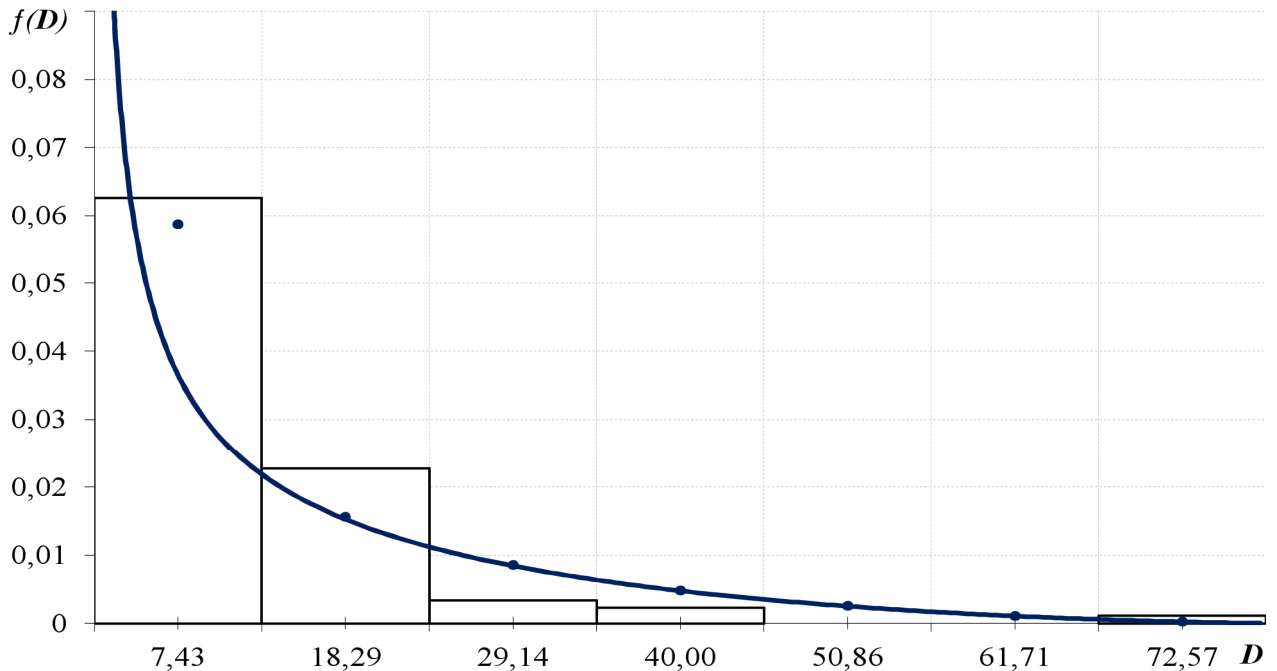


Рис. 2.5. Негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення для платформи mid-range (бета-розподіл):

- – емпіричний розподіл величини D ;
- – щільність ймовірності бета-розподілу зі знайденими параметрами

По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для побудови негаусівської ймовірнісної моделі випадкової величини D було вибрано розподіл Джонсона сім'ї S_B (2.4). Значення параметрів щільності ймовірності розподілу Джонсона (2.4) для апроксимації випадкової величини D , знайдені в результаті рішення задачі (2.14), були такими: $\gamma_D = 5,682$, $\eta_D = 1,13146$, $\varphi_D = 1,04977$ і $\lambda_D = 1166,2$. Значення критерію Пірсона складає $\chi^2 = 5,82$ (критичне значення $\chi_{кр}^2 = 5,99$, $\nu = 7-4-1=2$, $\alpha = 0,05$). Гістограма випадкової величини D та її теоретичний розподіл у вигляді щільності ймовірності розподілу Джонсона $f(D)$ сім'ї S_B (2.4) зі знайденими параметрами наведені на рис. 2.6.

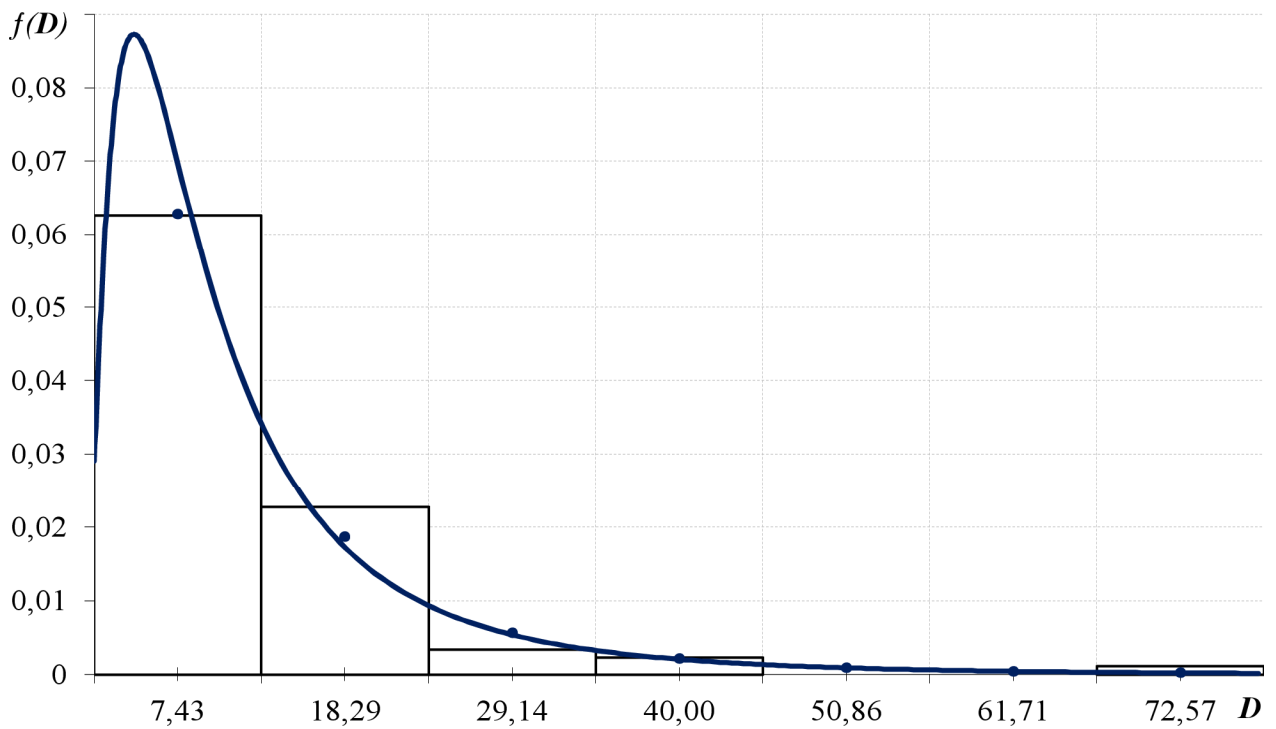


Рис. 2.6. Негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення для платформи mid-range (розподіл Джонсона):

▭ – емпіричний розподіл величини D ;

—●— – щільність ймовірності розподілу Джонсона сім'ї S_B

Для визначення довірчого інтервалу точкової оцінки математичного сподівання випадкової величини D , використовуючи перетворення Джонсона сім'ї S_B (2.17) з вказаними вище параметрами, були отримані значення нормалізованої випадкової величини z . Оцінки ймовірнісних характеристик величини z такі: математичне сподівання $\bar{z}_D = -0,04$, середньоквадратичне відхилення $S_z = 0,95$, асиметрія $\hat{A}_z = -0,035$, ексцес $\hat{\varepsilon}_z = 3,05$. Значення критерію Пірсона складає $\chi^2 = 4,04$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$).

Обчислене за (2.20) для випадкової величини D значення 95% довірчого інтервалу точкової оцінки математичного сподівання $[\alpha_1(D)]$ було $[\hat{\alpha}_1(D)] = [9,61, 13,41]$.

Якщо вважати, що випадкова величина D має нормальний розподіл, як це зазвичай робиться при непараметричному оцінюванні [66], то для цього випадку непараметричне оцінювання приведе до зменшення довжини довірчого

інтервалу точкової оцінки математичного сподівання на 21% в порівнянні з наведеним підходом на основі використання перетворення Джонсона.

2.4 Побудова негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи mainframe

Побудуємо негаусівську ймовірнісну модель тривалості проектів з розробки програмного забезпечення для платформи mainframe, використовуючи підхід, показаний в підрозділі 2.1.

Для побудови негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення для платформи mainframe було використано випадкову величину D – емпіричні значення тривалості 215 проектів з розробки програмного забезпечення для платформи mainframe з бази даних International Software Benchmarking Standards Group (ISBSG) [10, 11]. Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 2,743$, ексцес $\hat{\varepsilon}_D = 14,48$; математичне сподівання $\bar{D} = 11,78$, середньоквадратичне відхилення $S_D = 10,61$. Значення критерію Пірсона складає $\chi^2 = 8,66 \cdot 10^6$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9-2-1=6$, $\alpha = 0,05$).

Значення параметрів щільності ймовірності бета-розподілу (2.2) для апроксимації випадкової величини D , знайдені в результаті рішення задачі (2.13), були такими: $a = 0,9092$, $b = 18276$, $\varphi = 0,9999$ і $\lambda = 225272$. Значення критерію Пірсона складає $\chi^2 = 12,12$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 9-4-1=4$, $\alpha = 0,05$), тобто за цим критерієм дана ймовірнісна модель не дозволяє адекватно апроксимувати розподіл емпіричних даних тривалості проектів з

розробки програмного забезпечення для платформи mainframe. Гістограма випадкової величини D та її теоретичний розподіл у вигляді щільності ймовірності бета-розподілу $f(D)$ зі знайденими параметрами наведені на рис. 2.7.

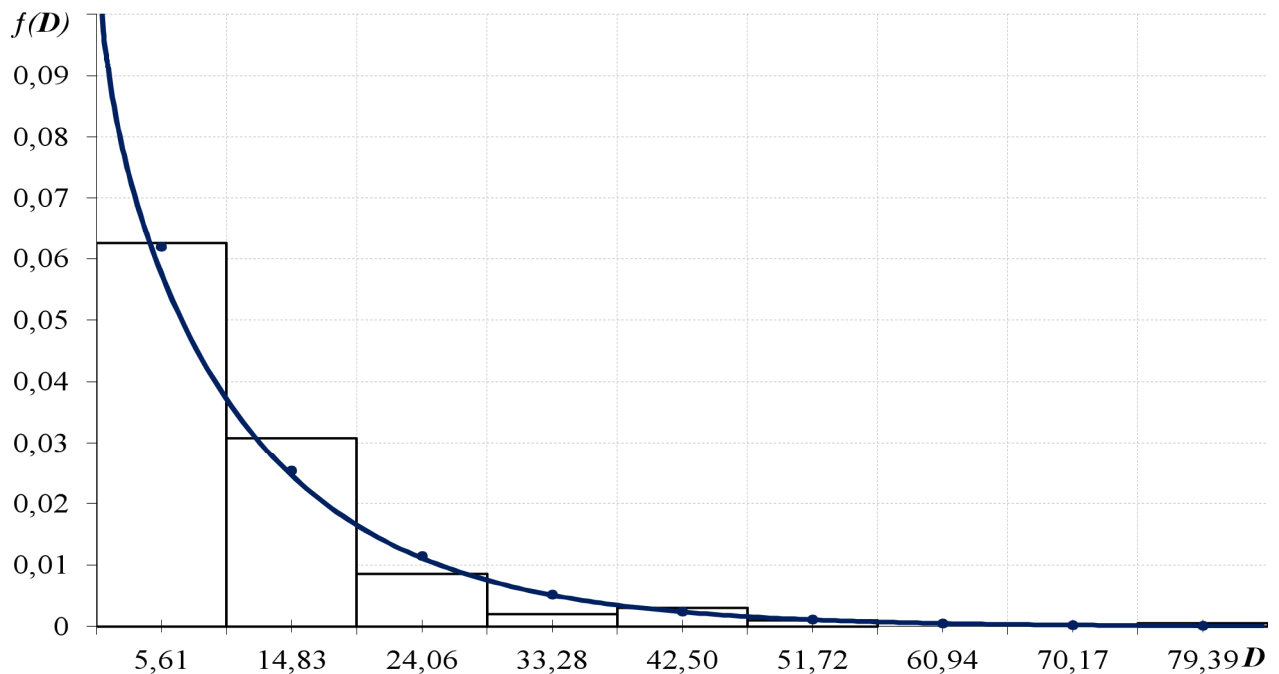


Рис. 2.7. Негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення для платформи mainframe (бета-розподіл):

- – емпіричний розподіл величини D ;
- – щільність ймовірності бета-розподілу зі знайденими параметрами

По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для побудови негаусівської ймовірнісної моделі випадкової величини D було вибрано розподіл Джонсона сім'ї S_B (2.4). Значення параметрів щільності ймовірності розподілу Джонсона (2.4) для апроксимації випадкової величини D , знайдені в результаті рішення задачі (2.14), були такими: $\gamma_D = 9,22106$, $\eta_D = 1,30531$, $\phi_D = -0,25458$ і $\lambda_D = 10513,2$. Значення критерію Пірсона складає $\chi^2 = 8,62$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 9 - 4 - 1 = 4$, $\alpha = 0,05$). Гістограма випадкової величини D та її теоретичний розподіл у вигляді щільності ймовірності розподілу Джонсона $f(D)$ сім'ї S_B (2.4) зі знайденими параметрами наведені на рис. 2.8.

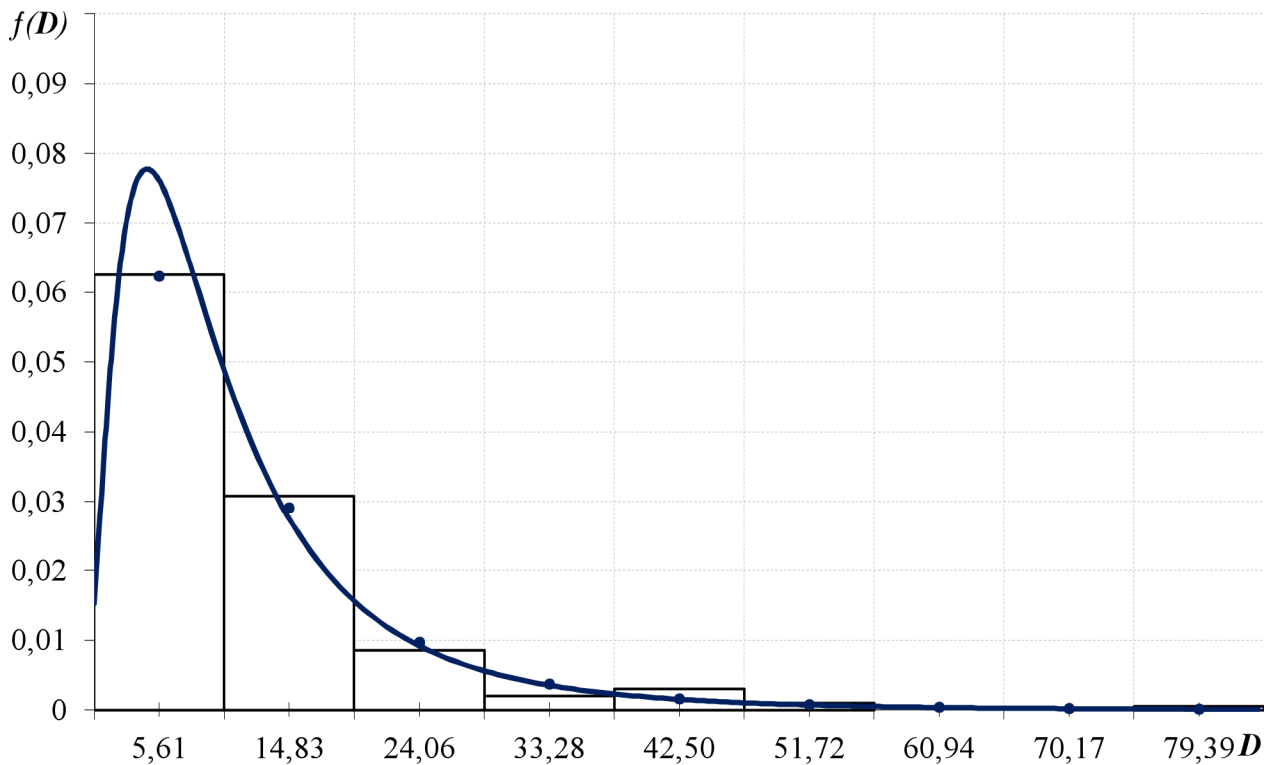


Рис. 2.8. Негавсівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення для платформи mainframe (розподіл Джонсона):

□ – емпіричний розподіл величини D ;

—●— – щільність ймовірності розподілу Джонсона сім'ї S_B

Для визначення довірчого інтервалу точкової оцінки математичного сподівання випадкової величини D , використовуючи перетворення Джонсона сім'ї S_B (2.17) з вказаними вище параметрами, були отримані значення нормалізованої випадкової величини z . Оцінки ймовірнісних характеристик величини z такі: математичне сподівання $\bar{z}_D = 0$, середньоквадратичне відхилення $S_z = 1$, асиметрія $\hat{A}_z = 0$, ексцес $\hat{\varepsilon}_z = 3,03$. Значення критерію Пірсона складає $\chi^2 = 5,95$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$).

Обчислене за (2.20) для випадкової величини D значення 95% довірчого інтервалу точкової оцінки математичного сподівання $[\alpha_1(D)]$ було $[\hat{\alpha}_1(D)] = [10,60, 13,08]$.

Якщо вважати, що випадкова величина D має нормальний розподіл, як це зазвичай робиться при непараметричному оцінюванні [66], то для цього

випадку непараметричне оцінювання приводе до зменшення довжини довірчого інтервалу точкової оцінки математичного сподівання на 15% в порівнянні з наведеним підходом на основі використання перетворення Джонсона.

Результати апроксимації випадкової величини D для платформ PC, mid-range та mainframe наведено в таблиці 2.1. Перевірка адекватності ймовірнісних моделей та їх порівняння, використовуючи критерій Пірсона χ^2 показала, що розподіл Джонсона сім'ї S_B (2.4) дозволяє адекватно апроксимувати розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe ($\chi^2 < \chi_{кр}^2$), тоді як бета-розподіл не дозволяє адекватно апроксимувати розподіл емпіричних даних тривалості ($\chi^2 > \chi_{кр}^2$ для платформ mid-range та mainframe).

Таблиця 2.1 – Результати апроксимації випадкової величини D

Платформа ПЗ	χ^2 (розподіл Джонсона)	χ^2 (бета-розподіл)	$\chi_{кр}^2$
PC	2,76	2,78	5,99
mid-range	5,82	12,69	5,99
mainframe	8,62	12,12	9,49

2.5 Висновки до розділу 2

В даному розділі було виконано побудову негаусівських ймовірнісних моделей тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe у вигляді розподілу Джонсона сім'ї S_B .

1. Було розглянуто бета-розподіл – основну ймовірнісну модель, яка використовується в існуючих методах оцінювання тривалості проектів з розробки програмного забезпечення. Визначено, що ця ймовірнісна модель

не завжди адекватно апроксимує розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення. Для емпіричних даних тривалості проектів з розробки програмного забезпечення з бази даних International Software Benchmarking Standards Group бета-розподілом можна апроксимувати лише емпіричні дані для платформи PC. Емпіричні дані для платформ mid-range та mainframe не можуть бути адекватно апроксимовані бета-розподілом. У разі використання бета-розподілу неможливо виконувати оцінювання довірчих інтервалів точкових оцінок математичного сподівання проектів з розробки програмного забезпечення.

2. Знаходження довірчих інтервалів точкових оцінок математичного сподівання випадкової величини можливе лише в тих випадках, коли закон розподілу емпіричних даних є нормальним чи експоненціальним. Тому виникає необхідність замість бета-розподілу використовувати інші ймовірнісні моделі.
3. Для того щоб краще апроксимувати емпіричних даних тривалості проектів з розробки програмного забезпечення, а також щоб отримати можливість інтервального оцінювання тривалості, використано таку ймовірнісну модель як розподіл Джонсона. Побудовано негаусівські ймовірнісні моделі тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe у вигляді розподілу Джонсона сім'ї S_B . Ці ймовірнісні моделі краще апроксимують розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення, ніж бета-розподіл за критерієм Пірсона χ^2 .
4. Отримано довірчі інтервали точкових оцінок математичного сподівання тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe на основі нормалізуючого перетворення Джонсона сім'ї S_B .

Результати даного розділу були опубліковані в [47-52, 78, 79].

РОЗДІЛ 3

ПОБУДОВА НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ ТРИВАЛОСТІ ПРОЕКТІВ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

На сьогоднішній день при оцінюванні тривалості проектів з розробки програмного забезпечення часто використовуються нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів, такі як COCOMO [9], ISBSG [10, 11]. При побудові вказаних регресійних моделей виконувалась нормалізація емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення. Необхідність використання нормалізуючих перетворень при побудові регресійних моделей виникає через те, що закон розподілу емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення відрізняється від нормального закону розподілу [9, 10, 47-56], і тому неможливо побудувати адекватні лінійні регресійні моделі.

При побудові нелінійних регресійних моделей COCOMO та ISBSG для нормалізації вказаних емпіричних даних було використано десятковий логарифм [9, 10, 11]. Перетворення у вигляді десяткового логарифму не завжди дозволяє добре нормалізувати емпіричні дані, що приводить до необхідності використовувати інші перетворення. Для вдосконалення нелінійних регресійних моделей COCOMO та ISBSG необхідно з нормалізуючих перетворень (3.1), (3.2), (2.15) вибрати нормалізуюче перетворення, яке дозволить краще нормалізувати емпіричні дані та побудувати нові нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі вибраного нормалізуючого перетворення. Також необхідно побудувати довірчі інтервали нелінійних регресій та інтервали прогнозування нелінійних регресій тривалості

проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range та mainframe за рахунок використання нормалізуючих перетворень.

3.1 Аналіз емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення для побудови нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення

Розглянемо основні етапи побудови нелінійних регресійних моделей на основі нормалізуючих перетворень та проведемо аналіз емпіричних даних, використаних для побудови регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range та mainframe.

Нелінійні регресійні моделі на основі нормалізуючих перетворень будуються наступним чином (метод нормалізуючих перетворень): виконується нормалізація емпіричних даних з використанням нормалізуючого перетворення; за нормалізованими даними будується лінійна регресія; за побудованим рівнянням лінійної регресії з використанням перетворення, зворотного до нормалізуючого, будується нелінійна регресійна модель, а також рівняння нижньої та верхньої границь довірчого інтервалу нелінійної регресії, рівняння нижньої та верхньої границь інтервалу прогнозування нелінійної регресії.

Для вдосконалення регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів було використано дані по 348 проектах з розробки програмного забезпечення з бази даних International Software Benchmarking Standards Group (ISBSG) [10, 11].

Для вдосконалення нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі десяткового логарифму (модель ISBSG [10, 11]) було використано змішані дані по тривалості та трудомісткості 348 проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe з бази даних International Software Benchmarking Standards Group [10, 11]. Одиниця вимірювання тривалості – місяць, одиниця вимірювання трудомісткості – людино-година. Діапазон значень тривалості – 1-84 місяці, діапазон значень трудомісткості – 97-104690 людино-годин.

Для вдосконалення нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі десяткового логарифму (модель ISBSG [10, 11]) для платформи PC було використано дані по тривалості та трудомісткості 52 проектів з розробки програмного забезпечення для платформи PC з бази даних International Software Benchmarking Standards Group [10, 11]. Одиниця вимірювання тривалості – місяць, одиниця вимірювання трудомісткості – людино-година. Діапазон значень тривалості – 2-22 місяці, діапазон значень трудомісткості – 170-9296 людино-годин.

Для вдосконалення нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі десяткового логарифму (модель ISBSG [10, 11]) для платформи mid-range було використано дані по тривалості та трудомісткості 81 проекту з розробки програмного забезпечення для платформи mid-range з бази даних International Software Benchmarking Standards Group [10, 11]. Діапазон значень тривалості – 2-78 місяців, діапазон значень трудомісткості – 400-106480 людино-годин.

Для вдосконалення нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі десяткового логарифму (модель ISBSG [10, 11]) для платформи mainframe було використано дані по тривалості та трудомісткості

215 проектів з розробки програмного забезпечення для платформи mainframe з бази даних International Software Benchmarking Standards Group [10, 11]. Діапазон значень тривалості – 1-52 місяці, діапазон значень трудомісткості – 114-59809 людино-годин.

3.2 Нормалізація випадкових величини із використанням нормалізуючих перетворень

Необхідність використання нормалізуючих перетворень при побудові регресійних моделей виникає через те, що закон розподілу емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення відрізняється від нормального закону розподілу [9, 10, 47-56], і тому неможливо побудувати адекватні лінійні регресійні моделі.

Розглянемо перетворення, які можна використати для нормалізації випадкових величин.

Перетворення у вигляді десяткового логарифму. Нормалізацію випадкової величини x за перетворенням у вигляді десяткового логарифму виконуємо за

$$z = \log_{10} x \quad (3.1)$$

де z – нормована нормально розподілена випадкова величина з нульовим математичним сподіванням і одиничною дисперсією;

x – випадкова величина, яка нормалізується;

Перетворення Бокса-Кокса. Перетворення Бокса-Кокса відноситься до перетворень, які не потребують побудови емпіричного розподілу.

Нормалізацію випадкової величини x за перетворенням Бокса-Кокса [65] виконуємо за

$$z = \begin{cases} \frac{x^\lambda - 1}{\lambda}, & \text{якщо } \lambda \neq 0; \\ \ln(x), & \text{якщо } \lambda = 0. \end{cases} \quad (3.2)$$

де z – нормована нормально розподілена випадкова величина з нульовим математичним сподіванням і одиничною дисперсією;

x – випадкова величина, яка нормалізується;

λ – параметр розподілу.

В перетворенні (3.2) параметр λ визначається шляхом максимізації логарифму функції правдоподібності

$$l(\lambda) = C - \frac{n}{2} \ln \sum_{i=1}^n \frac{(z_i - \bar{z})^2}{n} + (\lambda - 1) \sum_{i=1}^n \ln(x_i), \quad (3.3)$$

де C – постійна, яку визначають з умови нормування;

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i.$$

Перетворення Джонсона. В загальному випадку нормалізуюче перетворення Джонсона має вигляд (2.15) [68]. Джонсон запропонував три різні сім'ї функцій q : сім'я S_L (2.16), сім'я S_B (2.17), сім'я S_U (2.18).

Сім'я перетворення Джонсона підбирається по діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A^2 та ексцесу ε [69-75].

Параметри перетворення Джонсона γ , η , λ і ϕ знаходять за емпіричним розподілом [63] шляхом рішення задачі математичного програмування (2.13), або використовуючи непараметричний підхід [70] шляхом рішення задачі математичного програмування (2.14).

Вибір перетворення, яке краще підходить для нормалізації випадкової величини. Виконуючи нормалізацію випадкової величини x за перетвореннями (3.1), (3.2), (2.15) буде отримано значення нормально розподіленої величини z . Для вибору перетворення, яке краще підходить для

нормалізації випадкової величини x необхідно виконати нормалізацію величини x за кожним з перетворень (3.1), (3.2), (2.15) та порівняти результати нормалізації. Щоб порівняти результати нормалізації випадкової величини x та перевірити гіпотезу про нормальний закон розподілу випадкової величини z , можна використати значення критерію узгодженості Пірсона χ^2 , а також порівнявши значення величини δ – розходження характеристик розподілу випадкової величини та характеристик нормального розподілу:

$$\delta = A^2 + |\varepsilon - 3|^2 \quad (3.4),$$

де A – асиметрія розподілу випадкової величини,
 ε – ексцес розподілу випадкової величини.

Розраховане за (3.4) розходження δ показує наскільки характеристики розподілу випадкової величини відрізняються від нормального. Щоб вибрати перетворення, яке дозволяє краще виконати нормалізацію випадкової величини порівнюємо визначені за (3.4) значення δ , та значення χ^2 . Кращим є нормалізуюче перетворення, що дає випадкову величину z з нормальним розподілом, для якої буде отримано менші значення δ та χ^2 .

3.3 Вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення

3.3.1 Вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для всіх платформ

Зробимо вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для всіх платформ.

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe з бази даних ISBSG [10, 11] (348 проектів). Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 3,11$, ексцес $\hat{\epsilon}_D = 17,92$; математичне сподівання $\bar{D} = 11,30$, середньоквадратичне відхилення $S_D = 10,00$. Значення критерію Пірсона складає $\chi^2 = 3,60 \cdot 10^8$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$). Гістограму випадкової величини D приведено на рис. 3.1.

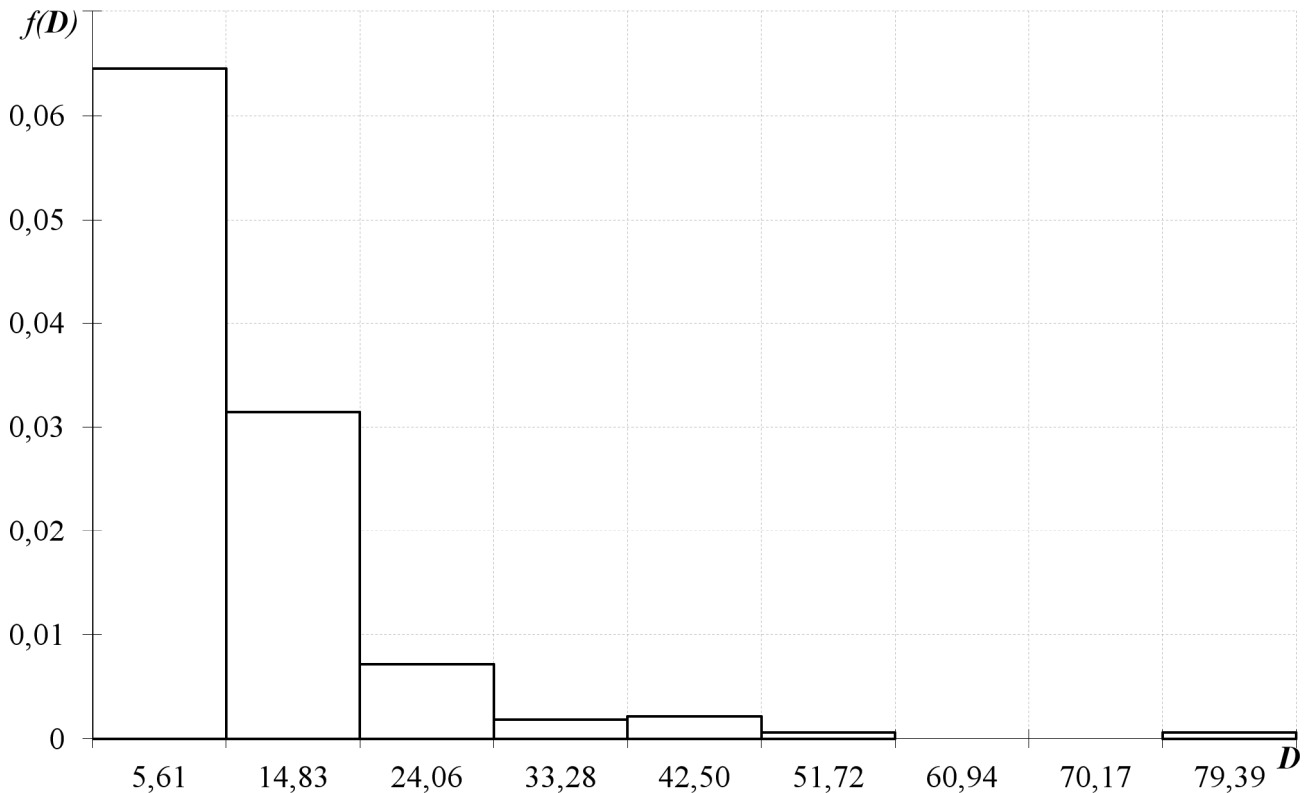


Рис. 3.1. Розподіл випадкової величини D – емпіричних значень тривалості проектів з розробки програмного забезпечення

Необхідно визначити яке з наведених в нормалізуючих перетворень дозволить краще виконати нормалізацію випадкової величини D .

Нормалізація випадкової величини D із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = 0,93$, середньоквадратичне відхилення $S_z = 0,33$, асиметрія $\hat{A}_z = -0,020$, ексцес $\hat{\varepsilon}_z = 3,19$. Значення критерію Пірсона складає $\chi^2 = 14,01$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,036$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.2.

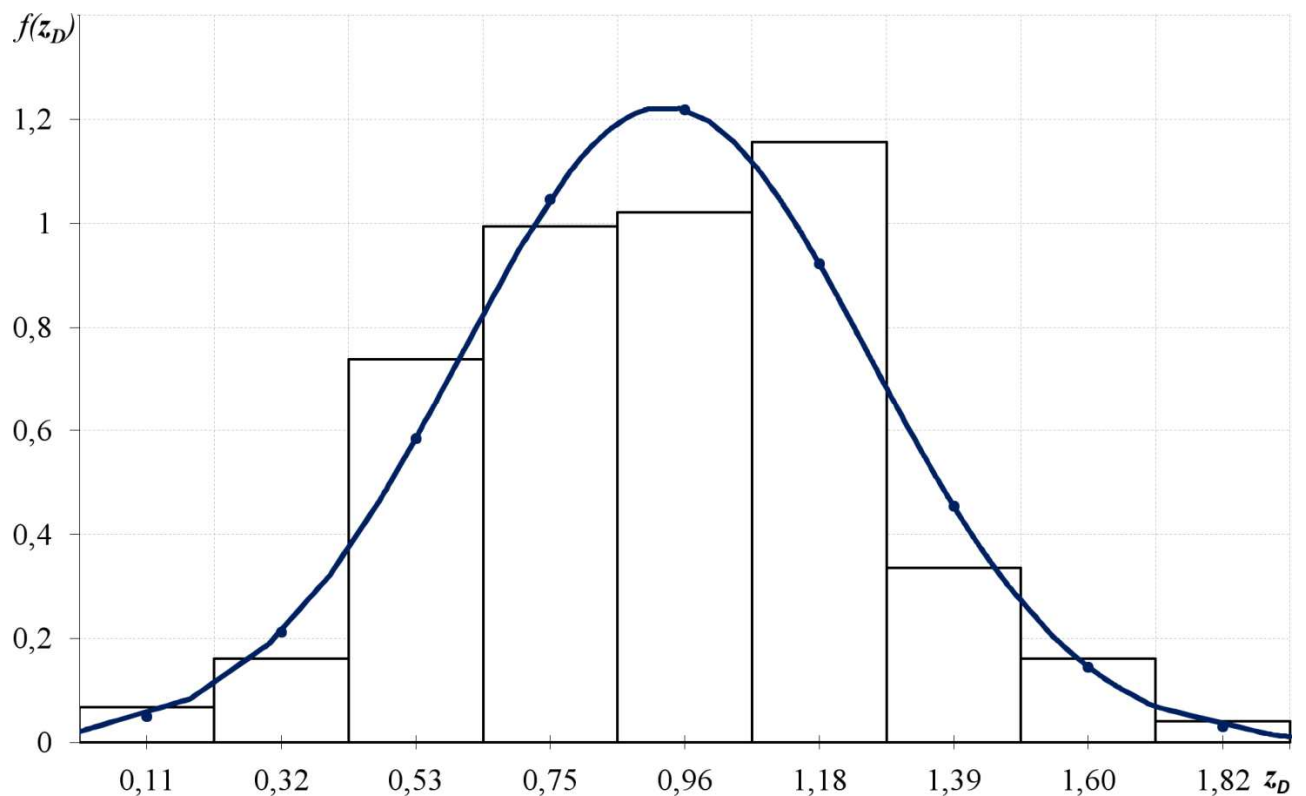


Рис. 3.2. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень тривалості проектів з розробки програмного забезпечення:

▭ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_D . Значення розрахованого за (3.3) параметру $\lambda = 0,01$. Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_t = 2,17$, середньоквадратичне відхилення $S_z = 0,77$, асиметрія $\hat{A}_z = 0,004$, ексцес $\hat{\varepsilon}_z = 3,19$. Значення критерію Пірсона складає $\chi^2 = 6,76$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,037$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.3.

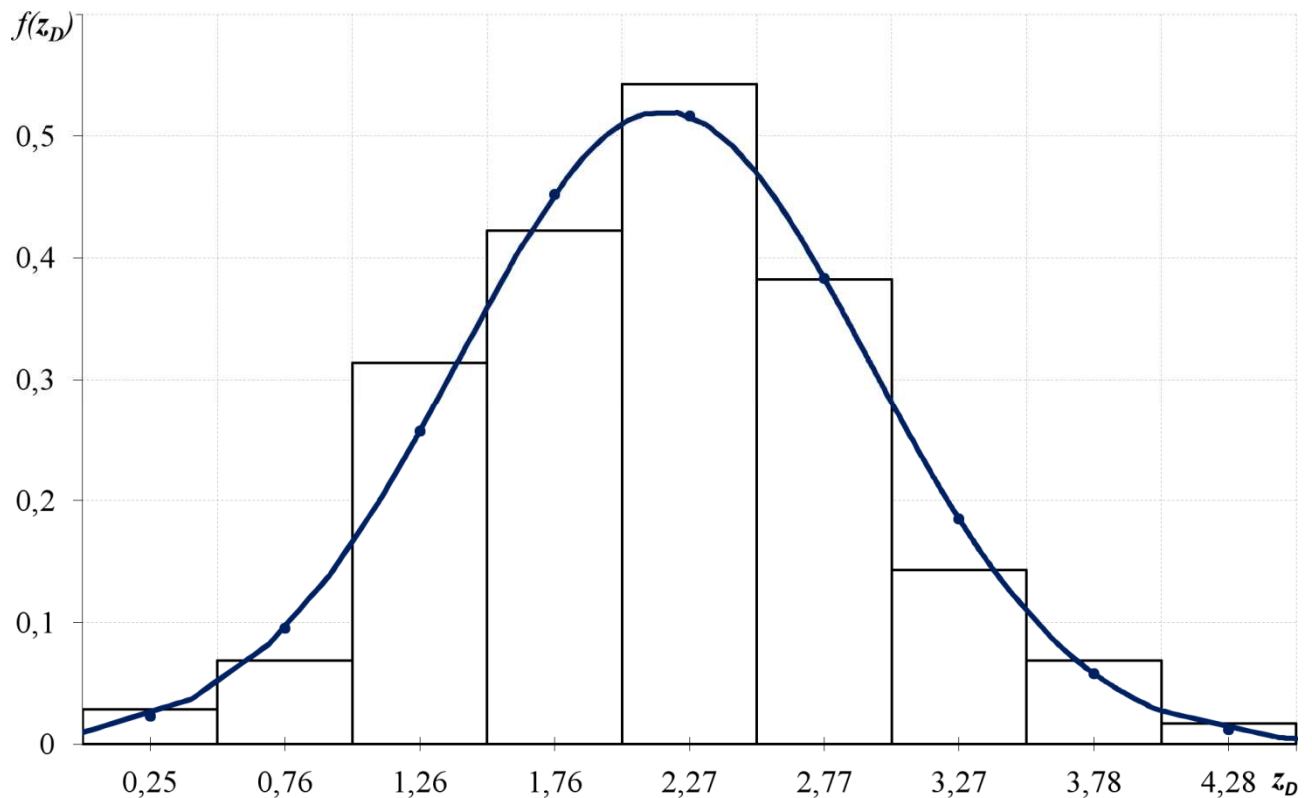


Рис. 3.3. Розподіл нормалізованих за перетворенням Бокса-Кокса значень тривалості проектів з розробки програмного забезпечення:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями

оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для нормалізації випадкової величини D було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення параметрів перетворення, знайдені в результаті рішення задачі (2.14), були такими: $\gamma_D = 15,905$, $\eta_D = 1,237$, $\phi_D = -0,106$ і $\lambda_D = 3352506$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = -0,001$, середньоквадратичне відхилення $S_z = 0,91$, асиметрія $\hat{A}_z = 0,017$, ексцес $\hat{\varepsilon}_z = 3,16$. Значення критерію Пірсона складає $\chi^2 = 6,53$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9-2-1=6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,026$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.3.

Результати нормалізації величини D за перетвореннями (3.1), (3.2), (2.15) та оцінки ймовірнісних характеристик величини z_D показують, що перетворення (3.2) та (2.17) дозволяють отримати випадкову величину z_D з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 12,59$ ($\nu = 9-2-1=6$, $\alpha = 0,05$). Для перетворення (3.1) значення критерію Пірсона χ^2 є більшим, ніж критичне значення $\chi_{кр}^2$, тому це перетворення неможна використовувати для нормалізації випадкової величини D .

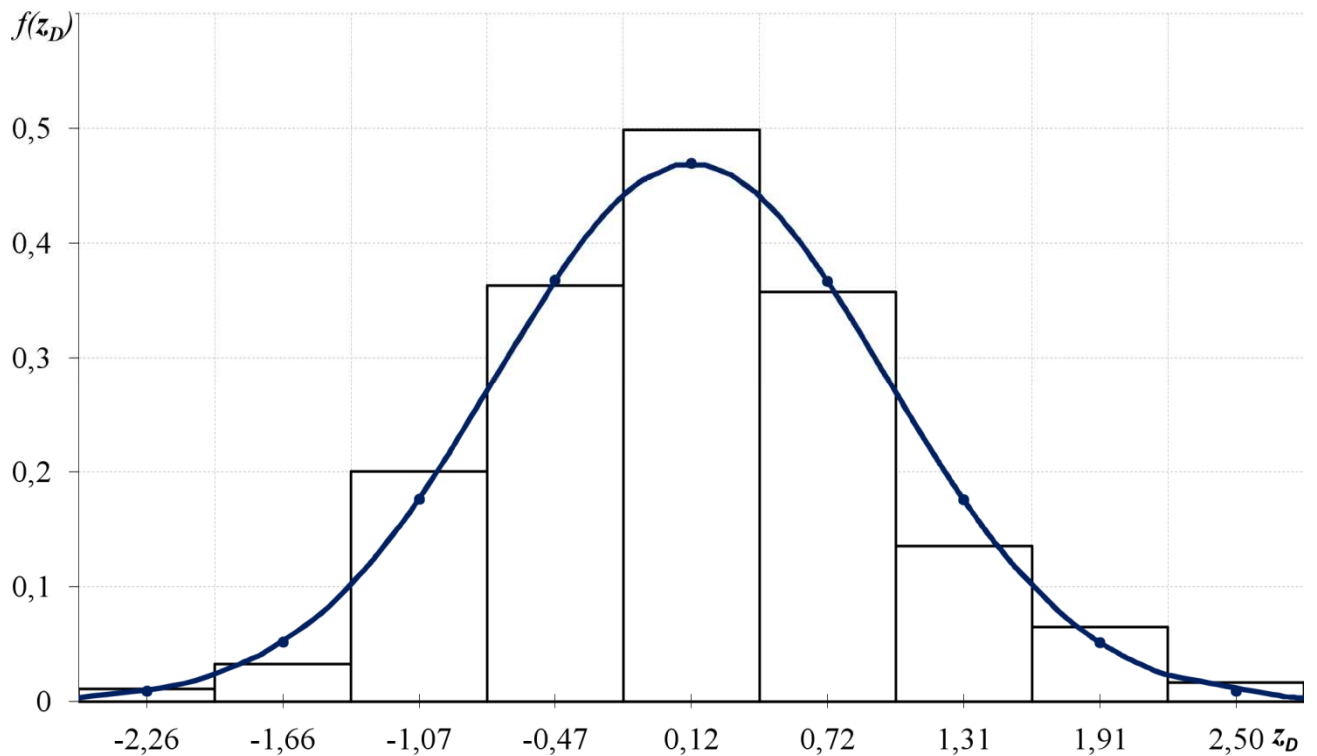


Рис. 3.4. Розподіл нормалізованих за перетворенням Джонсона значень тривалості проектів з розробки програмного забезпечення:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення використовуємо розраховану за (3.4) похибку δ . Серед перетворень (3.2) та (2.17) найменше значення похибки δ було отримане при нормалізації випадкової величини D , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.3.2 Вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи PC

Зробимо вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи РС.

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення для платформи РС з бази даних ISBSG [10, 11] (52 проекти). Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 1,41$, ексцес $\hat{\varepsilon}_D = 5,68$; математичне сподівання $\bar{D} = 9,26$, середньоквадратичне відхилення $S_D = 5,59$. Значення критерію Пірсона складає $\chi^2 = 2,81 \cdot 10^1$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$). Гістограму випадкової величини D приведено на рис. 3.5.

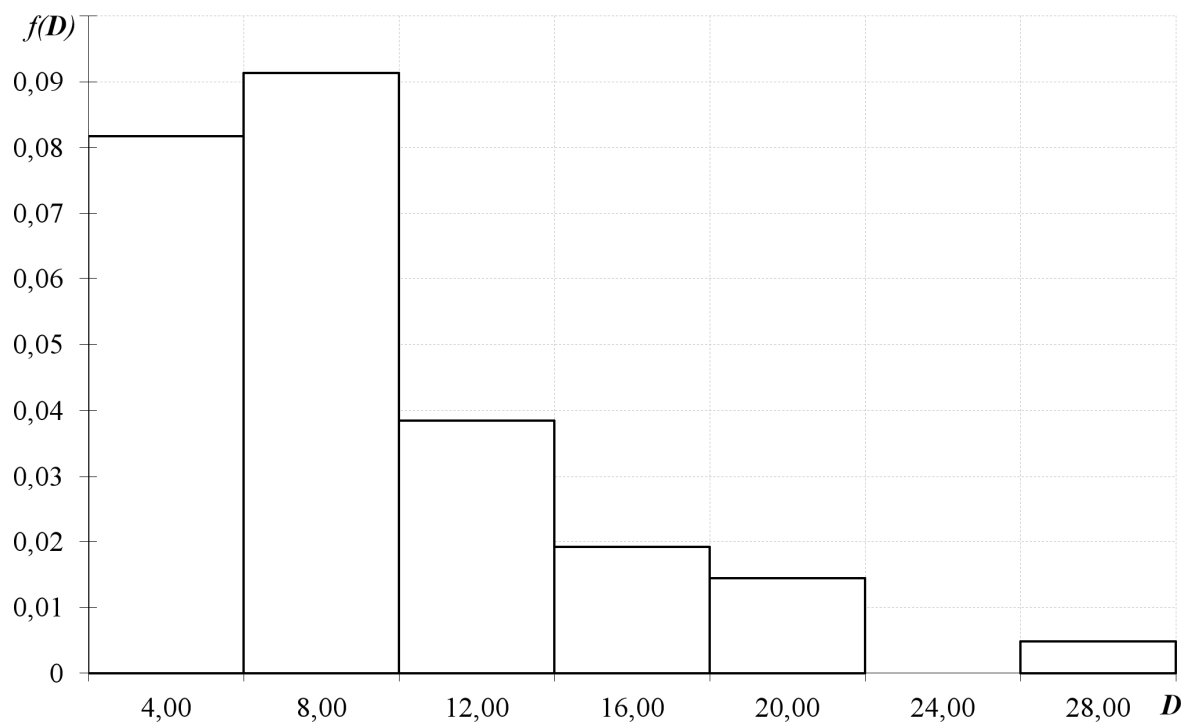


Рис. 3.5. Розподіл випадкової величини D – емпіричних значень тривалості проектів з розробки програмного забезпечення для платформи РС

Нормалізація випадкової величини D із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової

величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = 0,89$, середньоквадратичне відхилення $S_z = 0,26$, асиметрія $\hat{A}_z = -0,224$, ексцес $\hat{\varepsilon}_z = 2,87$. Значення критерію Пірсона складає $\chi^2 = 2,57$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,067$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.6.

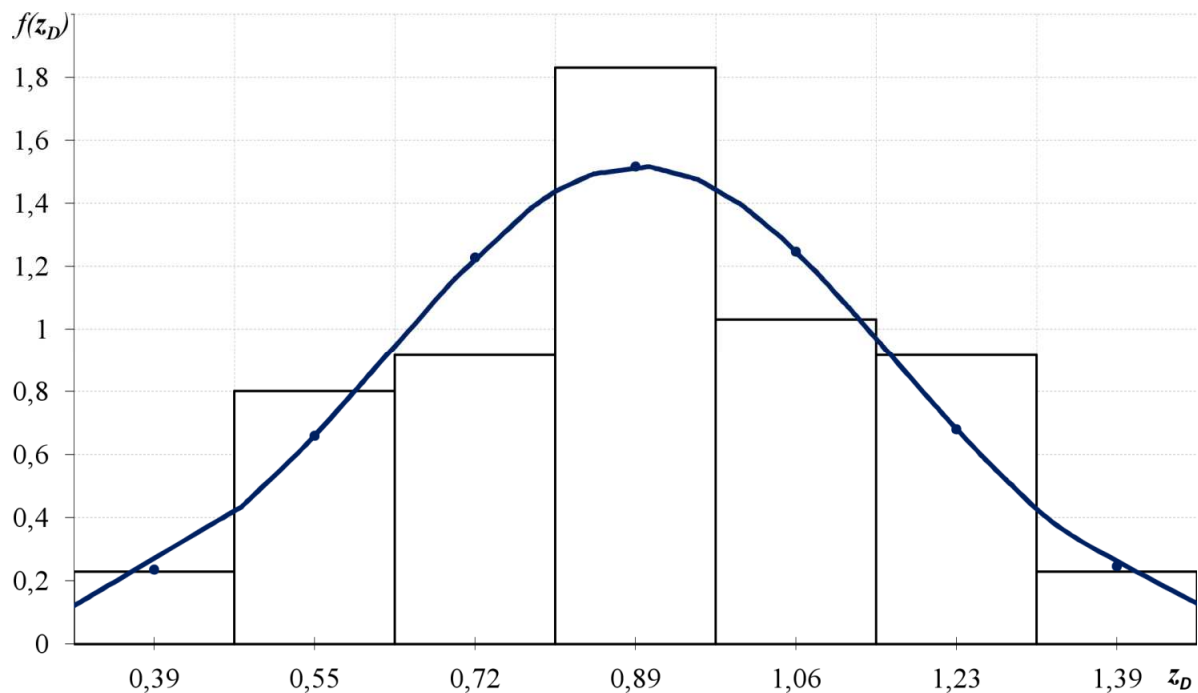


Рис. 3.6. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень тривалості проектів з розробки програмного забезпечення для платформи РС:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_D . Значення розрахованого за (3.3) параметру $\lambda = 0,14$. Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_t = 2,41$, середньоквадратичне

відхилення $S_z = 0,80$, асиметрія $\hat{A}_z = 0$, ексцес $\hat{\varepsilon}_z = 2,84$. Значення критерію Пірсона складає $\chi^2 = 7,29$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,025$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.7.

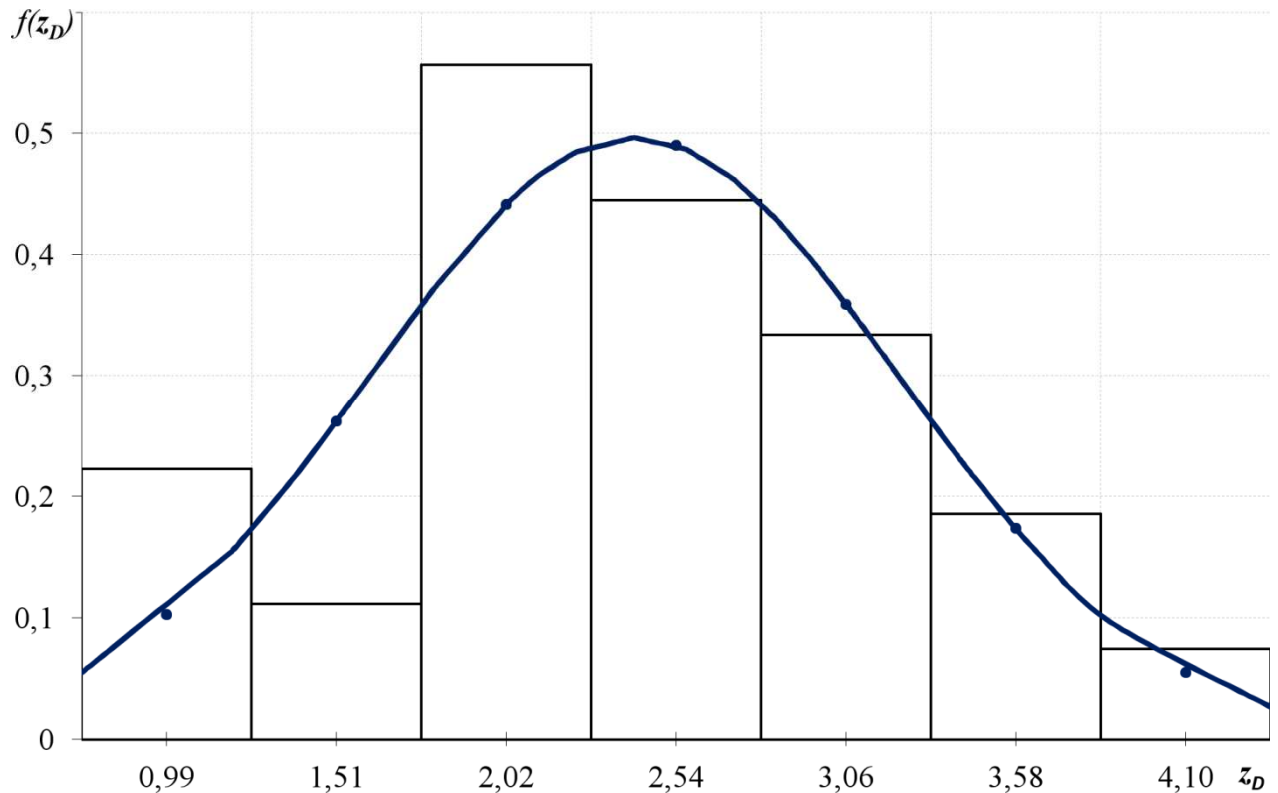


Рис. 3.7. Розподіл нормалізованих за перетворенням Бокса-Кокса значень тривалості проектів з розробки програмного забезпечення для платформи РС:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл.

Нормалізація випадкової величини D із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для нормалізації випадкової величини D було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення параметрів перетворення, знайдені в результаті рішення задачі (2.14) були такими: $\gamma_D = 2,38983$, $\eta_D = 1,30278$, $\phi_D = 0,43283$ і $\lambda_D = 54,7471$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = 0$, середньоквадратичне відхилення $S_z = 1,01$, асиметрія $\hat{A}_z = 0,00$, ексцес $\hat{\varepsilon}_z = 3,12$. Значення критерію Пірсона складає $\chi^2 = 7,85$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7 - 2 - 1 = 4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,016$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.8.

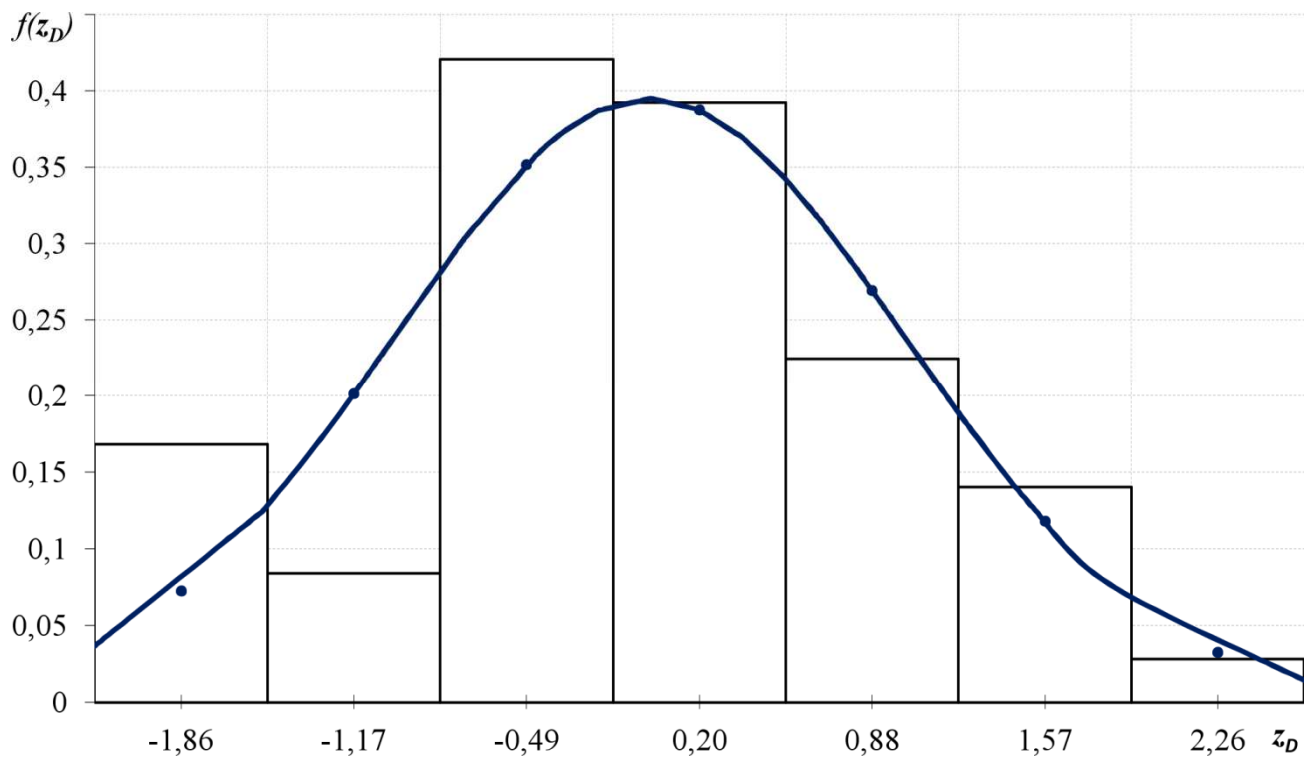


Рис. 3.8. Розподіл нормалізованих за перетворенням Джонсона значень тривалості проектів з розробки програмного забезпечення для платформи РС:

□ — емпіричний розподіл нормалізованої величини;

—●— — нормальний розподіл

Результати нормалізації величини D за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_D наведені в таблиці 3.1.

Таблиця 3.1 – Оцінки ймовірнісних характеристик випадкової величини z_D для платформи РС

Нормалізуюче перетворення	\bar{z}_D	S_z	\hat{A}_z	$\hat{\epsilon}_z$	χ^2	δ
Десятковий логарифм (3.1)	0,89	0,26	-0,224	2,87	2,57	0,067
Перетворення Бокса-Кокса (3.2)	2,41	0,80	0,00	2,84	7,29	0,025
Перетворення Джонсона S_B (2.17)	0,00	1,01	0,00	3,12	7,85	0,016

З таблиці 3.1 видно, що всі використані перетворення дозволяють отримати випадкову величину z_D з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 9,49$ ($\nu = 7-2-1=4$, $\alpha = 0,05$).

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи РС використовуємо розраховану за (3.4) похибку δ . Як видно з таблиці 3.1 серед перетворень (3.1), (3.2) та (2.17) найменше значення похибки δ було отримане при нормалізації випадкової величини D , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи РС краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.3.3 Вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mid-range

Зробимо вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mid-range.

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення для платформи mid-range з бази даних ISBSG [10, 11] (81 проект). Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 3,830$, ексцес $\hat{\varepsilon}_D = 23,90$; математичне сподівання $\bar{D} = 11,34$, середньоквадратичне відхилення $S_D = 10,45$. Значення критерію Пірсона складає $\chi^2 = 8,61 \cdot 10^5$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$). Гістограму випадкової величини D приведено на рис. 3.9.

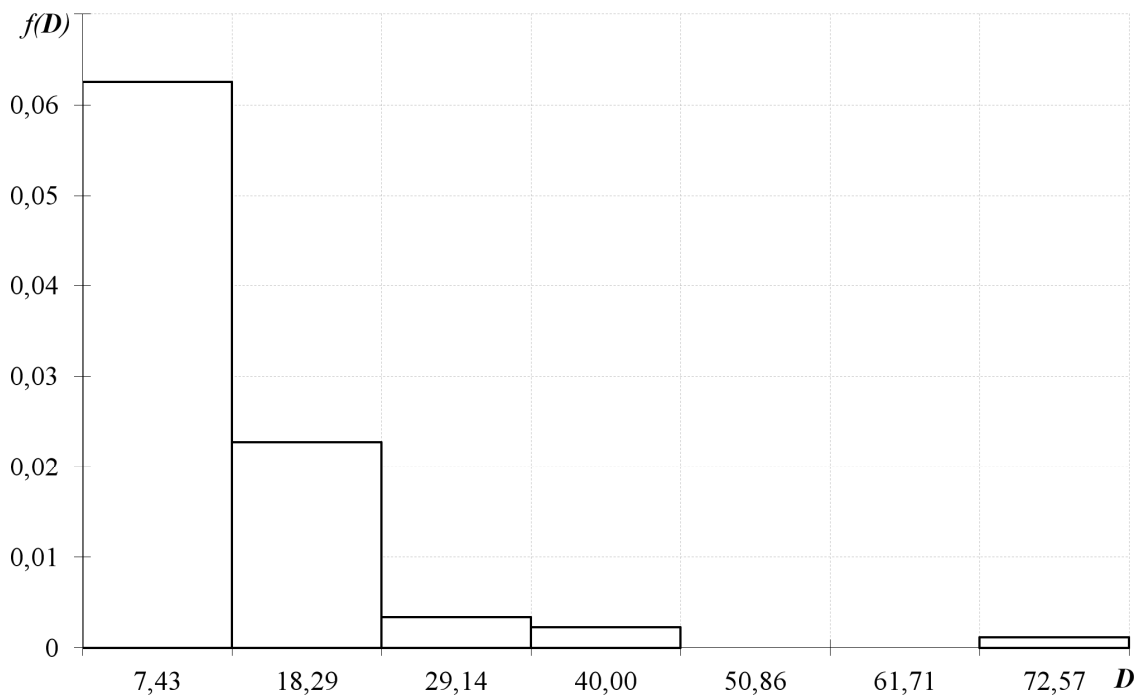


Рис. 3.9. Розподіл випадкової величини D – емпіричних значень тривалості проектів з розробки програмного забезпечення для платформи mid-range

Нормалізація випадкової величини D із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = 0,94$, середньоквадратичне відхилення $S_z = 0,31$, асиметрія $\hat{A}_z = 0,242$, ексцес $\hat{\varepsilon}_z = 3,19$. Значення критерію Пірсона складає $\chi^2 = 10,74$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,095$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.10.

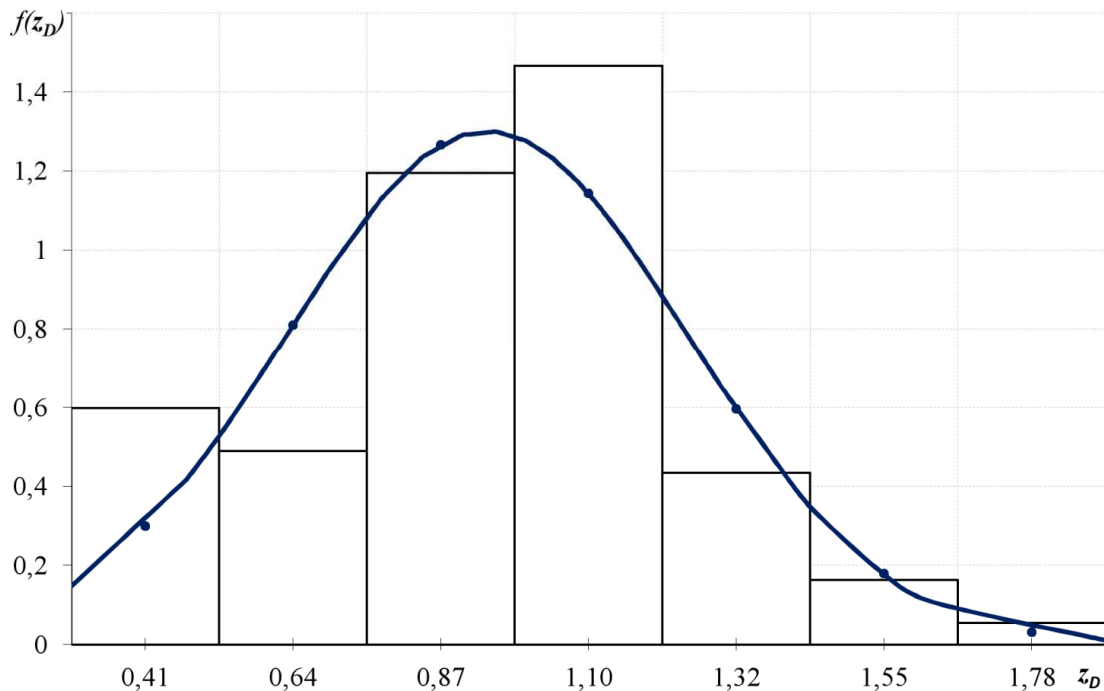


Рис. 3.10. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень тривалості проектів з розробки програмного забезпечення для платформи mid-range:

- ▭ – емпіричний розподіл нормалізованої величини;
- – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були

отримані значення нормалізованої випадкової величини z_D . Значення розрахованого за (3.3) параметру $\lambda = -0,11$. Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_t = 1,91$, середньоквадратичне відхилення $S_z = 0,55$, асиметрія $\hat{A}_z = 0,014$, ексцес $\hat{\varepsilon}_z = 2,86$. Значення критерію Пірсона складає $\chi^2 = 17,43$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7 - 2 - 1 = 4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,020$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.11.

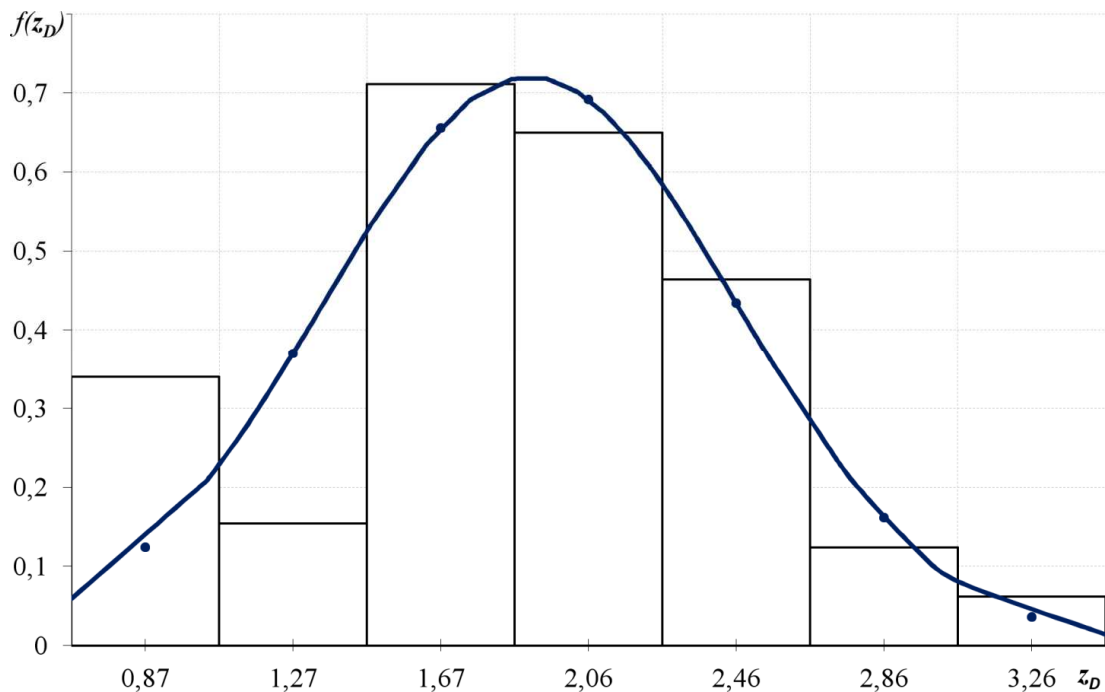


Рис. 3.11. Розподіл нормалізованих за перетворенням Бокса-Кокса значень тривалості проектів з розробки програмного забезпечення для платформи mid-range:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для нормалізації випадкової величини D було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення

параметрів перетворення, знайдені в результаті рішення задачі (2.14) були такими: $\gamma_D = 5,682$, $\eta_D = 1,13146$, $\varphi_D = 1,04977$ і $\lambda_D = 1166,2$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = -0,04$, середньоквадратичне відхилення $S_z = 0,95$, асиметрія $\hat{A}_z = -0,035$, ексцес $\hat{\varepsilon}_z = 3,05$. Значення критерію Пірсона складає $\chi^2 = 4,04$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,004$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.12.

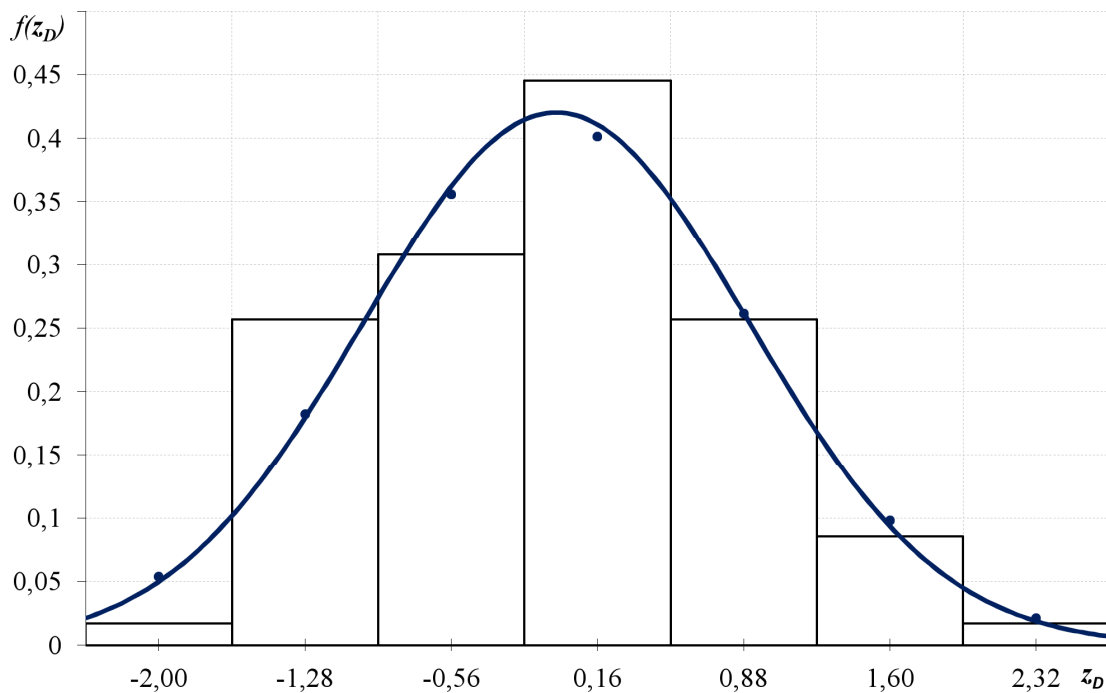


Рис. 3.12. Розподіл нормалізованих за перетворенням Джонсона значень тривалості проектів з розробки програмного забезпечення для платформи mid-range:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Результати нормалізації величини D за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_D наведені в таблиці 3.2.

Таблиця 3.2 – Оцінки ймовірнісних характеристик випадкової величини z_D для платформи mid-range

Нормалізуюче перетворення	\bar{z}_D	S_z	\hat{A}_z	$\hat{\varepsilon}_z$	χ^2	δ
Десятковий логарифм (3.1)	0,94	0,31	0,242	3,19	10,74	0,095
Перетворення Бокса-Кокса (3.2)	1,91	0,55	0,014	2,86	17,43	0,020
Перетворення Джонсона S_B (2.17)	-0,04	0,95	-0,035	3,05	4,04	0,004

З таблиці 3.2 видно, що перетворення (2.17) дозволяє отримати випадкову величину z_D з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 9,49$ ($\nu = 7-2-1=4$, $\alpha = 0,05$). Для перетворень (3.1) та (3.2) значення критерію Пірсона χ^2 є більшим, ніж критичне значення $\chi_{кр}^2$, тому ці перетворення неможна використовувати для нормалізації випадкової величини D . Отже, для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mid-range краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.3.4 Вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mainframe

Зробимо вибір нормалізуючого перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mainframe.

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення для платформи mainframe з бази даних ISBSG [10, 11] (215 проектів). Розподіл випадкової величини D відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини D значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_D = 2,743$, ексцес $\hat{\varepsilon}_D = 14,48$; математичне сподівання $\bar{D} = 11,78$, середньоквадратичне відхилення $S_D = 10,61$. Значення критерію Пірсона складає $\chi^2 = 8,66 \cdot 10^6$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$). Гістограму випадкової величини D приведено на рис. 3.13.

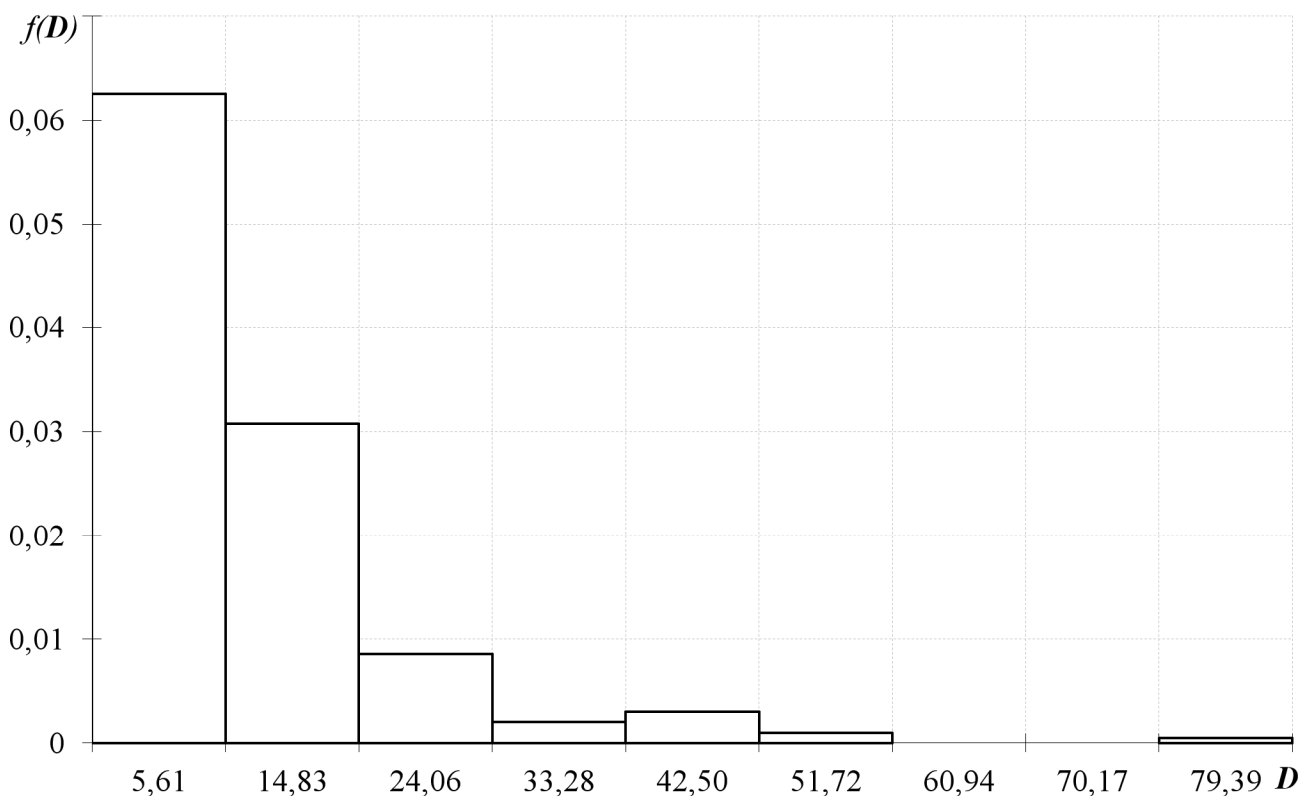


Рис. 3.13. Розподіл випадкової величини D – емпіричних значень тривалості проектів з розробки програмного забезпечення для платформи mainframe

Нормалізація випадкової величини D із використанням перетворення у вигляді десяткового логарифму. Використовуючи

перетворення (3.1) були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = 0,94$, середньоквадратичне відхилення $S_z = 0,35$, асиметрія $\hat{A}_z = -0,094$, ексцес $\hat{\varepsilon}_z = 3,10$. Значення критерію Пірсона складає $\chi^2 = 7,64$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,020$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.14.

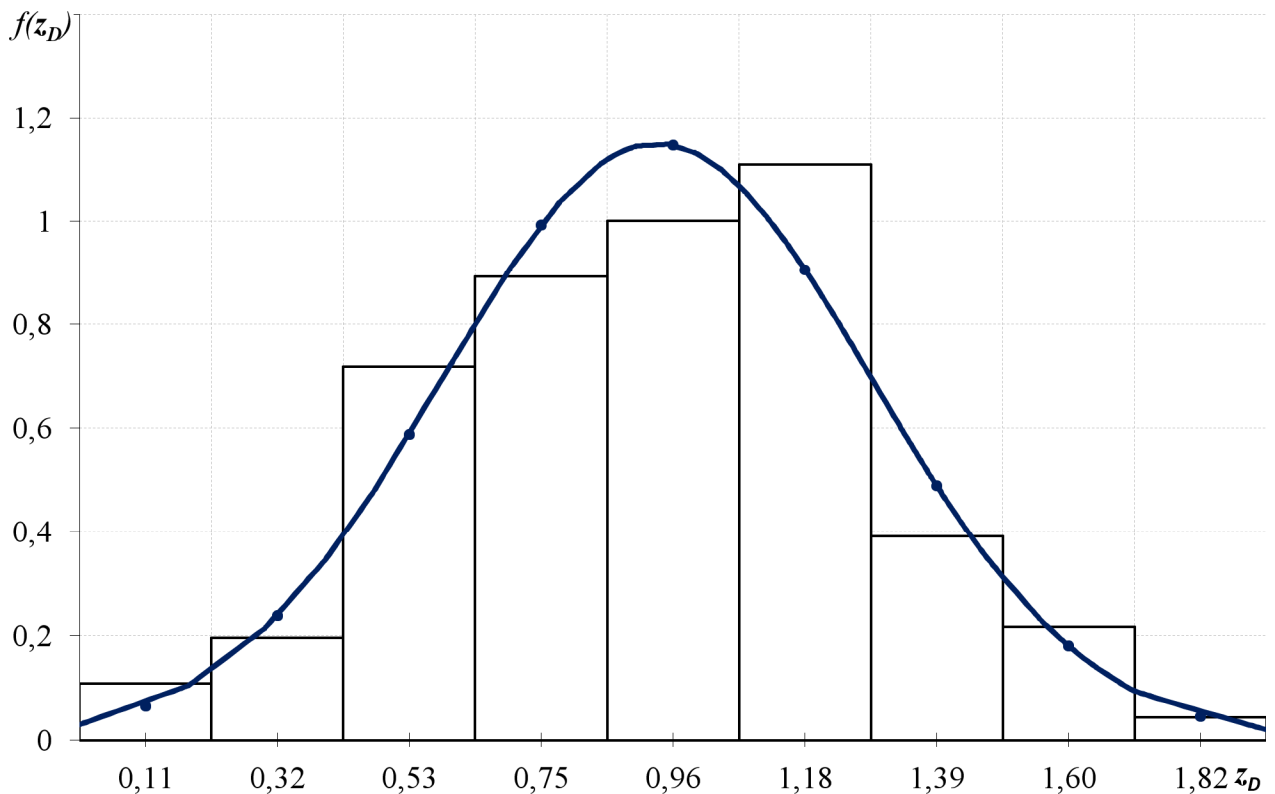


Рис. 3.14. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень тривалості проектів з розробки програмного забезпечення для платформи mainframe:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_D . Значення

розрахованого за (3.3) параметру $\lambda = 0,04$. Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_t = 2,27$, середньоквадратичне відхилення $S_z = 0,87$, асиметрія $\hat{A}_z = 0,005$, ексцес $\hat{\varepsilon}_z = 3,10$. Значення критерію Пірсона складає $\chi^2 = 6,29$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,009$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.15.

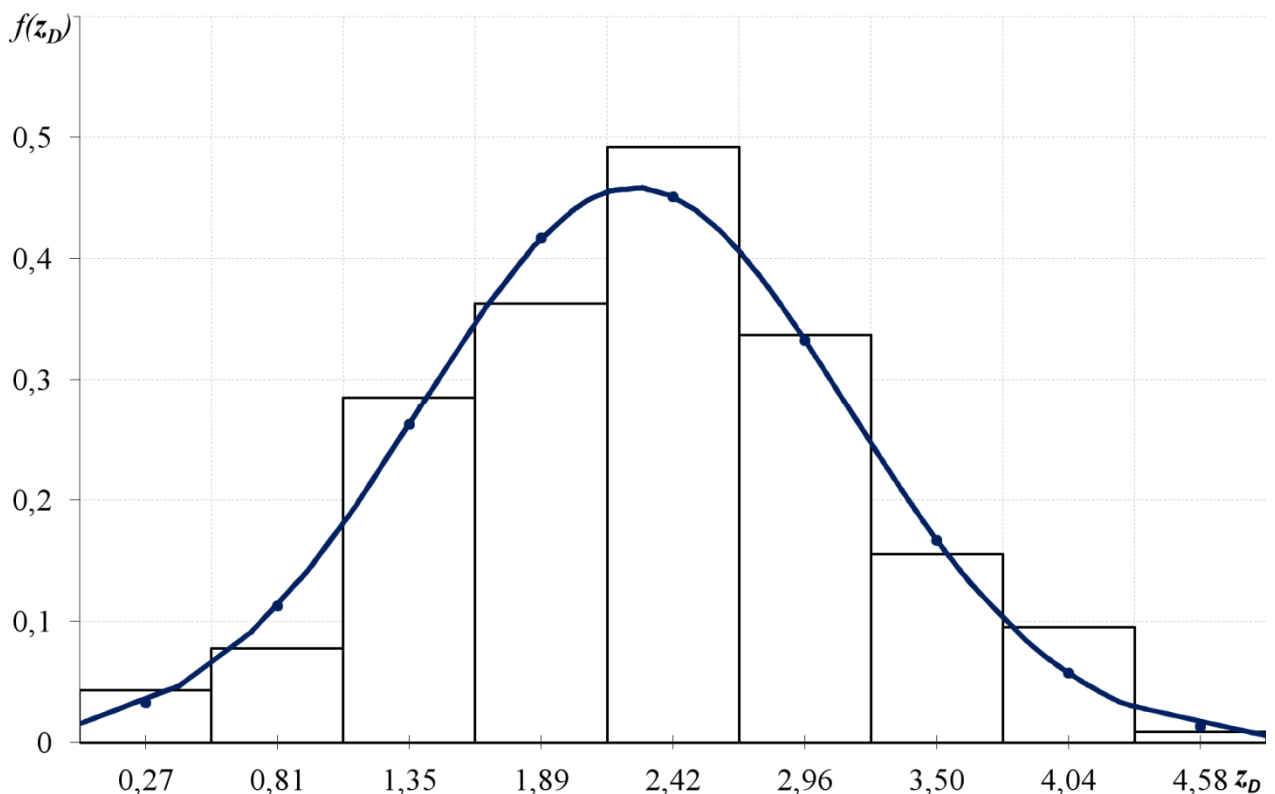


Рис. 3.15. Розподіл нормалізованих за перетворенням Бокса-Кокса значень тривалості проектів з розробки програмного забезпечення для платформи mainframe:

- – емпіричний розподіл нормалізованої величини;
- – нормальний розподіл

Нормалізація випадкової величини D із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_D^2 та ексцесу ε_D для нормалізації випадкової величини D було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення

параметрів перетворення, знайдені в результаті рішення задачі (2.14) були такими: $\gamma_D = 9,22106$, $\eta_D = 1,30531$, $\varphi_D = -0,25458$ і $\lambda_D = 10513,2$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_D . Оцінки ймовірнісних характеристик величини z_D такі: математичне сподівання $\bar{z}_D = 0$, середньоквадратичне відхилення $S_z = 1$, асиметрія $\hat{A}_z = 0$, ексцес $\hat{\varepsilon}_z = 3,03$. Значення критерію Пірсона складає $\chi^2 = 5,95$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,001$. Гістограма отриманої випадкової величини z_D та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_D)$ наведені на рис. 3.16.

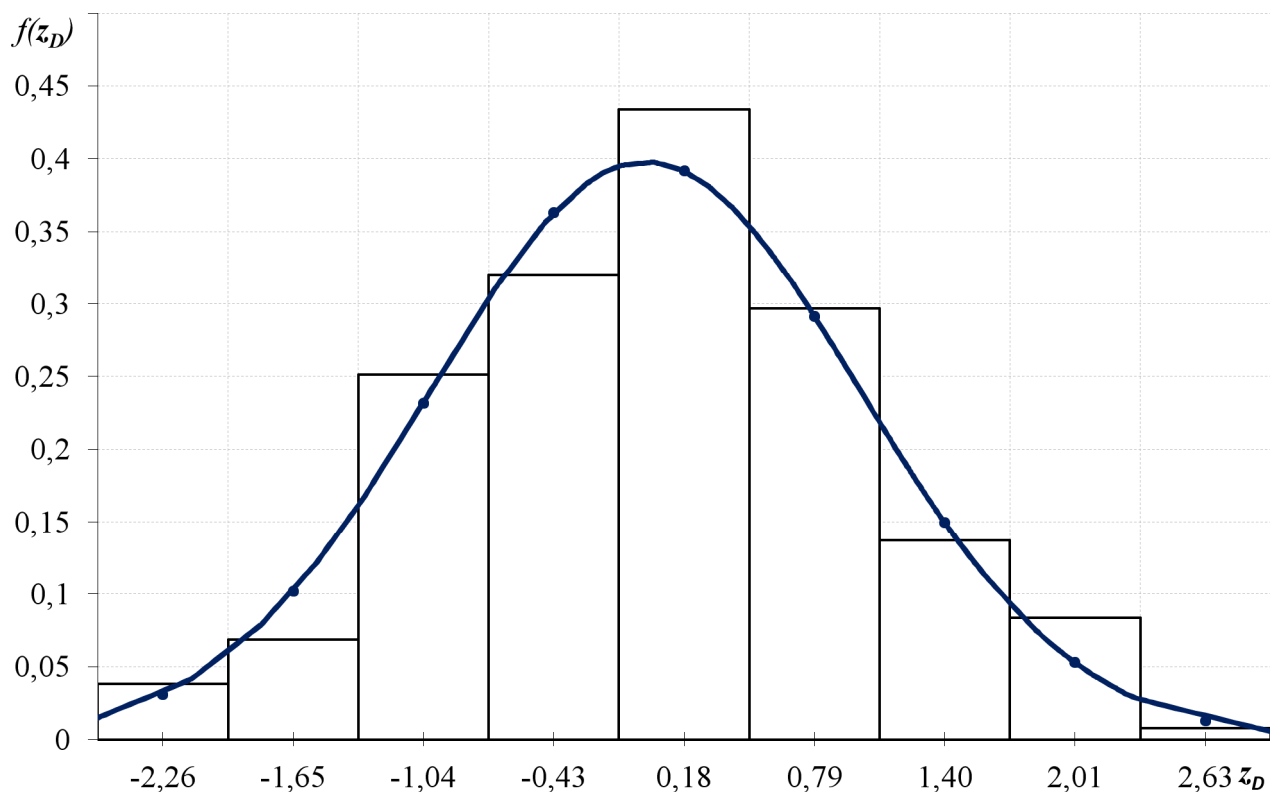


Рис. 3.16. Розподіл нормалізованих за перетворенням Джонсона значень тривалості проектів з розробки програмного забезпечення для платформи mainframe:

▭ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Результати нормалізації величини D за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_D наведені в таблиці 3.3.

Таблиця 3.3 – Оцінки ймовірнісних характеристик випадкової величини z_D для платформи mainframe

Нормалізуюче перетворення	\bar{z}_D	S_z	\hat{A}_z	$\hat{\epsilon}_z$	χ^2	δ
Десятковий логарифм (3.1)	0,94	0,35	-0,094	3,10	7,64	0,020
Перетворення Бокса-Кокса (3.2)	2,27	0,87	0,005	3,10	6,29	0,009
Перетворення Джонсона S_B (2.17)	0,00	1,00	0,000	3,03	5,95	0,001

З таблиці 3.3 видно, що всі використані перетворення дозволяють отримати випадкову величину z_D з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 12,59$ ($\nu = 9-2-1=6$, $\alpha = 0,05$).

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mainframe використовуємо розраховану за (3.4) похибку δ . Як видно з таблиці 3.3 серед перетворень (3.1), (3.2) та (2.17) найменше значення похибки δ було отримане при нормалізації випадкової величини D , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення для платформи mainframe краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.4 Вибір нормалізуючого перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення

3.4.1 Вибір нормалізуючого перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для всіх платформ

Нехай випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe з бази даних ISBSG [10, 11] (348 проектів). Розподіл випадкової величини E відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини E значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_E = 4,87$, ексцес $\hat{\epsilon}_E = 34,66$; математичне сподівання $\bar{E} = 5933,79$, середньоквадратичне відхилення $S_E = 11810,19$. Значення критерію Пірсона складає $\chi^2 = 2,52 \cdot 10^{12}$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$). Гістограму випадкової величини E приведено на рис. 3.17.

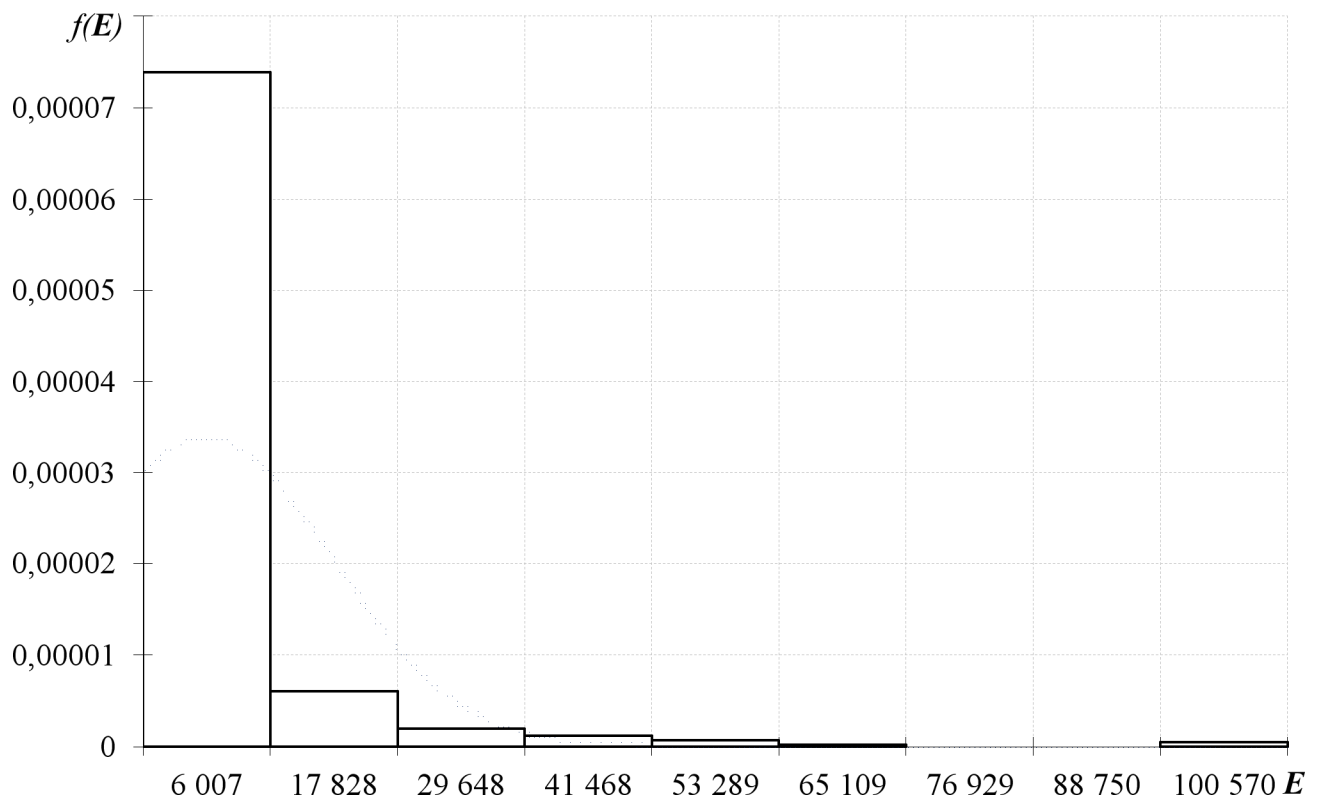


Рис. 3.17. Розподіл випадкової величини E – емпіричних значень трудомісткості проектів з розробки програмного забезпечення

Необхідно визначити яке з наведених в нормалізуючих перетворень дозволить краще виконати нормалізацію випадкової величини E .

Нормалізація випадкової величини E із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 3,34$, середньоквадратичне відхилення $S_z = 0,60$, асиметрія $\hat{A}_z = 0,269$, ексцес $\hat{\varepsilon}_z = 2,71$. Значення критерію Пірсона складає $\chi^2 = 12,13$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,159$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.18.

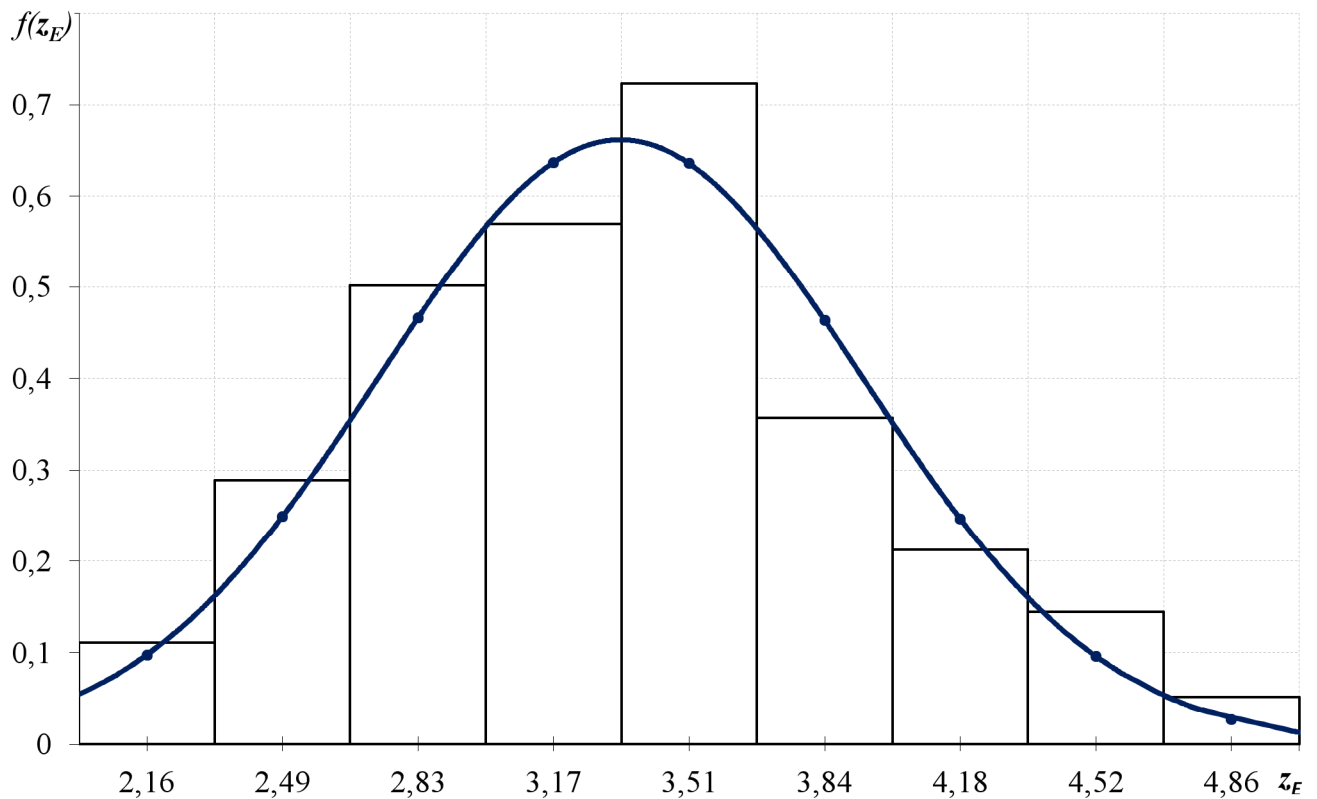


Рис. 3.18. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень трудомісткості проектів з розробки програмного забезпечення:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_E . Значення розрахованого за (3.3) параметру $\lambda = 0,01$. Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 5,70$, середньоквадратичне відхилення $S_z = 0,75$, асиметрія $\hat{A}_z = 0,002$, ексцес $\hat{\varepsilon}_z = 2,59$. Значення критерію Пірсона складає $\chi^2 = 7,31$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,171$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.19.

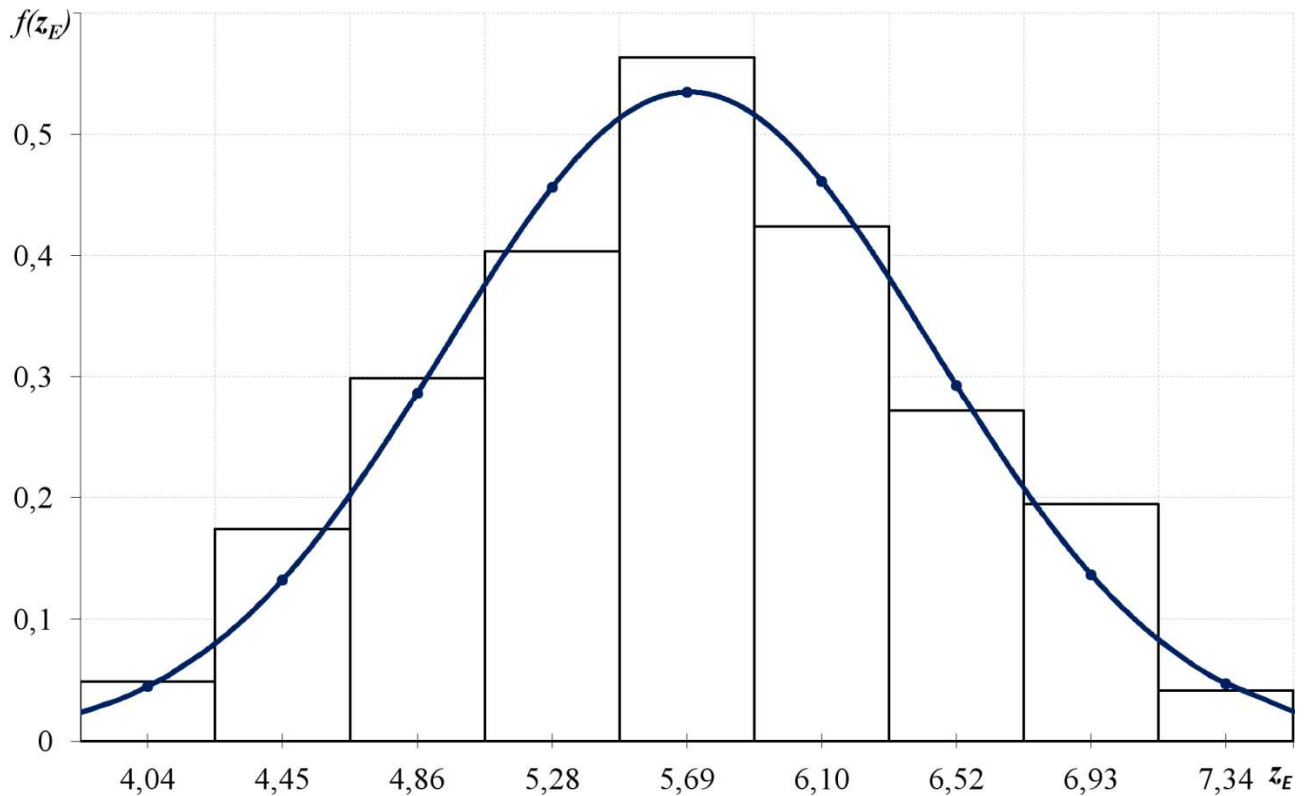


Рис. 3.19. Розподіл нормалізованих за перетворенням Бокса-Кокса значень трудомісткості проектів з розробки програмного забезпечення:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_E^2 та ексцесу ε_E для нормалізації випадкової величини E було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення параметрів перетворення, знайдені в результаті рішення задачі (2.14), були такими: $\gamma_E = 3,80913$, $\eta_E = 0,65406$, $\varphi_E = 82,20693$ і $\lambda_E = 667093$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 0$, середньоквадратичне відхилення $S_z = 1$, асиметрія $\hat{A}_z = 0$, ексцес $\hat{\varepsilon}_z = 3,02$. Значення критерію Пірсона складає $\chi^2 = 4,21$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0$. Гістограма

отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.3.

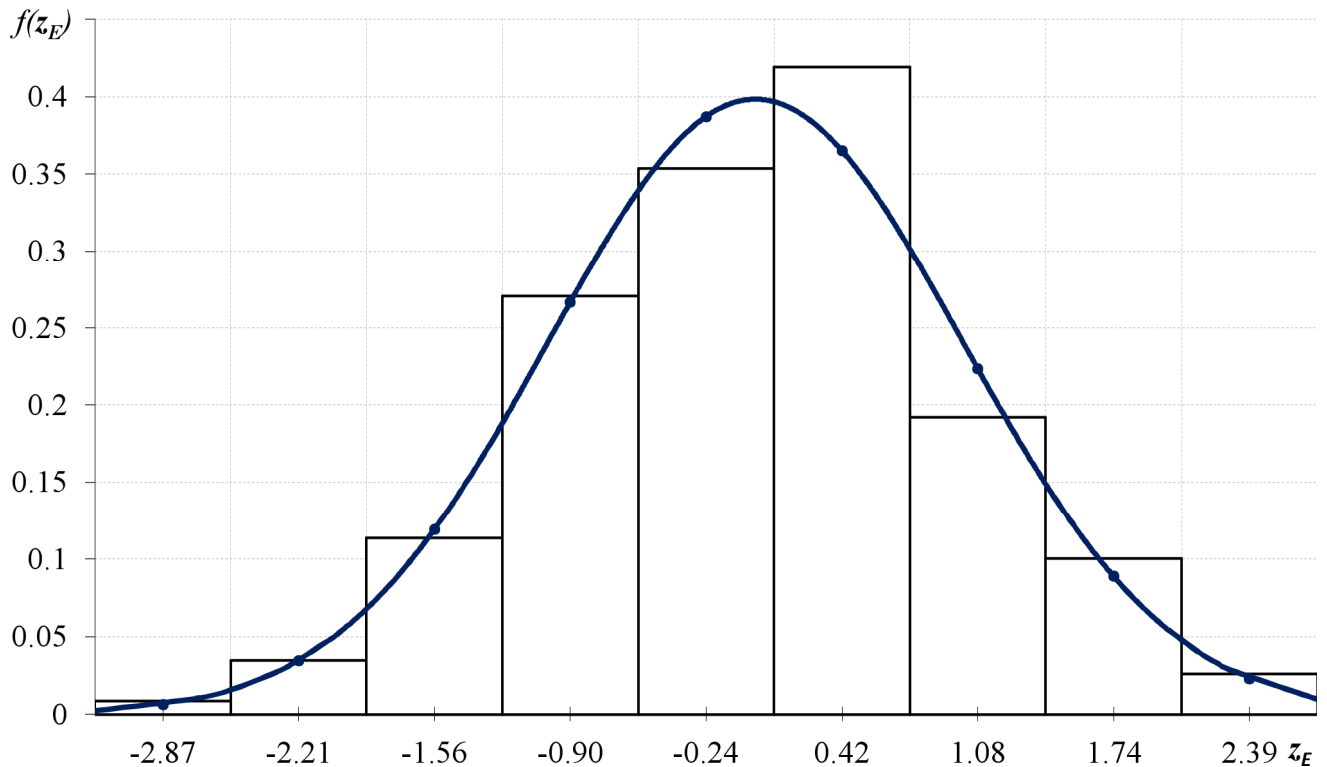


Рис. 3.20. Розподіл нормалізованих за перетворенням Джонсона значень трудомісткості проектів з розробки програмного забезпечення:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Результати нормалізації величини E за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_E показують, що всі використані перетворення дозволяють отримати випадкову величину z_E з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 12,59$ ($\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$).

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення використовуємо розраховану за (3.4) похибку δ . Серед перетворень (3.1), (3.2) та (2.17) найменше значення похибки

δ було отримане при нормалізації випадкової величини E , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.4.2 Вибір нормалізуючого перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи РС

Нехай випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформи РС з бази даних ISBSG [10, 11] (52 проекти). Розподіл випадкової величини E відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини E значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_E = 2,468$, ексцес $\hat{\varepsilon}_E = 9,44$; математичне сподівання $\bar{E} = 2348,77$, середньоквадратичне відхилення $S_E = 3232,77$. Значення критерію Пірсона складає $\chi^2 = 4,75 \cdot 10^{12}$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$). Гістограму випадкової величини E приведено на рис. 3.21.

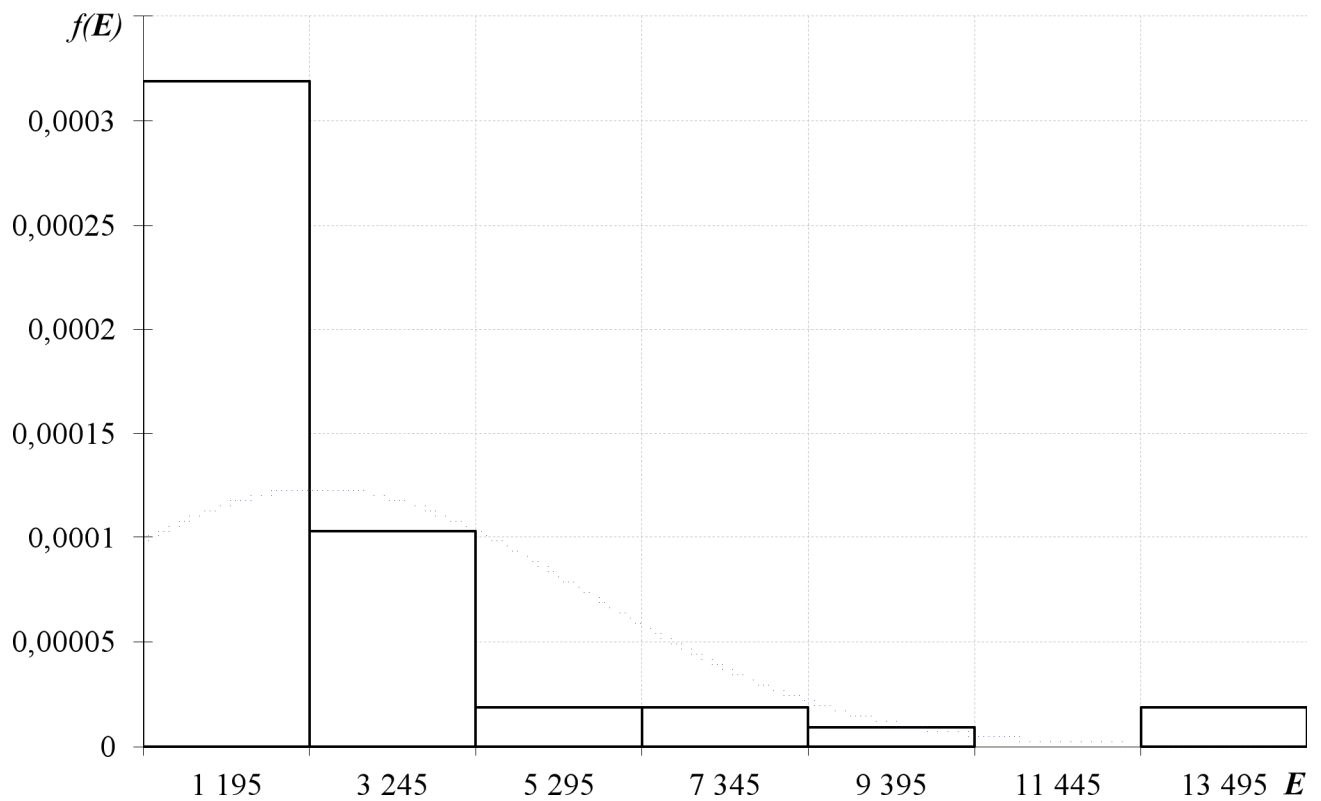


Рис. 3.21. Розподіл випадкової величини E – значень трудомісткості проектів з розробки програмного забезпечення для платформи РС

Необхідно визначити яке з наведених в нормалізуючих перетворень дозволить краще виконати нормалізацію випадкової величини E .

Нормалізація випадкової величини E із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 3,06$, середньоквадратичне відхилення $S_z = 0,52$, асиметрія $\hat{A}_z = 0,330$, ексцес $\hat{\varepsilon}_z = 2,28$. Значення критерію Пірсона складає $\chi^2 = 11,26$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,632$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.22.

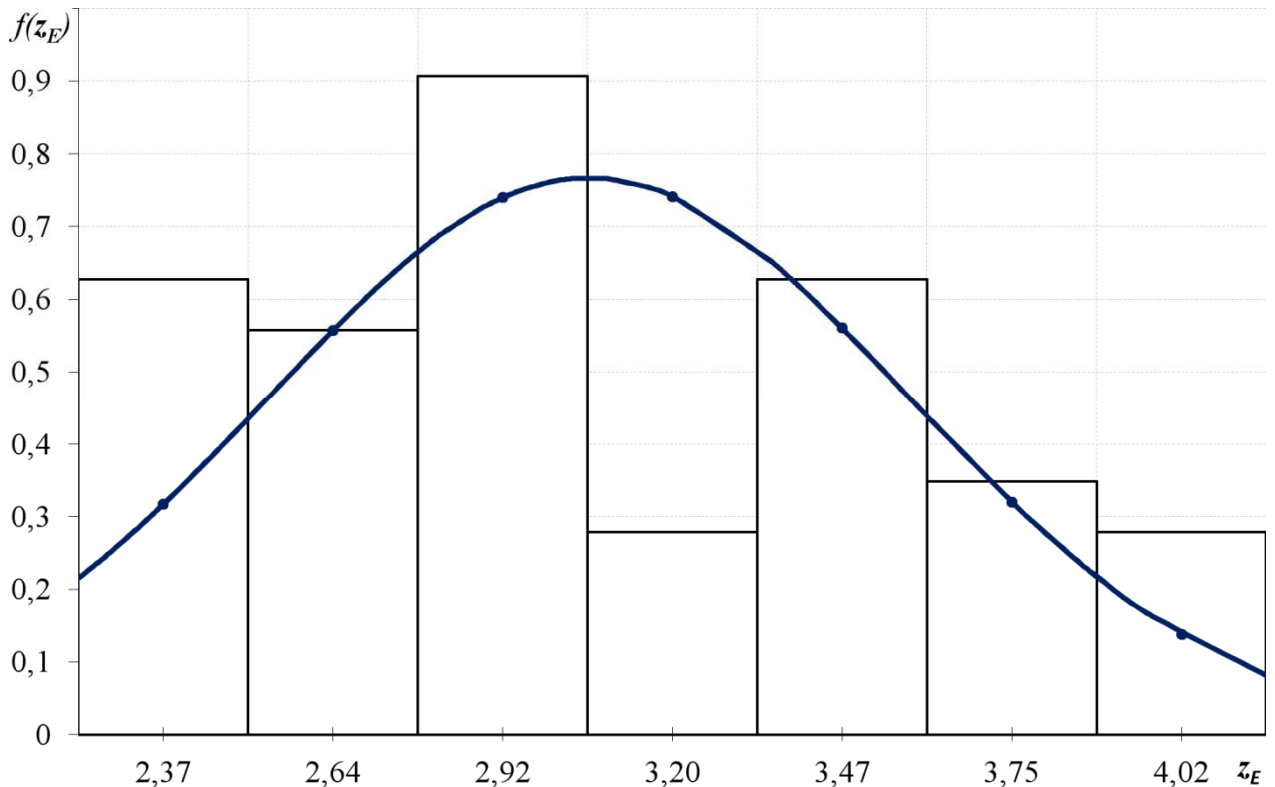


Рис. 3.22. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень трудомісткості проектів з розробки програмного забезпечення для платформи РС:

□ – емпіричний розподіл нормалізованої величини;
 —●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_E . Значення розрахованого за (3.3) параметру $\lambda = -0,14$. Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 4,44$, середньоквадратичне відхилення $S_z = 0,44$, асиметрія $\hat{A}_z = 0,045$, ексцес $\hat{\varepsilon}_z = 2,14$. Значення критерію Пірсона складає $\chi^2 = 7,61$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,747$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.23.

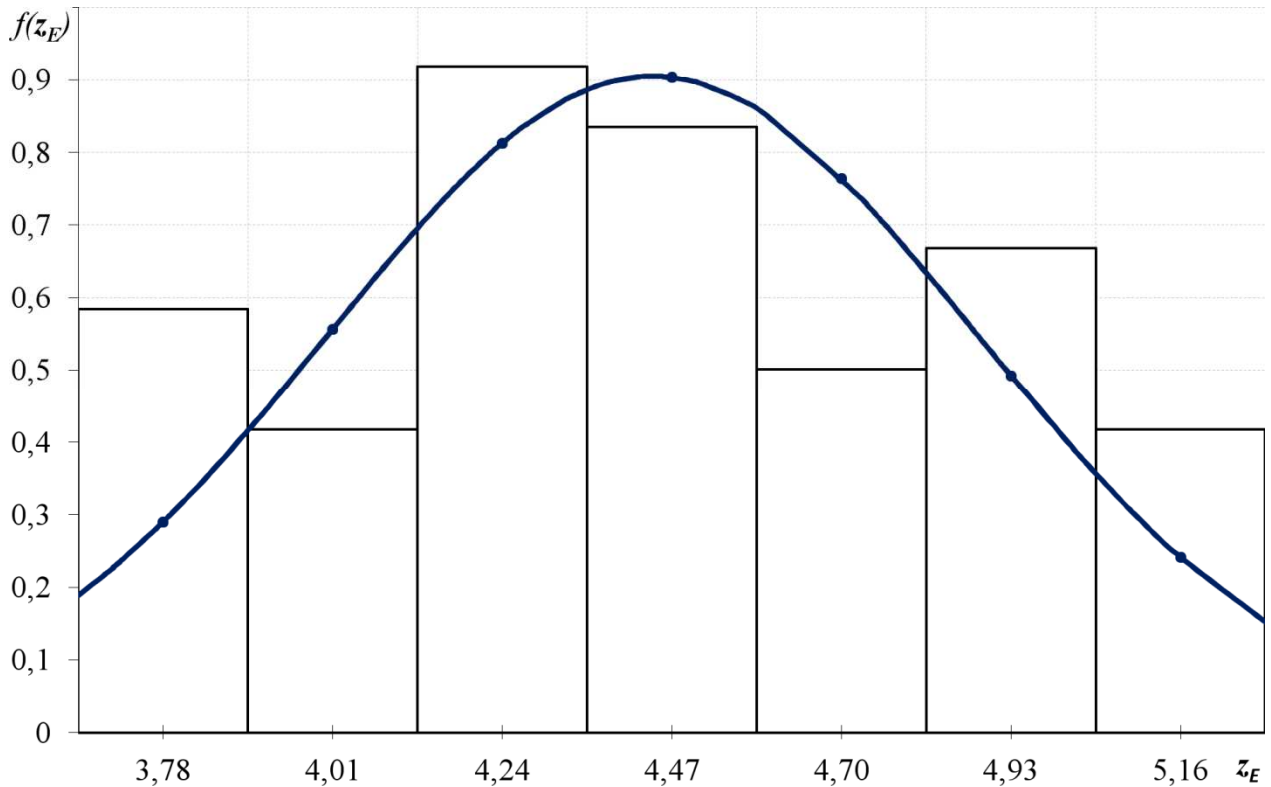


Рис. 3.23. Розподіл нормалізованих за перетворенням Бокса-Кокса значень трудомісткості проектів з розробки програмного забезпечення для платформи РС:

□ – емпіричний розподіл нормалізованої величини;
 —●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_E^2 та ексцесу ε_E для нормалізації випадкової величини E було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення параметрів перетворення, знайдені в результаті рішення задачі (2.14), були такими: $\gamma_E = 1,61212$, $\eta_E = 0,53883$, $\phi_E = 158,85317$ і $\lambda_E = 18248,7$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 0$, середньоквадратичне відхилення $S_z = 1,01$, асиметрія $\hat{A}_z = 0$, ексцес $\hat{\varepsilon}_z = 3,12$.

Значення критерію Пірсона складає $\chi^2 = 2,09$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7 - 2 - 1 = 4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,016$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.24.

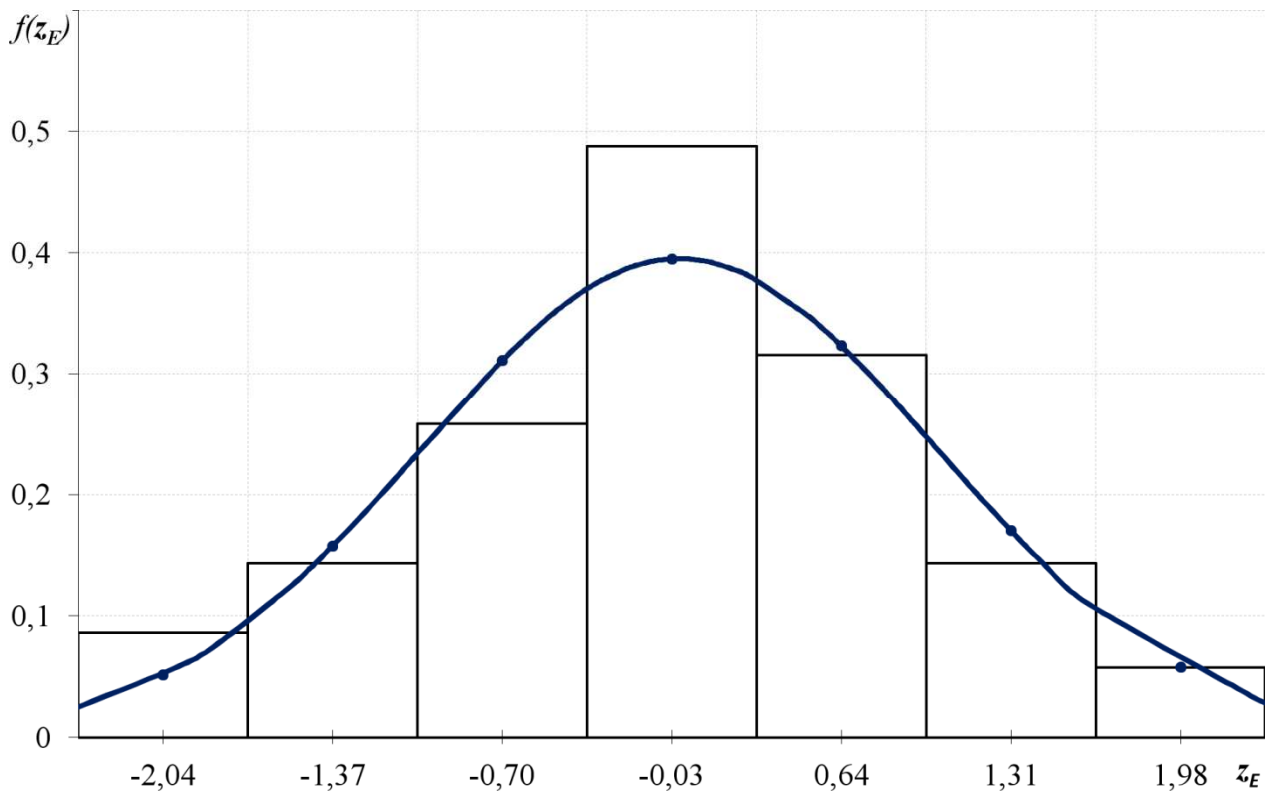


Рис. 3.24. Розподіл нормалізованих за перетворенням Джонсона значень трудомісткості проектів з розробки програмного забезпечення для платформи РС:

- – емпіричний розподіл нормалізованої величини;
- – нормальний розподіл

Результати нормалізації величини E за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_E наведені в таблиці 3.4.

Таблиця 3.4 – Оцінки ймовірнісних характеристик випадкової величини z_E

Нормалізуюче перетворення	\bar{z}_E	S_z	\hat{A}_z	$\hat{\varepsilon}_z$	χ^2	δ
Десятковий логарифм (3.1)	3,06	0,52	0,330	2,28	11,26	0,632
Перетворення Бокса-Кокса (3.2)	4,44	0,44	0,045	2,14	7,61	0,747
Перетворення Джонсона S_B (2.17)	0,00	1,01	0,000	3,12	2,09	0,016

З таблиці 3.4 видно, що перетворення (3.2) та (2.17) дозволяють отримати випадкову величину z_E з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 9,49$, ($\nu = 7-2-1=4$, $\alpha = 0,05$). ж

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи РС використовуємо розраховану за (3.4) похибку δ . Як видно з таблиці 3.4 серед перетворень (3.2) та (2.17) найменше значення похибки δ було отримане при нормалізації випадкової величини E , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи РС краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.4.3 Вибір нормалізуючого перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи mid-range

Нехай випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформи mid-range з бази даних ISBSG [10, 11] (81 проект). Розподіл випадкової величини E відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини E значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_E = 5,738$, ексцес $\hat{\varepsilon}_E = 43,95$; математичне сподівання $\bar{E} = 6949,21$, середньоквадратичне відхилення $S_E = 13199,69$. Значення критерію Пірсона складає $\chi^2 = 9,21 \cdot 10^8$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$). Гістограму випадкової величини E приведено на рис. 3.25.

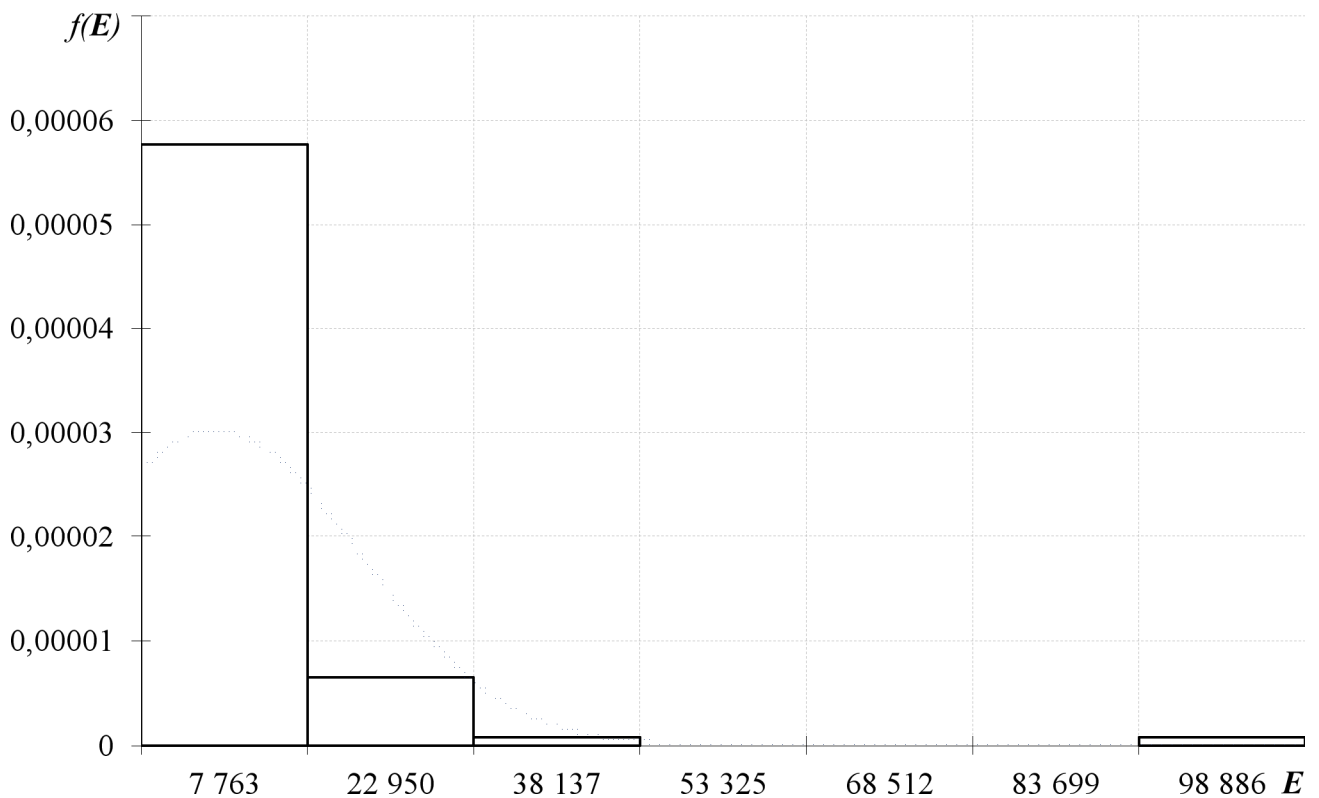


Рис. 3.25. Розподіл випадкової величини E – значень трудомісткості проектів з розробки програмного забезпечення для платформи mid-range

Необхідно визначити яке з наведених в нормалізуючих перетворень дозволить краще виконати нормалізацію випадкової величини E .

Нормалізація випадкової величини E із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової

величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 3,51$, середньоквадратичне відхилення $S_z = 0,53$, асиметрія $\hat{A}_z = 0,221$, ексцес $\hat{\varepsilon}_z = 3,06$. Значення критерію Пірсона складає $\chi^2 = 1,94$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,053$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.26.

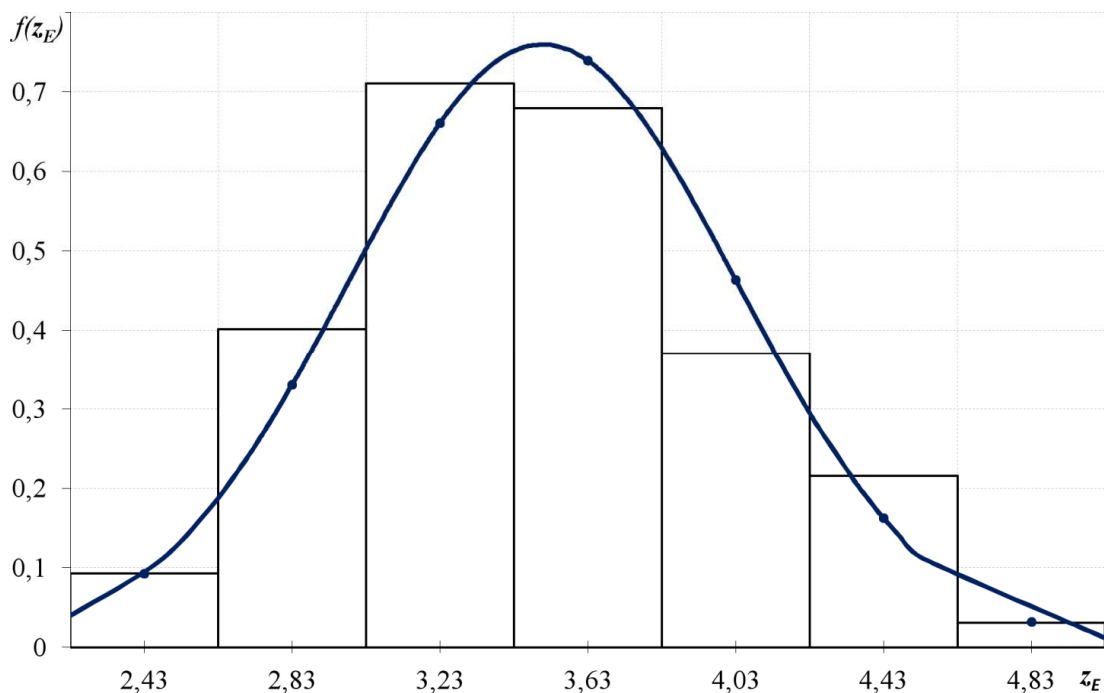


Рис. 3.26. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень трудомісткості проектів з розробки програмного забезпечення для платформи mid-range:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_E . Значення розрахованого за (3.3) параметру $\lambda = -0,06$. Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 6,37$, середньоквадратичне

відхилення $S_z = 0,44$, асиметрія $\hat{A}_z = 0,091$, ексцес $\hat{\varepsilon}_z = 3,35$. Значення критерію Пірсона складає $\chi^2 = 2,3$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7-2-1=4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,132$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.27.

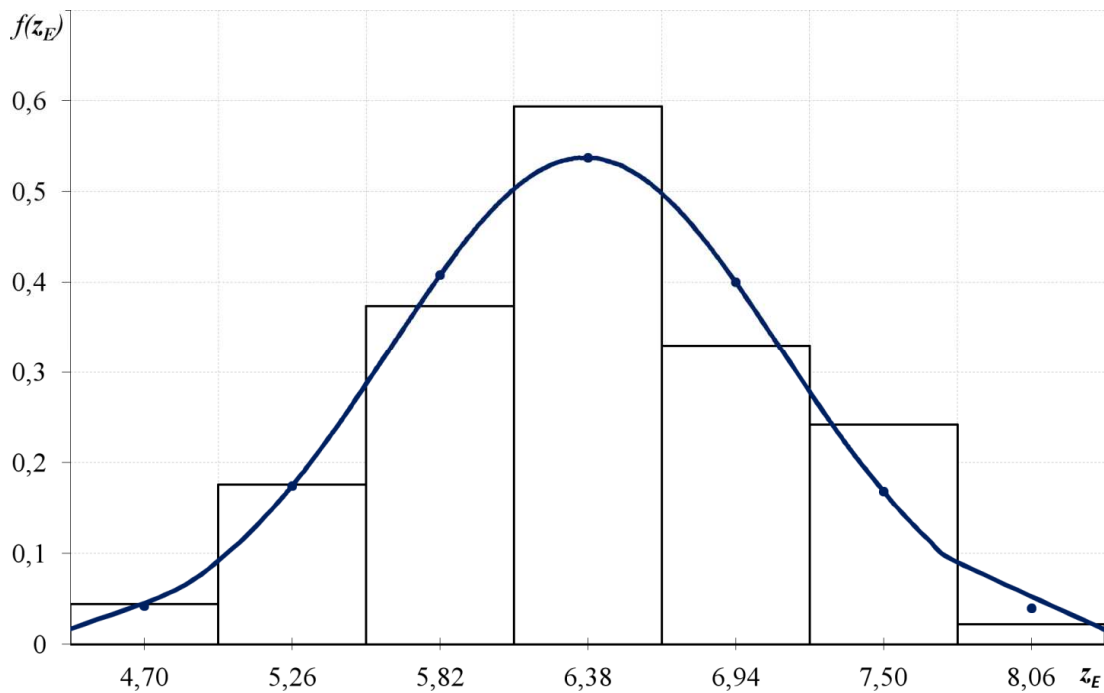


Рис. 3.27. Розподіл нормалізованих за перетворенням Бокса-Кокса значень трудомісткості проектів з розробки програмного забезпечення для платформи mid-range:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_E^2 та ексцесу ε_E для нормалізації випадкової величини E було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення параметрів перетворення, знайдені в результаті рішення задачі (2.14), були такими: $\gamma_E = 4,0686$, $\eta_E = 0,77997$, $\phi_E = 82,51282$ і $\lambda_E = 565242$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 0$, середньоквадратичне відхилення $S_z = 1,01$, асиметрія $\hat{A}_z = 0,009$, ексцес $\hat{\varepsilon}_z = 2,98$. Значення критерію Пірсона складає $\chi^2 = 2,8$ (критичне значення $\chi_{кр}^2 = 9,49$, $\nu = 7 - 2 - 1 = 4$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.28.

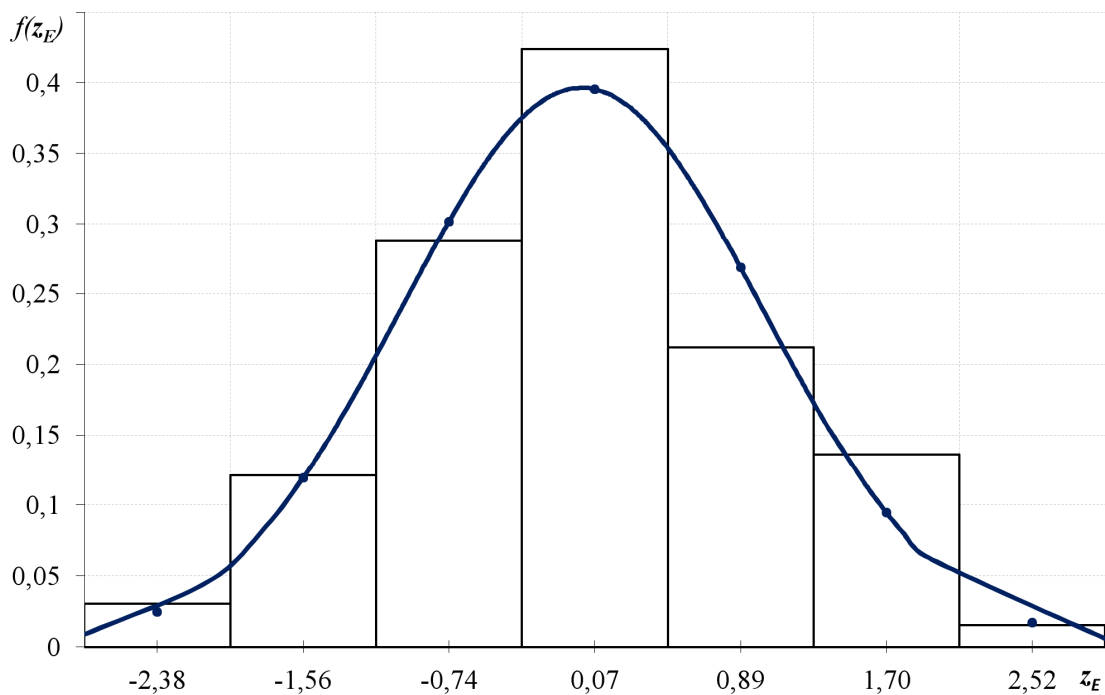


Рис. 3.28. Розподіл нормалізованих за перетворенням Джонсона значень трудомісткості проектів з розробки програмного забезпечення для платформи mid-range:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Результати нормалізації величини E за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_E наведені в таблиці 3.5.

Таблиця 3.5 – Оцінки ймовірнісних характеристик випадкової величини z_E

Нормалізуюче перетворення	\bar{z}_E	S_z	\hat{A}_z	$\hat{\varepsilon}_z$	χ^2	δ
Десятковий логарифм (3.1)	3,51	0,53	0,221	3,06	1,94	0,053
Перетворення Бокса-Кокса (3.2)	6,37	0,74	0,091	3,35	2,3	0,132
Перетворення Джонсона S_B (2.17)	0,00	1,01	0,009	2,98	2,8	0,000

З таблиці 3.5 видно, що всі використані перетворення дозволяють отримати випадкову величину z_E з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 9,49$, ($\nu = 7-2-1=4$, $\alpha = 0,05$).

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи mid-range використовуємо розраховану за (3.4) похибку δ . Як видно з таблиці 3.5 серед перетворень (3.1), (3.2) та (2.17) найменше значення похибки δ було отримане при нормалізації випадкової величини E , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи mid-range краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.4.4 Вибір нормалізуючого перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи mainframe

Нехай випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформи mainframe з бази даних ISBSG [10, 11] (215 проектів). Розподіл випадкової величини E відрізняється від нормального закону розподілу, асиметрія та ексцес розподілу величини E значно відрізняються від відповідних характеристик нормального розподілу: асиметрія $\hat{A}_E = 4,038$, ексцес $\hat{\varepsilon}_E = 24,7$; математичне сподівання $\bar{E} = 6418,31$, середньоквадратичне відхилення $S_E = 12441,63$. Значення критерію Пірсона складає $\chi^2 = 1,23 \cdot 10^{10}$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$). Гістограму випадкової величини E приведено на рис. 3.29.

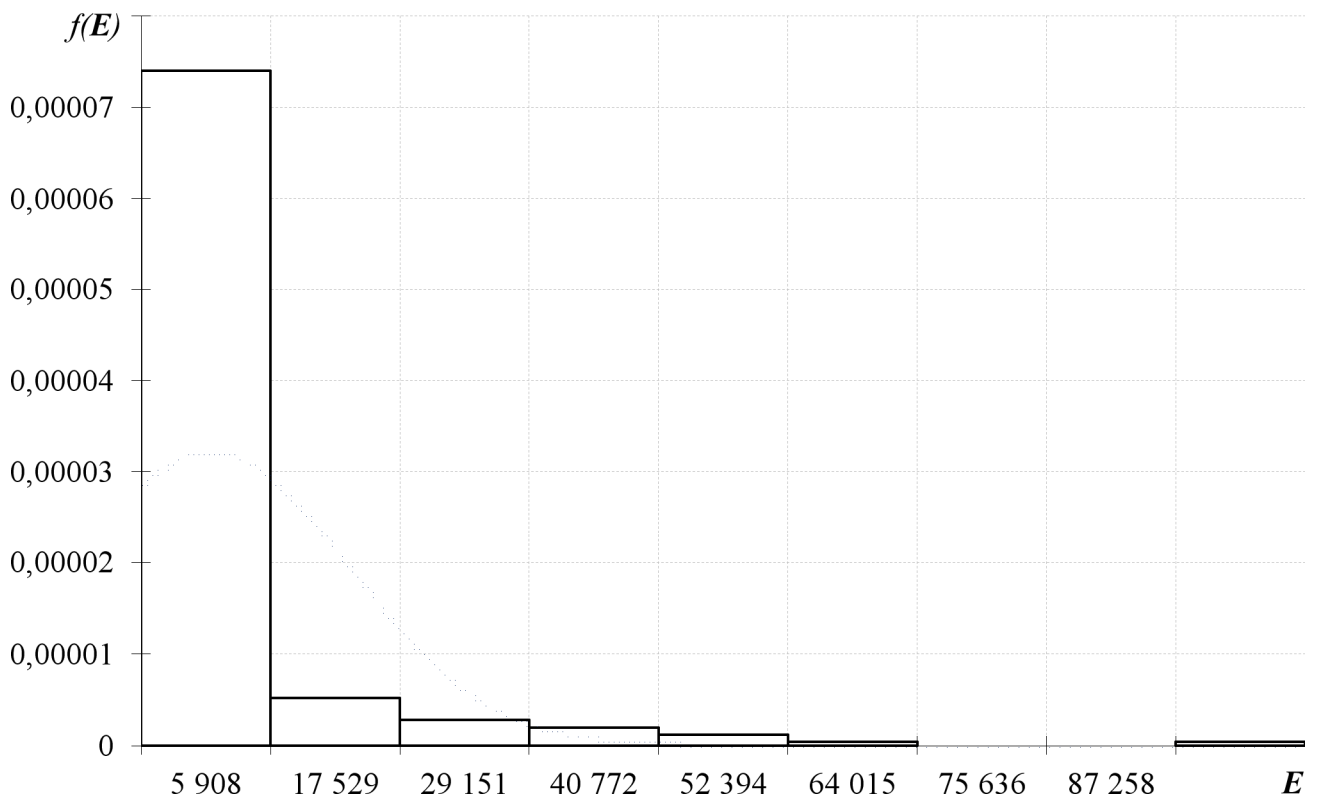


Рис. 3.29. Розподіл випадкової величини E – значень трудомісткості проектів з розробки програмного забезпечення для платформи mainframe

Необхідно визначити яке з наведених в нормалізуючих перетворень дозволить краще виконати нормалізацію випадкової величини E .

Нормалізація випадкової величини E із використанням перетворення у вигляді десяткового логарифму. Використовуючи перетворення (3.1) були отримані значення нормалізованої випадкової

величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 3,34$, середньоквадратичне відхилення $S_z = 0,63$, асиметрія $\hat{A}_z = 0,301$, ексцес $\hat{\varepsilon}_z = 2,68$. Значення критерію Пірсона складає $\chi^2 = 13,34$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,196$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.30.

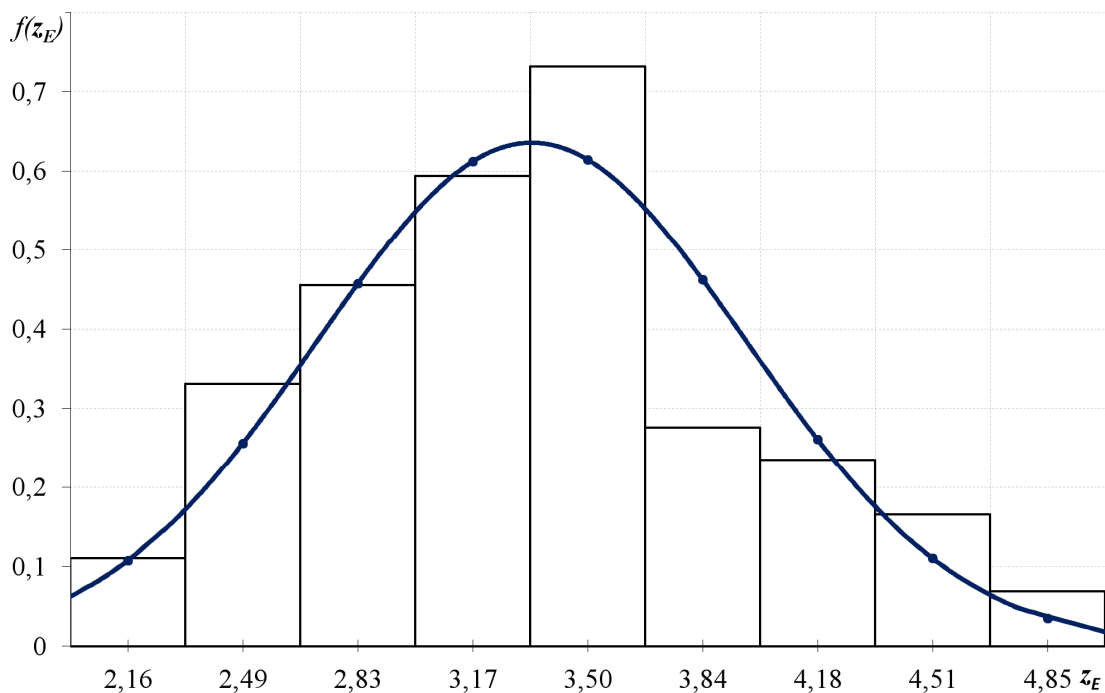


Рис. 3.30. Розподіл нормалізованих за перетворенням у вигляді десяткового логарифму значень трудомісткості проектів з розробки програмного забезпечення для платформи mainframe:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Бокса-Кокса. Використовуючи перетворення (3.2) були отримані значення нормалізованої випадкової величини z_E . Значення розрахованого за (3.3) параметру $\lambda = -0,08$. Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 5,70$, середньоквадратичне

відхилення $S_z = 0,77$, асиметрія $\hat{A}_z = 0,030$, ексцес $\hat{\varepsilon}_z = 2,58$. Значення критерію Пірсона складає $\chi^2 = 5,81$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,178$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.31.

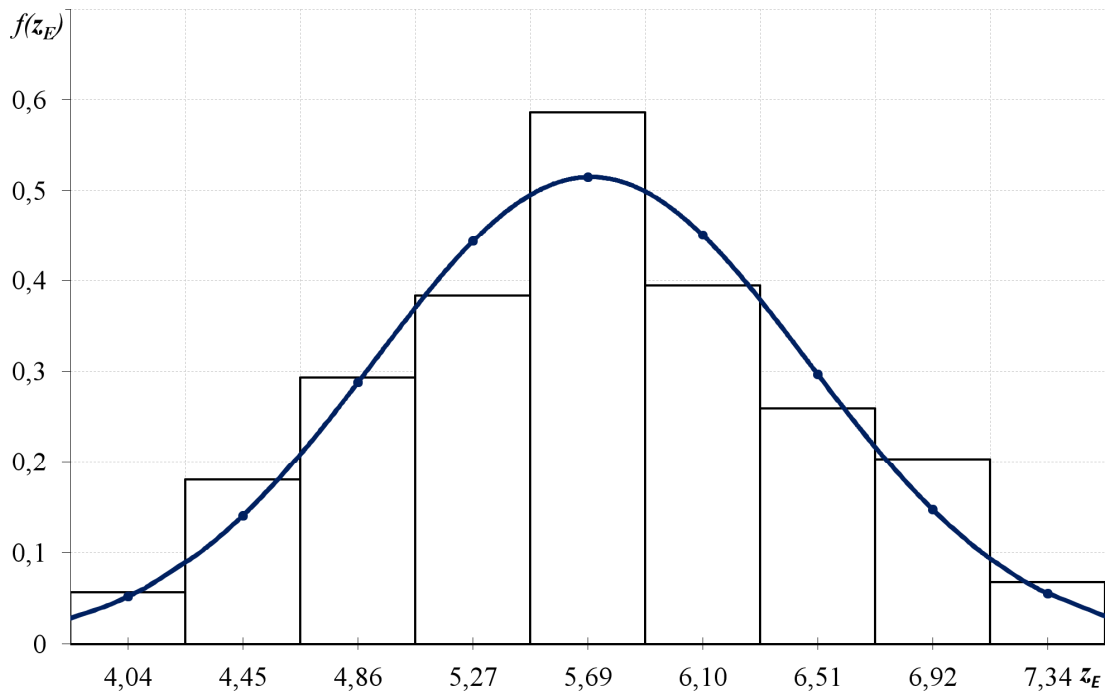


Рис. 3.31. Розподіл нормалізованих за перетворенням Бокса-Кокса значень трудомісткості проектів з розробки програмного забезпечення для платформи mainframe:

□ – емпіричний розподіл нормалізованої величини;

—●— – нормальний розподіл

Нормалізація випадкової величини E із використанням перетворення Джонсона. По діаграмі Джонсона (рис. 2.2) за значеннями оцінок асиметрії у квадраті A_E^2 та ексцесу ε_E для нормалізації випадкової величини E було вибрано перетворення Джонсона сім'ї S_B (2.17). Значення параметрів перетворення, знайдені в результаті рішення задачі (2.14), були такими: $\gamma_E = 3,73121$, $\eta_E = 0,62765$, $\phi_E = 81,46192$ і $\lambda_E = 753187$.

Використовуючи перетворення (2.17) з вказаними вище параметрами були отримані значення нормалізованої випадкової величини z_E . Оцінки ймовірнісних характеристик величини z_E такі: математичне сподівання $\bar{z}_E = 0$, середньоквадратичне відхилення $S_z = 1$, асиметрія $\hat{A}_z = 0$, ексцес $\hat{\varepsilon}_z = 3,03$. Значення критерію Пірсона складає $\chi^2 = 4,34$ (критичне значення $\chi_{кр}^2 = 12,59$, $\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$), значення розрахованої за (3.4) похибки $\delta = 0,001$. Гістограма отриманої випадкової величини z_E та її теоретичний розподіл у вигляді щільності ймовірності нормального розподілу $f(z_E)$ наведені на рис. 3.32.

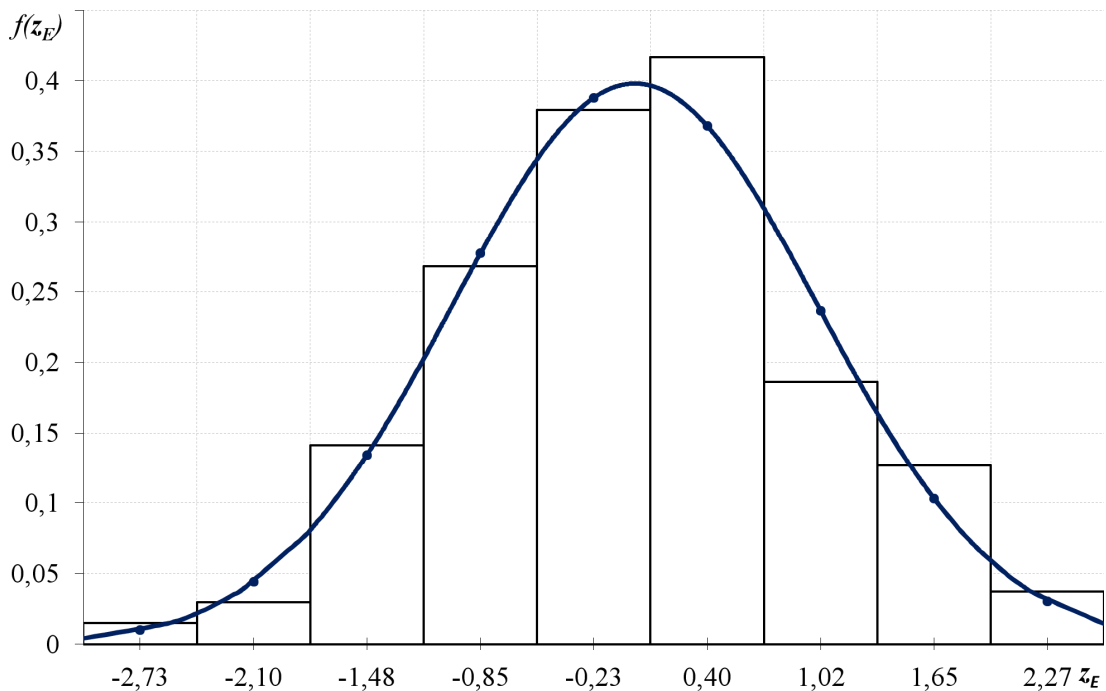


Рис. 3.32. Розподіл нормалізованих за перетворенням Джонсона значень трудомісткості проектів з розробки програмного забезпечення для платформи mainframe:

- – емпіричний розподіл нормалізованої величини;
- – нормальний розподіл

Результати нормалізації величини E за перетвореннями (3.1), (3.2), (2.17) та оцінки ймовірнісних характеристик величини z_E наведені в таблиці 3.6.

Таблиця 3.6 – Оцінки ймовірнісних характеристик випадкової величини z_E

Нормалізуюче перетворення	\bar{z}_E	S_z	\hat{A}_z	$\hat{\varepsilon}_z$	χ^2	δ
Десятковий логарифм (3.1)	3,34	0,63	0,301	2,68	13,34	0,196
Перетворення Бокса-Кокса (3.2)	5,70	0,77	0,030	2,58	5,81	0,178
Перетворення Джонсона S_B (2.17)	0,00	1,00	0,000	3,03	4,34	0,001

З таблиці 3.6 видно, що перетворення (3.2) та (2.17) дозволяють отримати випадкову величину z_E з нормальним законом розподілу: значення критерію Пірсона χ^2 при використанні вказаних перетворень було менше критичного значення $\chi_{кр}^2 = 12,59$ ($\nu = 9 - 2 - 1 = 6$, $\alpha = 0,05$). Для перетворення (3.1) значення критерію Пірсона χ^2 є більшим, ніж критичне значення $\chi_{кр}^2$, тому це перетворення неможна використовувати для нормалізації випадкової величини E .

Щоб порівняти результати нормалізації і вибрати нормалізуюче перетворення для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи mainframe використовуємо розраховану за (3.4) похибку δ . Як видно з таблиці 3.6 серед перетворень (3.2) та (2.17) найменше значення похибки δ було отримане при нормалізації випадкової величини E , з використанням перетворення (2.17). Отже, для нормалізації емпіричних даних трудомісткості проектів з розробки програмного забезпечення для платформи mainframe краще використовувати нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17).

3.5 Побудова нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення

Як було показано в попередніх підрозділах, розподіл емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення відрізняється від нормального закону розподілу. Тому для побудови нелінійних регресійних моделей буде використано нормалізовані значення тривалості та трудомісткості проектів з розробки програмного забезпечення. Для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17). Як було показано в попередніх підрозділах, серед нормалізуючих перетворень (3.1), (3.2), (2.15) саме перетворення Джонсона (2.15) сім'ї S_B (2.17) дозволяє краще нормалізувати вказані емпіричні дані.

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення (в місяцях), а випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення (в людино-годинах).

З випадкових величин D та E , використовуючи нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) зі знайденими в попередніх підрозділах параметрами $\{\gamma_D, \eta_D, \phi_D, \lambda_D\}$ та $\{\gamma_E, \eta_E, \phi_E, \lambda_E\}$, буде отримано випадкові величини z_D та z_E з нормальним законом розподілу. Так як розподіл величин z_D та z_E є нормальним, то це дозволяє побудувати рівняння лінійної регресії для нормалізованих значень:

$$\hat{z}_D(z_E) = b_0 + b_1 z_E, \quad (3.5)$$

де b_0 і b_1 - коефіцієнти рівняння лінійної регресії.

Значення коефіцієнтів b_0 і b_1 рівняння лінійної регресії (3.5) знаходяться за методом найменших квадратів [87]:

$$b_1 = \frac{\overline{z_E z_D} - \overline{z_E} \cdot \overline{z_D}}{\overline{z_E^2} - \overline{z_E}^2}; \quad b_0 = \overline{z_D} - b_1 \cdot \overline{z_E}, \quad (3.6)$$

$$\text{де } \overline{z_D} = \frac{1}{n} \sum_{i=1}^n z_{Di}; \quad \overline{z_E} = \frac{1}{n} \sum_{i=1}^n z_{Ei}; \quad \overline{z_E z_D} = \frac{1}{n} \sum_{i=1}^n z_{Ei} z_{Di}; \quad \overline{z_E^2} = \frac{1}{n} \sum_{i=1}^n z_{Ei}^2.$$

Довірчий інтервал лінійної регресії (3.5) задається як [87]

$$[\overline{z_D}(z_E)] = \hat{z}_D(z_E) \pm t_{\alpha/2, n-2} \cdot \sqrt{s_{z_D}^2 \sqrt{\frac{1}{n} + \frac{(z_E - \overline{z_E})^2}{S_{z_E}}}}, \quad (3.7)$$

де n - це кількість емпіричних значень;

$$s_{z_D}^2 = \frac{1}{n-2} \sum_{i=1}^n (z_{Di} - \hat{z}_D(z_{Ei}))^2; \quad S_{z_E} = \sum_{i=1}^n (z_{Ei} - \overline{z_E})^2;$$

$t_{\alpha/2, n-2}$ - квантіль t -розподілу Стьюдента, визначається за таблицею верхніх 100 α % точок t -розподілу Стьюдента за рівнем значимості $\alpha/2$ та кількістю ступенів вільності $n-2$.

Інтервал прогнозування лінійної регресії (3.5) задається як [87]

$$[z_D(z_E)] = \hat{z}_D(z_E) \pm t_{\alpha/2, n-2} \cdot \sqrt{s_{z_D}^2 \sqrt{1 + \frac{1}{n} + \frac{(z_E - \overline{z_E})^2}{S_{z_E}}}}, \quad (3.8)$$

де n - це кількість емпіричних значень;

$$s_{z_D}^2 = \frac{1}{n-2} \sum_{i=1}^n (z_{Di} - \hat{z}_D(z_{Ei}))^2; \quad S_{z_E} = \sum_{i=1}^n (z_{Ei} - \overline{z_E})^2;$$

$t_{\alpha/2, n-2}$ - квантіль t -розподілу Стьюдента, визначається за таблицею верхніх 100 α % точок t -розподілу Стьюдента за рівнем значимості $\alpha/2$ та кількістю ступенів вільності $n-2$.

Враховуючи (3.5) і те, що для нормалізації емпіричних даних тривалості проектів з розробки програмного забезпечення та емпіричних даних трудомісткості проектів з розробки програмного забезпечення використовувалося перетворення Джонсона (2.15) сім'ї S_B (2.17) (тобто $z_D = \gamma_D + \eta_D \ln\left(\frac{D - \varphi_D}{\lambda_D + \varphi_D - D}\right)$; $z_E = \gamma_E + \eta_E \ln\left(\frac{E - \varphi_E}{\lambda_E + \varphi_E - E}\right)$), то рівняння нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів може бути представлена як

$$\hat{D}(E) = \varphi_D + \frac{p\lambda_D}{p+1}, \quad (3.9)$$

$$\text{де } p = \exp\left\{\frac{\hat{z}_D(z_E) - \gamma_D}{\eta_D}\right\}; \quad z_D(z_E) = b_0 + b_1 z_E; \quad z_E = \gamma_E + \eta_E \ln\left(\frac{E - \varphi_E}{\lambda_E + \varphi_E - E}\right);$$

b_0 і b_1 - коефіцієнти рівняння лінійної регресії (3.5) для нормалізованих значень за перетворенням Джонсона (2.15) сім'ї S_B (2.17).

Враховуючи (3.9), нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів тривалості задається як

$$D(E) = \hat{D}(E) + \varepsilon, \quad (3.10)$$

де ε – це випадкова похибка, яка має розподіл Джонсона сім'ї S_B .

Враховуючи (3.7), довірчий інтервал $[\bar{D}(E)]$ нелінійної регресії (3.9) задається як

$$[\bar{D}(E)] = \varphi_D + \frac{[\bar{q}]\lambda_D}{[\bar{q}] + 1}, \quad (3.11)$$

$$\text{де } [\bar{q}] = \exp\left\{\frac{[\bar{z}_D(z_E)] - \gamma_D}{\eta_D}\right\}, \quad [\bar{z}_D(z_E)] \text{ задається як (3.7).}$$

Враховуючи (3.8), інтервал прогнозування $[D(E)]$ нелінійної регресії (3.9) задається як

$$[D(E)] = \varphi_D + \frac{[q]\lambda_D}{[q]+1}, \quad (3.12)$$

де $[q] = \exp\left\{\frac{[z_D(z_E)] - \gamma_D}{\eta_D}\right\}$, $[z_D(z_E)]$ задається як (3.8).

Якщо виконувати нормалізацію за перетворенням (3.1) (тобто $z_D = \log_{10} D$; $z_E = \log_{10} E$, як це було зроблено при побудові моделей СОСОМО та ISBSG), то, враховуючи (3.5), нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів може бути представлена як

$$\hat{D}(E) = 10^{b_0} E^{b_1}, \quad (3.13)$$

де b_0 і b_1 - коефіцієнти рівняння лінійної регресії (3.5) для нормалізованих значень за перетворенням у вигляді десяткового логарифму (3.1).

Для того щоб порівнювати рівняння нелінійних регресій можна використовувати такий критерій як сума квадратів відхилень між тривалістю, прогнозованою за рівнянням регресії та емпіричними даними:

$$SSE = \sum_{i=1}^n (D_i - \hat{D}(E_i))^2, \quad (3.14)$$

де n - це кількість емпіричних значень;

D_i - фактичне значення, а $\hat{D}(E_i)$ - прогнозоване випадкової величини D .

Кращим є рівняння регресії, для якого значення SSE є меншим.

Також для порівняння рівнянь нелінійних регресій можна використовувати значення коефіцієнта детермінації R^2 :

$$R^2 = 1 - \frac{\sum_{i=1}^n (D_i - \hat{D}(E_i))^2}{\sum_{i=1}^n (D_i - \bar{D})^2}, \quad (3.15)$$

де n - це кількість емпіричних значень;

D_i - фактичне значення, а $\hat{D}(E_i)$ - прогнозоване випадкової величини D ;

\bar{D} - середнє значення випадкової величини D , $\bar{D} = \frac{1}{n} \sum_{i=1}^n D_i$.

Кращим є рівняння регресії, для якого значення R^2 є більшим.

Ще одним з критеріїв, який може бути використаний для порівняння рівнянь регресій є відносна кількість значень, які були спрогнозовані з певною точністю l [86]:

$$PRED(l) = \frac{100}{n} \sum_{i=1}^n \begin{cases} 1, & \text{якщо } MRE_i \leq l \\ 0, & \text{якщо } MRE_i > l \end{cases}, \quad (3.16)$$

де n - це кількість емпіричних даних; $MRE_i = \left| \frac{D_i - \hat{D}(E_i)}{D_i} \right|$;

D_i - фактичне значення, а $\hat{D}(E_i)$ - прогнозоване випадкової величини D .

Кращим є рівняння регресії, для якого значення $PRED(l)$ є більшим. При порівнянні звичайно використовують $PRED(0,25)$.

Відносна похибка за нелінійною регресійною моделлю задається як

$$\delta = \frac{1}{2} \left| \frac{\bar{D}(E_i)_{\max} - \bar{D}(E_i)_{\min}}{\hat{D}(E_i)} \right| \cdot 100\%,$$

де $\bar{D}(E_i)_{\max}$ - верхнє значення довірчого інтервалу (3.11) нелінійної регресії;

$\bar{D}(E_i)_{\min}$ - нижнє значення довірчого інтервалу (3.11) нелінійної регресії;

$\hat{D}(E_i)$ - прогнозоване за рівнянням регресії значення випадкової величини D .

3.5.1 Побудова нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для всіх платформ

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення, а випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформ PC, mid-range та mainframe з бази даних ISBSG [10, 11] (348 проектів).

Розподіл випадкових величин D і E відрізняється від нормального закону розподілу. Для нормалізації випадкової величини D було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_D = 15,905$, $\eta_D = 1,237$, $\phi_D = -0,106$ і $\lambda_D = 3352506$. Для нормалізації випадкової величини E було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_E = 3,80913$, $\eta_E = 0,65406$, $\phi_E = 82,20693$ і $\lambda_E = 667093$.

Після нормалізації, за нормалізованими випадковими величинами z_D та z_E було побудовано рівняння лінійної регресії (3.5). При її побудові 8 значень емпіричних даних було вилучено через високі значення зовнішніх студентизованих залишків та плеча. Отримані за (3.6) значення коефіцієнтів регресійної моделі були такими: $b_0 = -0,018$ і $b_1 = 0,553$. Значення t -розподілу Стьюдента, що використовувалось при побудові рівнянь довірчих інтервалів (3.7) та (3.8) було $t_{(0,05/2),(340-2)} = 1,968$. Рівняння лінійної регресії (3.5) з отриманими значеннями коефіцієнтів та 95% інтервал прогнозування лінійної регресії показані на рис. 3.33.

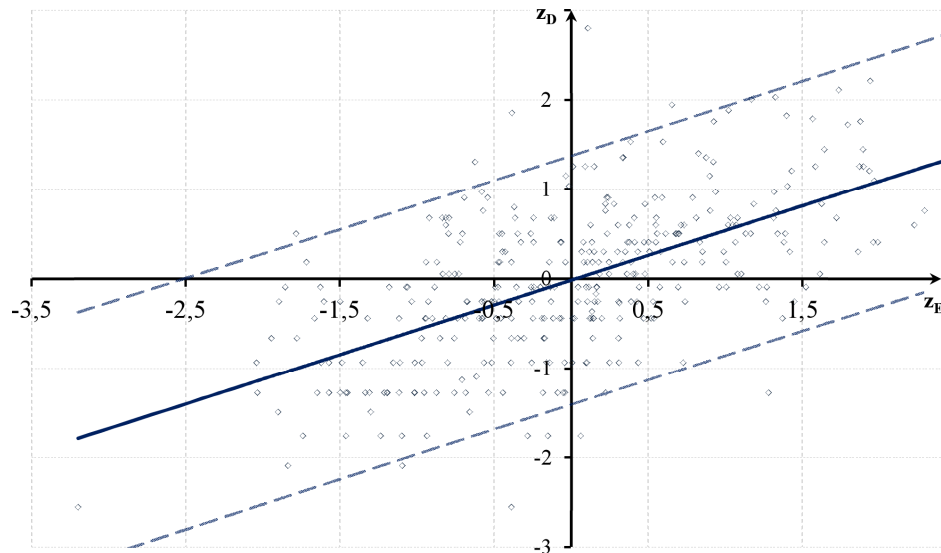


Рис. 3.33. Рівняння лінійної регресії та 95% інтервал прогнозування лінійної регресії для нормалізованих значень тривалості проектів з розробки програмного забезпечення в залежності від нормалізованих значень трудомісткості цих проектів для всіх платформ

Далі, за (3.9), використовуючи наведені вище значення параметрів перетворення Джонсона та отримані коефіцієнти рівняння лінійної регресії, отримуємо нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для всіх платформ. За (3.11), використовуючи наведене вище значення t -розподілу Стюдента, отримуємо 95% довірчий інтервал нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для всіх платформ. Аналогічно, за (3.12), отримуємо 95% інтервал прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для всіх платформ.

Також, для порівняння з нелінійною регресійною моделлю на основі перетворення Джонсона (3.9), було побудовано нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення у вигляді десяткового логарифму (3.13). Значення коефіцієнтів цієї регресійної моделі були такими: $b_0 = -0,145$ і $b_1 = 0,322$.

Отримані нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів та інтервал прогнозування нелінійної регресії на основі перетворення Джонсона показані на рис. 3.34. З цього рисунка видно, що більшість емпіричних даних попадають в отриманий довірчий інтервал.

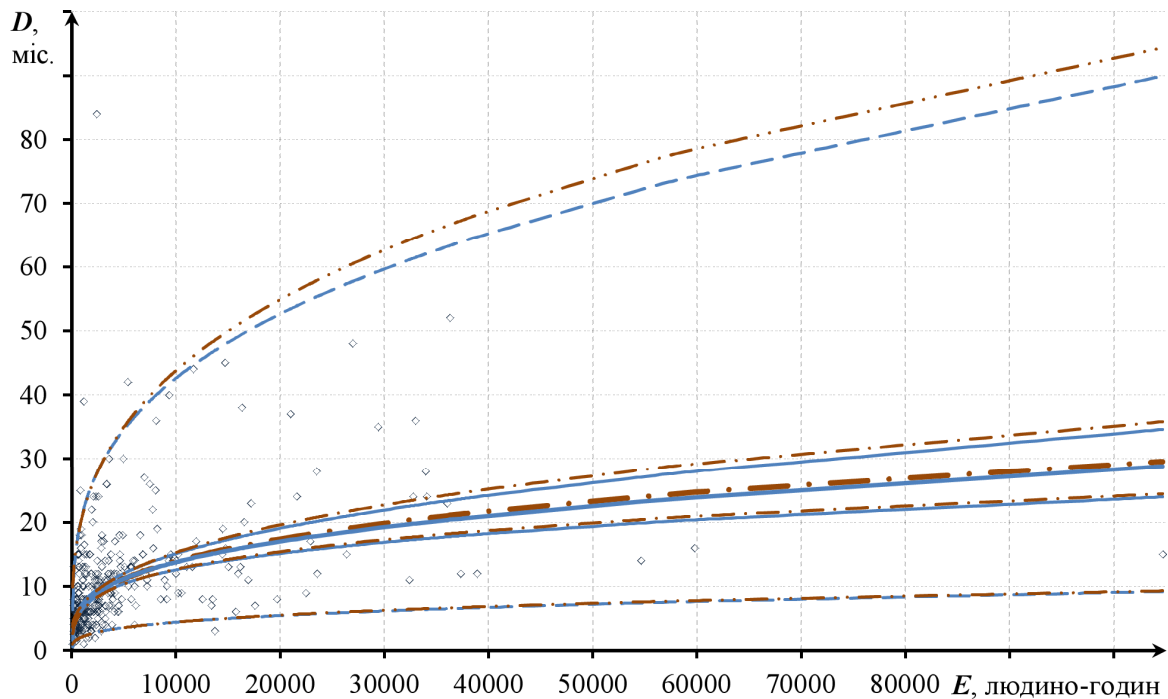


Рис. 3.34. Нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для всіх платформ:

- ◇ - емпіричні дані (E – трудомісткість проекту; D – тривалість робіт);
- (світло-синя) - нелінійна регресія на основі перетворення Джонсона;
- (темно-синя) - 95% довірчий інтервал нелінійної регресії на основі перетворення Джонсона;
- (темно-синя з крапками) - 95% інтервал прогнозування нелінійної регресії на основі перетворення Джонсона;
- (коричнева з крапками) - нелінійна регресія на основі десяткового логарифму;
- (коричнева з крапками та крапками) - 95% довірчий інтервал нелінійної регресії на основі перетворення у вигляді десяткового логарифму;
- (коричнева з крапками та крапками та крапками) - 95% інтервал прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму

Характеристики отриманих рівнянь нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для всіх платформ були наступні: сума квадратів відхилень між тривалістю, прогнозованою за моделлю та емпіричними даними (3.14) була $SSE = 22744$ для рівняння (3.9) і $SSE = 22766$ для рівняння (3.13); коефіцієнт детермінації (3.15) був $R^2 = 0,2033$ для рівняння (3.9) і $R^2 = 0,2025$ для рівняння (3.13); відносна кількість значень, які були спрогнозовані з 25% точністю (3.16) була $PRED(0,25) = 35,88\%$ для рівняння (3.9) і $PRED(0,25) = 35,29\%$ для рівняння (3.13). Відносна похибка для нелінійної регресійної моделі (3.10) не перевищує 15%.

Ширина 95% довірчого інтервалу нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона є на 5% меншою, ніж ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Ширина 95% інтервалу прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона є на 6% меншою, ніж ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму.

Нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона (3.9) має кращі характеристики ніж аналогічна модель на основі перетворення у вигляді десяткового логарифму (3.13), а довірчий інтервал нелінійної регресії і інтервал прогнозування нелінійної регресії на основі перетворення Джонсона є вужчим, ніж відповідні довірчі інтервали нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Тому, використання моделі (3.9) з отриманими вище параметрами дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення.

3.5.2 Побудова нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи РС

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення, а випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформи РС з бази даних ISBSG [10, 11] (52 проекти).

Розподіл випадкових величин D і E відрізняється від нормального закону розподілу. Для нормалізації випадкової величини D було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_D = 2,38983$, $\eta_D = 1,30278$, $\phi_D = 0,43283$ і $\lambda_D = 54,7471$. Для нормалізації випадкової величини E було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_E = 1,61212$, $\eta_E = 0,53883$, $\phi_E = 158,85317$ і $\lambda_E = 18248,7$.

Після нормалізації, за нормалізованими випадковими величинами z_D та z_E було побудовано рівняння лінійної регресії (3.5). При її побудові 3 емпіричні значення було вилучено через високі значення зовнішніх студентизованих залишків та плеча. Отримані за (3.6) значення коефіцієнтів регресійної моделі були такими: $b_0 = -0,051$ і $b_1 = 0,492$. Значення t -розподілу Стьюдента, що використовувалось при побудові рівнянь довірчих інтервалів (3.7) та (3.8) було $t_{(0,05/2),(49-2)} = 2,02$. Рівняння лінійної регресії (3.5) з отриманими значеннями коефіцієнтів та 95% довірчий інтервал для значень цієї моделі показані на рис. 3.35.

Далі, за (3.9), використовуючи наведені вище значення параметрів перетворення Джонсона та отримані коефіцієнти рівняння лінійної регресії, отримуємо нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи РС. За (3.11), використовуючи наведене вище значення t -розподілу Стьюдента, отримуємо 95% довірчий інтервал нелінійної регресії тривалості

проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи РС. Аналогічно, за (3.12), отримуємо 95% інтервал прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи РС.

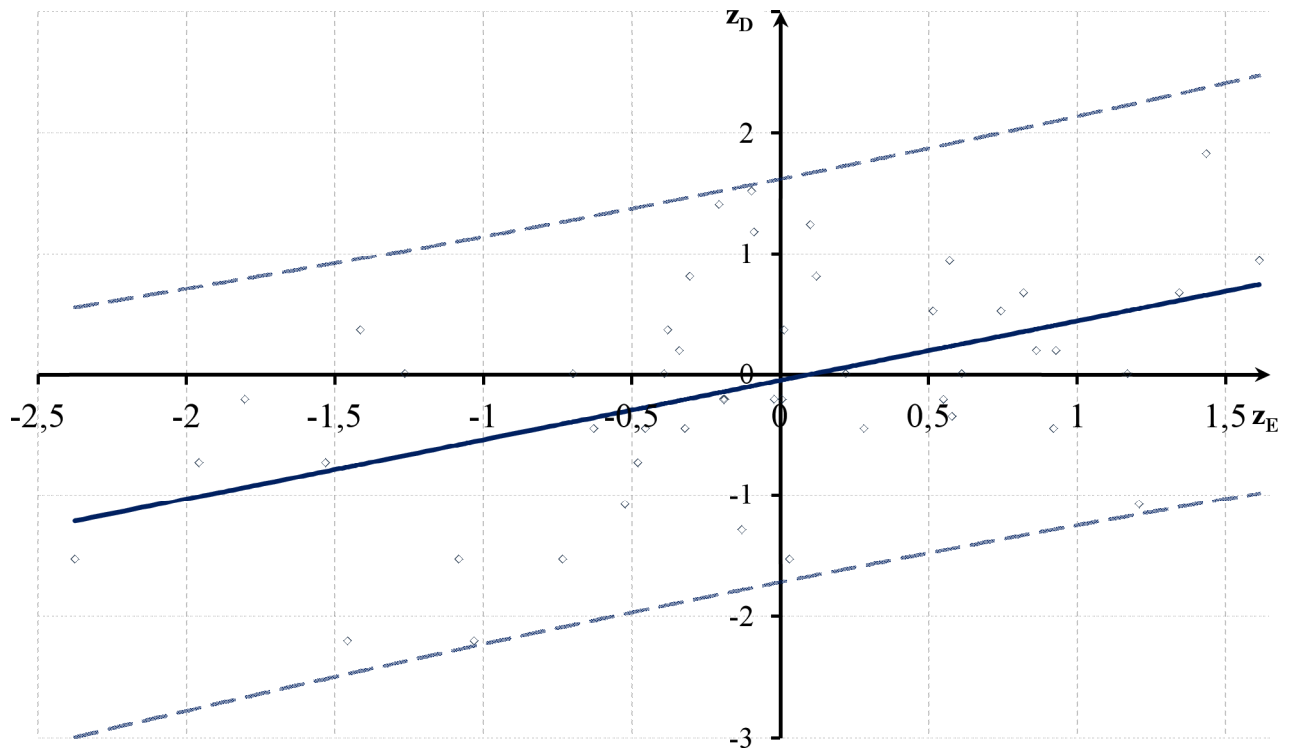


Рис. 3.35. Рівняння лінійної регресії та 95% інтервал прогнозування лінійної регресії для нормалізованих значень тривалості проектів з розробки програмного забезпечення в залежності від нормалізованих значень трудомісткості цих проектів для платформи РС

Також, для порівняння з нелінійною регресійною моделлю на основі перетворення Джонсона (3.9), було побудовано нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення у вигляді десяткового логарифму (3.13). Значення коефіцієнтів цієї регресійної моделі були такими: $b_0 = 0,14$ і $b_1 = 0,242$

Отримані нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів та інтервал прогнозування нелінійної регресії на основі перетворення Джонсона

показані на рис. 3.36. З цього рисунка видно, що більшість емпіричних даних попадають в отриманий довірчий інтервал.

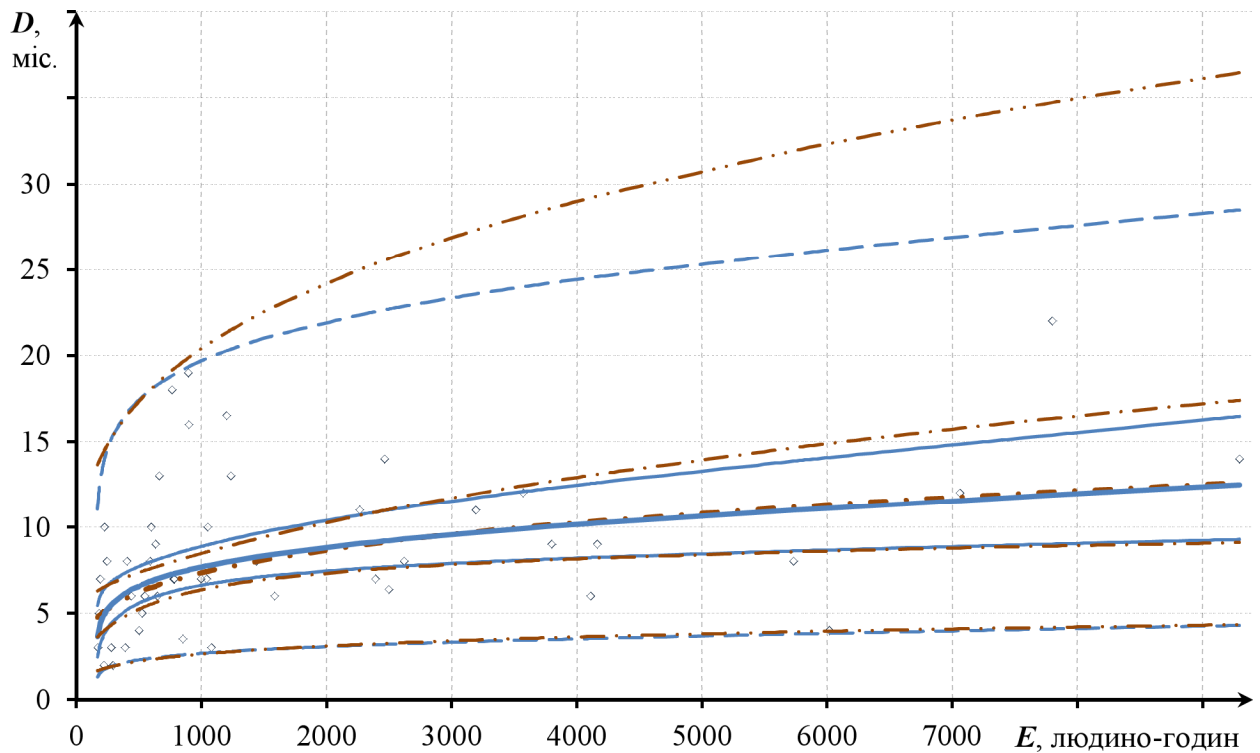


Рис. 3.36. Нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи РС:

- ◇ - емпіричні дані (E – трудомісткість проекту; D – тривалість робіт);
- (світло-синя) - нелінійна регресія на основі перетворення Джонсона;
- (темно-синя) - 95% довірчий інтервал нелінійної регресії на основі перетворення Джонсона;
- (темно-синя, пунктирна) - 95% інтервал прогнозування нелінійної регресії на основі перетворення Джонсона;
- (коричнева, пунктирна) - нелінійна регресія на основі десятичного логарифму
- (коричнева, пунктирна) - 95% довірчий інтервал нелінійної регресії на основі перетворення у вигляді десятичного логарифму;
- (коричнева, пунктирна) - 95% інтервал прогнозування нелінійної регресії на основі перетворення у вигляді десятичного логарифму

Характеристики отриманих рівнянь нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості

цих проектів для платформи РС були наступні: сума квадратів відхилень між тривалістю, прогнозованою за моделлю та емпіричними даними (3.14) була $SSE = 844$ для рівняння (3.9) і $SSE = 873$ для рівняння (3.13); коефіцієнт детермінації (3.15) був $R^2 = 0,16$ для рівняння (3.9) і $R^2 = 0,13$ для рівняння (3.13); відносна кількість значень, які були спрогнозовані з 25% точністю (3.16) була $PRED(0,25) = 47\%$ для рівняння (3.9) і $PRED(0,25) = 39\%$ для рівняння (3.13). Відносна похибка для нелінійної регресійної моделі (3.10) не перевищує 15%.

Ширина 95% довірчого інтервалу нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона для платформи РС є на 13% меншою, ніж ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Ширина 95% інтервалу прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона для платформи РС є на 25% меншою, ніж ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму.

Нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона (3.9) має кращі характеристики ніж аналогічна модель на основі перетворення у вигляді десяткового логарифму (3.13), а довірчий інтервал нелінійної регресії і інтервал прогнозування нелінійної регресії на основі перетворення Джонсона є вужчим, ніж відповідні довірчі інтервали нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Тому, використання моделі (3.9) з отриманими вище параметрами дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення для платформи РС.

3.5.3 Побудова нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mid-range

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення, а випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформи mid-range з бази даних ISBSG [10, 11] (81 проект).

Розподіл випадкових величин D і E відрізняється від нормального закону розподілу. Для нормалізації випадкової величини D було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_D = 5,682$, $\eta_D = 1,13146$, $\phi_D = 1,04977$ і $\lambda_D = 1166,2$. Для нормалізації випадкової величини E було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_E = 4,0686$, $\eta_E = 0,77997$, $\phi_E = 82,51282$ і $\lambda_E = 565242$.

Після нормалізації, за нормалізованими випадковими величинами z_D та z_E було побудовано рівняння лінійної регресії (3.5). При її побудові 6 значень емпіричних даних було вилучено через високі значення зовнішніх студентизованих залишків та плеча. Отримані за (3.6) значення коефіцієнтів регресійної моделі були такими: $b_0 = -0,093$ і $b_1 = 0,588$. Значення t -розподілу Стьюдента, що використовувалось при побудові рівнянь довірчих інтервалів (3.7) та (3.8) було $t_{(0,05/2),(76-2)} = 2$. Рівняння лінійної регресії (3.5) з отриманими значеннями коефіцієнтів та 95% інтервал прогнозування регресії показані на рис. 3.37.

Далі, за (3.9), використовуючи наведені вище значення параметрів перетворення Джонсона та отримані коефіцієнти рівняння лінійної регресії, отримуємо нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mid-range. За (3.11), використовуючи наведене вище значення t -розподілу Стьюдента, отримуємо 95% довірчий інтервал нелінійної регресії

тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mid-range.

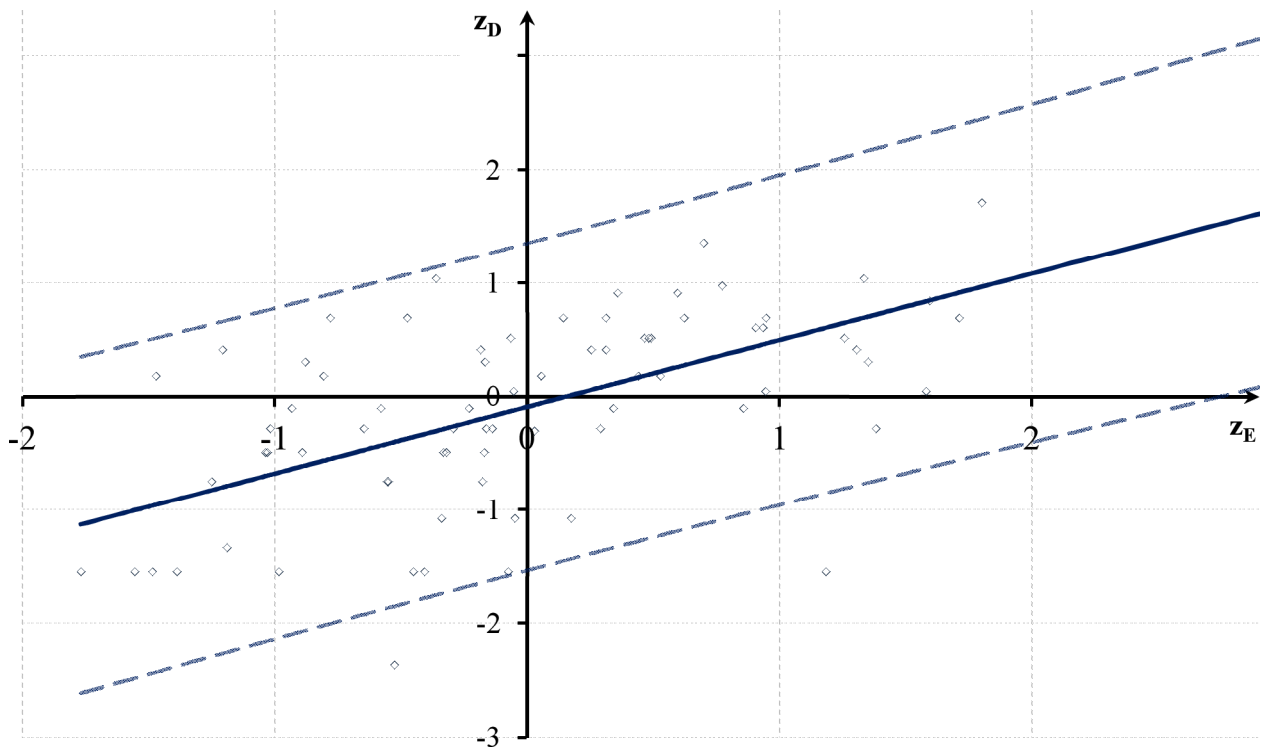


Рис. 3.37. Рівняння лінійної регресії та 95% інтервал прогнозування лінійної регресії для нормалізованих значень тривалості проектів з розробки програмного забезпечення в залежності від нормалізованих значень трудомісткості цих проектів для платформи mid-range

Також, для порівняння з нелінійною регресійною моделлю на основі перетворення Джонсона (3.9), було побудовано нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення у вигляді десяткового логарифму (3.13). Значення коефіцієнтів цієї регресійної моделі були такими: $b_0 = -0,344$ і $b_1 = 0,362$.

Отримані нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів та інтервал прогнозування нелінійної регресії на основі перетворення Джонсона показані на рис. 3.38. З цього рисунка видно, що більшість емпіричних даних попадають в отриманий довірчий інтервал.

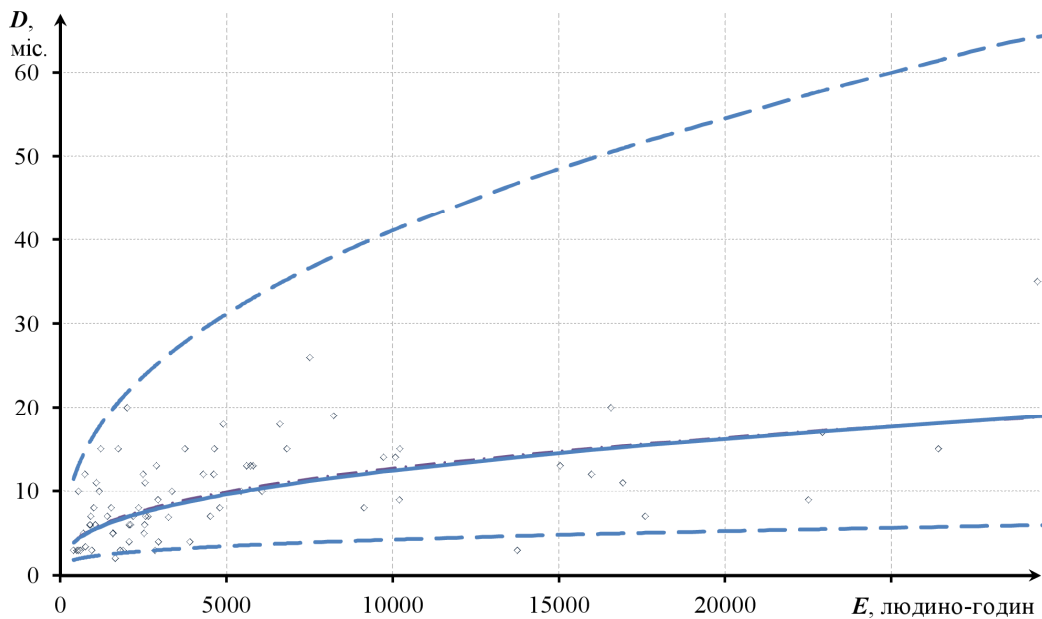


Рис. 3.38. Нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mid-range:

- ◇ - емпіричні дані (E – трудомісткість проекту; D – тривалість робіт);
- - регресійна модель на основі перетворення Джонсона;
- - - 95% інтервал прогнозування нелінійної регресії на основі пер. Джонсона;
- · - - нелінійної регресійна модель на основі перетворення у вигляді десятичного логарифму

Характеристики отриманих рівнянь нелінійної регресії тривалості в залежності від трудомісткості для платформи mid-range були наступні: сума квадратів відхилень між тривалістю, прогнозованою за моделлю та емпіричними даними (3.14) була $SSE = 3905$ для рівняння (3.9) і $SSE = 4138$ для рівняння (3.13); коефіцієнт детермінації (3.15) був $R^2 = 0,453$ для рівняння (3.9) і $R^2 = 0,420$ для рівняння (3.13); відносна кількість значень, які були спрогнозовані з 25% точністю (3.16) була $PRED(0,25) = 42\%$ для рівняння (3.9) і $PRED(0,25) = 41\%$ для рівняння (3.13). Відносна похибка для нелінійної регресійної моделі (3.10) не перевищує 15%.

Нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі

перетворення Джонсона (3.9) має кращі характеристики ніж аналогічна модель на основі перетворення у вигляді десяткового логарифму (3.13). Тому, використання моделі (3.9) з отриманими вище параметрами дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення для платформи mid-range.

3.5.4 Побудова нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mainframe

Нехай випадкова величина D – це емпіричні значення тривалості проектів з розробки програмного забезпечення, а випадкова величина E – це емпіричні значення трудомісткості проектів з розробки програмного забезпечення для платформи mainframe з бази даних ISBSG [10, 11] (215 проектів).

Розподіл випадкових величин D і E відрізняється від нормального закону розподілу. Для нормалізації випадкової величини D було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_D = 9,22106$, $\eta_D = 1,30531$, $\phi_D = -0,25458$ і $\lambda_D = 10513,2$. Для нормалізації випадкової величини E було вибрано нормалізуюче перетворення Джонсона (2.15) сім'ї S_B (2.17) з такими параметрами: $\gamma_E = 3,73121$, $\eta_E = 0,62765$, $\phi_E = 81,46192$ і $\lambda_E = 753187$.

Після нормалізації, за нормалізованими випадковими величинами z_D та z_E було побудовано рівняння лінійної регресії (3.5). При її побудові з емпіричні значення було вилучено через високі значення зовнішніх студентизованих залишків та плеча. Отримані за (3.6) значення коефіцієнтів регресійної моделі були такими: $b_0 = -0,045$ і $b_1 = 0,691$. Значення t -розподілу Стьюдента, що використовувалось при побудові довірчого інтервалу (3.7) було $t_{(0,05/2),(205-2)} = 1,97$. Рівняння лінійної регресії (3.5) з отриманими значеннями коефіцієнтів та 95% інтервал прогнозування регресії показані на рис. 3.39.

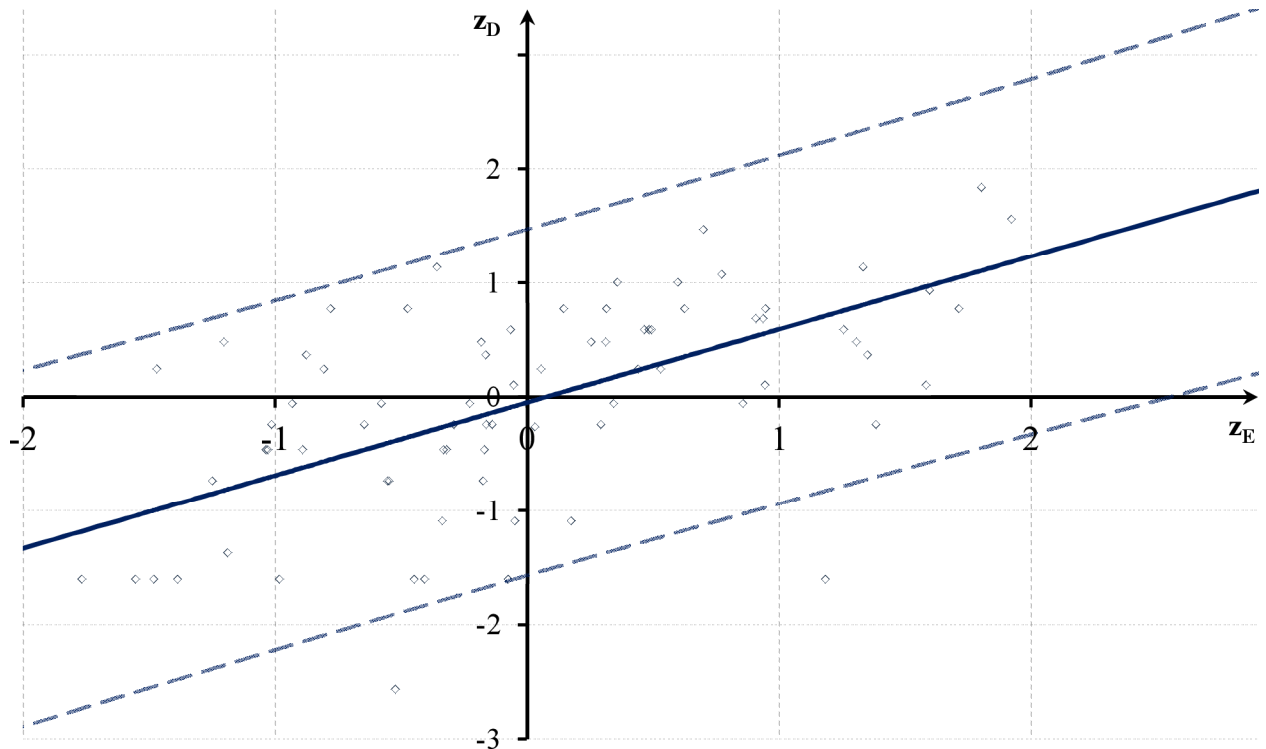


Рис. 3.39. Рівняння лінійної регресії та 95% інтервал прогнозування лінійної регресії для нормалізованих значень тривалості проектів з розробки програмного забезпечення в залежності від нормалізованих значень трудомісткості цих проектів для платформи mainframe

Далі, за (3.9), використовуючи наведені вище значення параметрів перетворення Джонсона та отримані коефіцієнти рівняння лінійної регресії, отримуємо нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mainframe. За (3.11), використовуючи наведене вище значення t -розподілу Стюдента, отримуємо 95% довірчий інтервал нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mainframe. Аналогічно, за (3.12), отримуємо 95% інтервал прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mainframe.

Також, для порівняння з нелінійною регресійною моделлю на основі перетворення Джонсона (3.9), було побудовано нелінійну регресійну модель тривалості проектів з розробки програмного забезпечення в залежності від

трудомісткості цих проектів на основі перетворення у вигляді десяткового логарифму (3.13). Значення коефіцієнтів цієї регресійної моделі були такими: $b_0 = -0,382$ і $b_1 = 0,373$.

Отримані нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів та інтервал прогнозування нелінійної регресії на основі перетворення Джонсона показані на рис. 3.40. З цього рисунка видно, що більшість емпіричних даних попадають в отриманий довірчий інтервал.

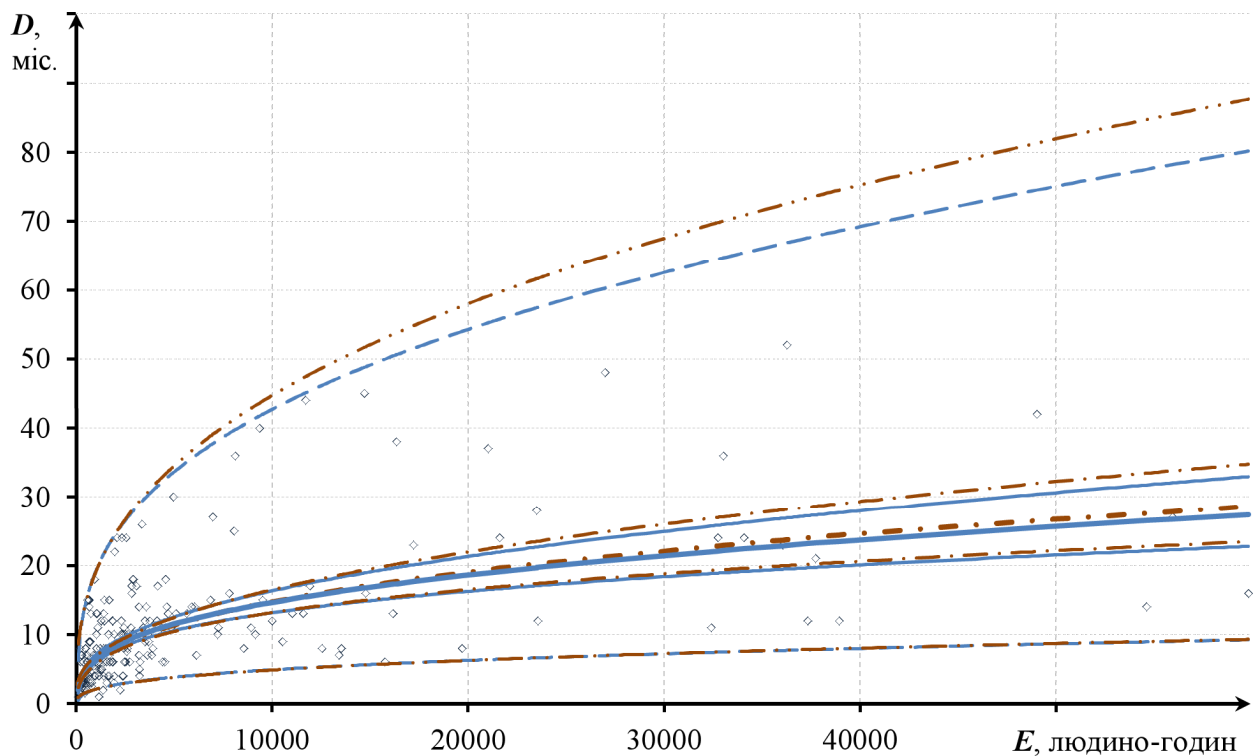


Рис. 3.40. Нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mainframe:

- ◇ - емпіричні дані (E – трудомісткість проекту; D – тривалість робіт)
- - нелінійна регресія на основі перетворення Джонсона;
- - 95% довірчий інтервал нелінійної регресії на основі пер. Джонсона;
- - 95% інтервал прогнозування нелінійної регресії на основі пер. Джонсона;
- · — - нелінійна регресія на основі десяткового логарифму;
- · — - 95% довірчий інтервал нелінійної регресії на основі дес. логарифму;
- · — - 95% інтервал прогнозування нелінійної регресії на основі дес. логарифму

Характеристики отриманих рівнянь нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформи mainframe були наступні: сума квадратів відхилень між тривалістю, прогнозованою за моделлю та емпіричними даними (3.14) була $SSE = 11046$ для рівняння (3.9) і $SSE = 11059$ для рівняння (3.13); коефіцієнт детермінації (3.15) був $R^2 = 0,366$ для рівняння (3.9) і $R^2 = 0,365$ для рівняння (3.13); відносна кількість значень, які були зпрогнозовані з 25% точністю (3.16) була $PRED(0,25) = 37\%$ для рівняння (3.9) і $PRED(0,25) = 36\%$ для рівняння (3.13). Відносна похибка для нелінійної регресійної моделі (3.10) не перевищує 15%.

Ширина 95% довірчого інтервалу нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона для платформи mainframe є на 10% меншою, ніж ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Ширина 95% інтервалу прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона для платформи mainframe є на 10% меншою, ніж ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму.

Нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів на основі перетворення Джонсона (3.9) має кращі характеристики ніж аналогічна модель на основі перетворення у вигляді десяткового логарифму (3.13), а довірчий інтервал нелінійної регресії і інтервал прогнозування нелінійної регресії на основі перетворення Джонсона є вужчим, ніж відповідні довірчі інтервали нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Тому, використання моделі (3.9) з отриманими вище параметрами дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення для платформи mainframe.

Порівняння рівнянь нелінійної регресії, довірчих інтервалів та інтервалів прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів наведено в таблиці 3.7. З даної таблиці видно, що за критеріями R^2 , SSE та $PRED(25\%)$ побудовані рівняння нелінійної регресії на основі перетворення Джонсона є кращими, ніж відповідні рівняння на основі перетворення у вигляді десяткового логарифму. В залежності від платформи ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення Джонсона є на 10-13% меншою, ніж ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення у вигляді десяткового логарифму. Також, в залежності від платформи ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення Джонсона є на 10-25% меншою, ніж ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму.

Таблиця 3.7 – Порівняння рівнянь нелінійної регресії, довірчих інтервалів та інтервалів прогнозування нелінійної регресії

Платформа ПЗ	Нелінійна регресія на основі перетворення Джонсона сім'ї S_B (нова)			Нелінійна регресія на основі десяткового логарифму (існуюча)			Зменш. інтервалів в порівнянні з інтервалами регресії на основі десяткового логарифму	
	SSE	R^2	$PRED_{25\%}$	SSE	R^2	$PRED_{25\%}$	$[\bar{D}(E)]$ (3.11)	$[D(E)]$ (3.12)
PC	844	0,16	47%	873	0,13	39%	13%	25%
mid-range	3905	0,453	42%	4138	0,420	41%	10%	10%
mainframe	11046	0,366	37%	11059	0,365	36%	10%	10%

3.6 Висновки до розділу 3

В даному розділі було побудовано нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range та mainframe на основі перетворення Джонсона.

1. Проведено аналіз емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення, показано що закон розподілів цих емпіричних даних відрізняється від нормального закону розподілу.
2. Зроблено порівняння перетворення у вигляді десяткового логарифму, перетворення Бокса-Кокса та перетворення Джонсона для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення. Вибрано нормалізуюче перетворення Джонсона сім'ї S_B для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення платформ PC, mid-range та mainframe, тому що це перетворення дозволяє краще виконувати нормалізацію емпіричних даних.
3. Побудовано нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range та mainframe *за рахунок* застосування нормалізуючого перетворення Джонсона сім'ї S_B , *що дозволяє* підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. За критеріями R^2 , SSE та $PRED(25\%)$ побудовані рівняння нелінійної регресії на основі перетворення Джонсона є кращими, ніж відповідні рівняння на основі перетворення у вигляді десяткового логарифму. Відносна похибка для побудованих нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів не перевищує 15%.

4. Побудовано рівняння нижньої та верхньої границь довірчих інтервалів нелінійних регресій тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range та mainframe за рахунок застосування нормалізуючого перетворення Джонсона сім'ї S_B , що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. В залежності від платформи ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення Джонсона є на 10-13% меншою, ніж ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення у вигляді десяткового логарифму.
5. Побудовано рівняння нижньої та верхньої границь інтервалів прогнозування нелінійних регресій тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для платформ PC, mid-range та mainframe за рахунок застосування нормалізуючого перетворення Джонсона сім'ї S_B , що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. В залежності від платформи ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення Джонсона є на 10-25% меншою, ніж ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму.

Результати даного розділу були опубліковані в [53-56, 80- 84].

РОЗДІЛ 4

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕРОБКИ ІНФОРМАЦІЇ ДЛЯ ОЦІНЮВАННЯ ТРИВАЛОСТІ ПРОЕКТІВ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

На даний час при розробці програмного забезпечення використовуються різні складні інформаційні технології переробки інформації для оцінювання тривалості програмних проектів (такі як Comparative Estimating Tool [88] від International Software Benchmarking Standards Group (ISBSG) чи Microsoft Project [89]). Незважаючи на використання вказаних інформаційних технологій не перестає існувати проблема низької достовірності оцінювання тривалості проектів з розробки програмного забезпечення: більшість (71% в 2015 році) проектів з розробки програмного забезпечення не виконуються в заданий термін [3-7]. В існуючих інформаційних технологіях переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення використовуються методи та моделі, які були побудовані без врахування реального розподілу емпіричних даних. Наприклад, метод PERT базується на припущенні про бета-розподіл, а моделі COCOMO та ISBSG – на припущенні про логнормальний розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення, хоча ці припущення не завжди виконуються [9, 10, 47-56]. Це призводить до зниження достовірності оцінювання тривалості проектів з розробки програмного забезпечення. Крім того вказані інформаційні технології мають відносно високу вартість (Comparative Estimating Tool – 374,32 USD, Microsoft Project Standard 2016 – 589,99 USD станом на 01.06.2016). Тому наукова задача створення нових технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення з метою підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення є актуальною та має практичну цінність.

В 2 розділі на основі нормалізуючого перетворення Джонсона було побудовано негаусівські ймовірнісні моделі тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe. В 3 розділі на основі емпіричних даних ISBSG з використанням нормалізуючого перетворення Джонсона було побудовано нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів та їх довірчі інтервали для платформ PC, mid-range, mainframe. Використання зазначених моделей дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. Тому доцільним є створення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення на основі математичних моделей, побудованих з використанням нормалізуючого перетворення Джонсона. Крім того, після отримання нових емпіричних даних про завершені проекти, існує можливість уточнення параметрів математичних моделей, що були побудовані в розділах 1 і 2, що дозволить підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення за рахунок використання уточнених параметрів. Для цього необхідно створити інженерну методику статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.

4.1 Автоматизована інформаційна система переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

Інформаційна технологія переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення є складовою автоматизованої інформаційної системи переробки інформації для оцінювання

тривалості проектів з розробки програмного забезпечення. Розглянемо визначення автоматизованої інформаційної системи та розробимо архітектуру автоматизованої системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення.

Автоматизована інформаційна система (automated information system) – це система, що реалізує інформаційну технологію виконання встановлених функцій і складається з персоналу і комплексу засобів автоматизації його діяльності [90, 91]. На практиці в середині автоматизованої системи виділяють кілька підсистем, необхідність такої дії зазвичай буває викликана організаційними і фінансовими причинами. Функціональна підсистема – складова частина автоматизованої системи, що реалізує одну або кілька близьких функцій. Носій функціональної ролі – користувач, безпосередньо зайнятий в автоматизованій діяльності та бере участь у виконанні певних функцій системи на певних етапах.

Архітектура автоматизованої системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення розробленої в рамках дисертаційного дослідження приведена на рис. 4.1.

Користувачі представлені функціональними ролями в системі:

– ОПР (особа, приймаюча рішення) – або менеджер, погоджує та затверджує умови контракту з замовником, включаючи тривалість проектів з розробки програмного забезпечення та відповідальність розробника за зрив запланованих термінів виконання;

– аналітик – виконує аналітичне обґрунтування і формує пропозицію щодо виділення необхідних ресурсів проекту, включаючи тривалість проекту з розробки програмного забезпечення;

– статистик – фахівець з питань статистичної обробки даних; виконує статистичну обробку даних та побудову математичних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів.



Рис. 4.1. Архітектура автоматизованої інформаційної системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

Підсистема зберігання даних складається з бази даних (БД) та системи управління базою даних (СУБД). Використовується для зберігання детальної інформації про виконані проекти з розробки програмного забезпечення та зберігання параметрів математичних моделей для різних платформ.

Підсистема введення та відображення даних призначена для:

- завантаження аналітиком емпіричних даних (інформації про завершені проекти) з файлів в базу даних;
- ручного введення аналітиком емпіричних даних (інформації про завершені проекти) в базу даних;
- перегляду та редагування аналітиком емпіричних даних, збережених в базі даних.

Підсистема розрахунку параметрів моделей призначена для:

- попередньої обробки статистиком емпіричних даних тривалості проектів з розробки програмного забезпечення, а також трудомісткості цих

проектів, з метою видалення даних, які значно відрізняються від основної вибірки;

- побудови статистиком ймовірнісних моделей тривалості проектів з розробки програмного забезпечення;

- побудови статистиком нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів.

Користувачами підсистеми є фахівці з питань статистичної обробки даних. Результатом роботи підсистеми є значення параметри негаусівських ймовірнісних моделей тривалості проектів з розробки програмного забезпечення; параметри нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів; параметри рівнянь нижньої та верхньої границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів. Вказані параметри зберігаються в базу даних для подальшого використання при оцінюванні тривалості проектів з розробки програмного забезпечення підсистемою оцінювання, яка наведена далі.

Підсистема оцінювання реалізує інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення, яка створюється в дисертаційній роботі. Підсистема оцінювання призначена для:

- оцінювання аналітиком середнього значення тривалості проекту з розробки програмного забезпечення за оцінкою трудомісткості цих проектів для платформ PC, mid-range, mainframe;

- визначення аналітиком 95% довірчого інтервалу отриманої оцінки середнього значення тривалості проекту з розробки програмного забезпечення;

- визначення аналітиком 95% інтервалу прогнозування тривалості проекту з розробки програмного забезпечення;

– моделювання аналітиком тривалості проектів з розробки програмного забезпечення;

– збереження аналітиком в базу даних отриманої оцінки середнього значення тривалості проекту з розробки програмного забезпечення; збереження значень нижньої та верхньої границь довірчого інтервалу середнього значення та інтервалу прогнозування тривалості проектів з розробки програмного забезпечення; збереження в базу даних змодельованих значень тривалості проектів з розробки програмного забезпечення;

– перегляд ОПР результатів попередніх оцінювань та моделювань тривалості проектів з розробки програмного забезпечення.

Побудуємо функціональну модель підсистеми оцінювання автоматизованої системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення використовуючи методологію DFD (Data Flow Diagrams) – діаграми потоків даних [92], що призначена для проектування інформаційних систем [93, 94]. Основу методології DFD становить графічна мова опису процесів в нотаціях Йордана Де-Марко або Гейна-Сарсона. Нині найбільш поширеною є нотація Гейна-Сарсона. Діаграма потоків даних автоматизованої системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення в нотації Гейна-Сарсона показана на рис. 4.2.

Складні потоки даних діаграми 4.2. мають наступні атрибути:

– параметри нелінійної регресійної моделі – параметри γ_D , η_D , ϕ_D , λ_D , γ_E , η_E , ϕ_E , λ_E , b_0 , b_1 , $s_{z_D}^2$, \bar{z}_E , S_{z_E} , n моделей (3.11), (3.12);

– параметри негаусівської ймовірнісної моделі – параметри γ , η , ϕ , λ моделі (2.4);

– довірчий інтервал тривалості проекту з розробки програмного забезпечення – нижня границя довірчого інтервалу тривалості $[\bar{D}(E)]$ (3.11), верхня границя довірчого інтервалу тривалості $[\bar{D}(E)]$ (3.11);

– інтервал прогнозування тривалості проекту з розробки програмного забезпечення – нижня границя інтервалу прогнозування тривалості $[D(E)]$ (3.12), верхня границя довірчого інтервалу тривалості $[D(E)]$ (3.12).



Рис. 4.2. Діаграма потоків даних підсистеми оцінювання тривалості проектів з розробки програмного забезпечення

4.2 Створення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

Термін “інформаційна технологія” має декілька визначень. Наприклад, [90, 99, 100] наголошують на системі процесів, методів і способів переробки інформації. Інше визначення цього терміну, яке прийняте в даному

дослідженні, наведено в [101, 102], де інформаційна технологія визначається як комплекс методів, способів і засобів, що забезпечує збереження, обробку, передачу і відображення інформації. Розглянемо детальніше мету створення, вирішувані задачі та відповідний комплекс інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного.

Мета створення інформаційної технології

Метою створення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення є підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення за рахунок використання математичних моделей тривалості проектів з розробки програмного забезпечення [47-56], які наведені в розділах 2 та 3.

Задачі, які вирішуються за допомогою інформаційної технології

Перелік задач, які вирішуються за допомогою пропонованої інформаційної технології повністю співпадає з відповідним переліком задач підсистеми оцінювання автоматизованої інформаційної системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення. Інформаційна технологія переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення призначена для вирішення наступних задач:

- оцінювання аналітиком середнього значення тривалості проекту з розробки програмного забезпечення за оцінкою трудомісткості цих проектів для платформ PC, mid-range, mainframe;

- визначення аналітиком 95% довірчого інтервалу отриманої оцінки середнього значення тривалості проекту з розробки програмного забезпечення;

- визначення аналітиком 95% інтервалу прогнозування тривалості проекту з розробки програмного забезпечення;

- моделювання аналітиком тривалості проектів з розробки програмного забезпечення;

– збереження аналітиком в базу даних отриманої оцінки середнього значення тривалості проекту з розробки програмного забезпечення; збереження значень нижньої та верхньої границь довірчого інтервалу середнього значення та інтервалу прогнозування тривалості проектів з розробки програмного забезпечення; збереження в базу даних змодельованих значень тривалості проектів з розробки програмного забезпечення;

– перегляд ОПР результатів попередніх оцінювань та моделювань тривалості проектів з розробки програмного забезпечення.

Математичне забезпечення інформаційної технології

До математичного забезпечення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення належать математичні моделі тривалості проектів з розробки програмного забезпечення (негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення (2.4), яка була побудована в 2 розділі; нелінійна регресійна модель тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів (3.9), яка була побудована в 3 розділі; рівняння нижньої та верхньої границь довірчого інтервалу (3.11) та інтервалу прогнозування (3.12) нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів, які були побудовані в 3 розділі), а також метод виключень для генерації значень випадкової величини з заданим законом розподілу [103], що використовується для моделювання тривалості проектів з розробки програмного забезпечення.

Метод виключень для генерації значень випадкової величини з заданим законом розподілу [103] полягає у наступному. Нехай потрібно згенерувати значення випадкової величини з законом розподілу, який заданий щільністю ймовірності $f(x)$. Для цього спочатку генерується пара значень випадкових чисел $x_i \in (a, b)$ та $y_i \in (0, f_{\max})$ з рівномірним законом розподілу, де f_{\max} – це максимальне значення щільності ймовірності $f(x)$ в діапазоні $x_i \in (a, b)$.

Якщо виконується нерівність $y_i \leq f(x_i)$, то значення y_i приймається як наступне значення випадкової величини з законом розподілу, який заданий щільністю ймовірності $f(x)$. Інакше – значення x_i і y_i відкидаються і генерується наступна пара значень.

Інформаційне забезпечення інформаційної технології

До інформаційного забезпечення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення входить база даних, що призначена для збереження результатів оцінювань та моделювань тривалості проектів з розробки програмного забезпечення, а також для збереження параметрів негаусівських ймовірнісних моделей тривалості проектів з розробки програмного забезпечення для різних платформ; параметрів нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів (3.9) для різних платформ; параметрів рівнянь нижньої та верхньої границь довірчого інтервалу (3.11) та інтервалу прогнозування (3.12) нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів для різних платформ. Необхідність збереження параметрів математичних моделей тривалості проектів з розробки програмного забезпечення в базі даних викликана вимогою можливості уточнення параметрів математичних моделей після надходження нових емпіричних даних тривалості проектів з розробки програмного забезпечення та використання уточнених параметрів в програмному забезпеченні без його перекомпіляції.

Побудуємо логічну модель бази даних інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення. Для цього для кожної таблиці в базі даних вкажемо перелік атрибутів та їх тип даних, а також первинний ключ.

Логічна модель таблиці параметрів математичних моделей тривалості проектів з розробки програмного забезпечення для різних платформ:

- ідентифікатор запису – числовий тип даних, первинний ключ;
- назва платформи – строковий тип даних, заборонені дублювання;;

- код платформи – строковий тип даних, заборонені дублювання;
- параметр γ_D моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр η_D моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр ϕ_D моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр λ_D моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр γ_E моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр η_E моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр ϕ_E моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр λ_E моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр b_0 моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр b_1 моделей (3.9), (3.11), (3.12) – числовий тип даних;
- параметр $s_{z_D}^2$ моделей (3.11), (3.12) – числовий тип даних;
- параметр \bar{z}_E моделей (3.11), (3.12) – числовий тип даних;
- параметр S_{z_E} моделей (3.11), (3.12) – числовий тип даних;
- параметр n моделей (3.11), (3.12) – числовий тип даних.

Логічна модель таблиці результатів оцінювань та моделювань тривалості проектів з розробки програмного забезпечення для різних платформ:

- ідентифікатор запису – числовий тип даних, первинний ключ;
- назва проекту – строковий тип даних, заборонені дублювання;
- код платформи – строковий тип даних;
- трудомісткість – числовий тип даних;
- тривалість – числовий тип даних;
- нижня границя довірчого інтервалу тривалості – числовий тип даних;
- верхня границя довірчого інтервалу тривалості – числовий тип даних;
- нижня границя інтервалу прогнозування тривалості – числовий тип даних;
- верхня границя інтервалу прогнозування тривалості – числовий тип даних;

- змодельовані значення тривалості – строковий тип даних;
- дата оцінювання – дата.

Для доступу до бази даних використовується система управління базами даних SQLite. Вибір вказаної СУБД пов'язаний з невеликою кількістю даних, які необхідно зберігати в базі даних та легкістю встановлення та налаштування СУБД SQLite. Крім того, при необхідності розширення, вказану СУБД можна буде легко змінити на більш потужну СУБД (MySQL чи PostgreSQL) не змінюючи запити, тому що СУБД SQLite використовує мову запитів SQL, хоча фізично всі данні зберігаються в одному бінарному файлі.

Побудуємо фізичну модель бази даних інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення з врахуванням побудованої логічної моделі і того, що буде використовуватися СУБД SQLite. Для цього для атрибутів кожної таблиці в базі даних вкажемо їх тип даних при використанні СУБД SQLite.

Фізична модель таблиці параметрів математичних моделей тривалості проектів з розробки програмного забезпечення для різних платформ:

- id – INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT;
- title – TEXT NOT NULL UNIQUE;
- platform – TEXT NOT NULL UNIQUE;
- d_gamma – NUMERIC NOT NULL;
- d_nu – NUMERIC NOT NULL;
- d_fi – NUMERIC NOT NULL;
- d_lamda – NUMERIC NOT NULL;
- e_gamma – NUMERIC NOT NULL;
- e_nu – NUMERIC NOT NULL;
- e_fi – NUMERIC NOT NULL;
- e_lamda – NUMERIC NOT NULL;
- b0 – NUMERIC NOT NULL;
- b1 – NUMERIC NOT NULL;
- zd_sse – NUMERIC NOT NULL;

- ze_avg – NUMERIC NOT NULL;
- ze_S – NUMERIC NOT NULL;
- n – INTEGER NOT NULL.

Фізична модель таблиці результатів оцінювань та моделювань тривалості проектів з розробки програмного забезпечення для різних платформ:

- id – INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT;
- title – TEXT NOT NULL UNIQUE;
- platform – TEXT NOT NULL;
- effort – NUMERIC NOT NULL;
- duration – NUMERIC NOT NULL;
- duration_confidence_interval_bottom – NUMERIC NOT NULL;
- duration_confidence_interval_top – NUMERIC NOT NULL;
- duration_estimation_interval_bottom – NUMERIC NOT NULL;
- duration_estimation_interval_top – NUMERIC NOT NULL;
- generated_duration – TEXT NOT NULL;
- created – DATE NOT NULL.

Програмне забезпечення інформаційної технології

Програмним забезпеченням інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення є комп'ютерна програма з візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62], віртуальна машина Java SE Runtime Environment 6 та сумісна з Java операційна система (Windows, Mac, unix).

Розглянемо детальніше комп'ютерну програму з візуальним інтерфейсом «Оцінювання тривалості розробки програмного забезпечення» [62], яка була розроблена в рамках даного дослідження як програмний засіб інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення з метою програмної реалізації математичного забезпечення цієї інформаційної технології.

В якості мови програмування для розробки програми була обрана кросплатформна об'єктно-орієнтована мова програмування Java.

Критерії, використовувані при виборі мови програмування:

- можливість роботи програми для оцінювання тривалості проектів з розробки програмного забезпечення в різних операційних системах;
- наявність необхідного набору візуальних компонентів для формування ергономічного інтерфейсу з невеликим часом відклику;
- існування математичних бібліотек, необхідних для виконання оцінювання і моделювання тривалості проектів з розробки програмного забезпечення з використанням математичних моделей, які входять до математичного забезпечення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення;
- наявність драйверів для роботи з системою управління базами даних SQLite, яка входить до інформаційного забезпечення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення;
- простота доступу до бази даних за рахунок використання однієї з реалізацій Java Persistence API – специфікації прикладного програмного інтерфейсу, яка описує роботу з реляційними даними в Java-програмах;
- можливість переключення програми для оцінювання тривалості проектів з розробки програмного забезпечення на роботу з іншими системами управління базами даних за рахунок використання для доступу до бази даних мови запитів Java Persistence Query Language – платформи-незалежної об'єктно-орієнтованої мови запитів, що є частиною специфікації Java Persistence API і дозволяє не змінювати запити при зміні системи управління базами даних;
- наявність безкоштовних IDE (Integrated Development Environment) – інтегрованих середовищ розробки прикладних програм;
- дозвіл на безкоштовне використання в комерційних проектах та можливість розповсюдження Java разом з програмою.

В зв'язку з вибором об'єктно-орієнтованої мови програмування Java, для того щоб показати структуру класів програми використаємо UML (Unified Modeling Language) [95-98] – стандартну мову для написання моделі аналізу, проектування і реалізації об'єктно-орієнтованих програмних систем.

Діаграма класів комп'ютерної програми «Оцінювання тривалості розробки програмного забезпечення» показана на рис. 4.3.



Рис. 4.3. Діаграма класів комп'ютерної програми з візуальним інтерфейсом «Оцінювання тривалості розробки програмного забезпечення»

При запуску даної програми спочатку викликається метод `EstimationApp::main()`, в якому відбувається підключення до бази даних та створюється і відображається екземпляр головного вікна програми. При цьому до головного вікна програми додається пункт меню для створення і відображення вікна перегляду збережених результатів оцінювання і моделювання тривалості проектів з розробки програмного забезпечення. В головному вікні створюється екземпляр класу `EstimationResults` – в цьому класі реалізовано математичне забезпечення інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення, а метод `EstimationResults::fillResults()` дозволяє виконати оцінювання та моделювання тривалості проектів з розробки програмного забезпечення та відобразити результати в головному вікні програми.

Діаграма станів комп'ютерної програми «Оцінювання тривалості розробки програмного забезпечення» показана на рис .4.4. Основним станом даної комп'ютерної програми є відображення головного вікна, під час якого відбувається очікування дій користувача. Перехід до інших станів відбувається в залежності від виконаних дій користувача, що відображено на рис. 4.4.

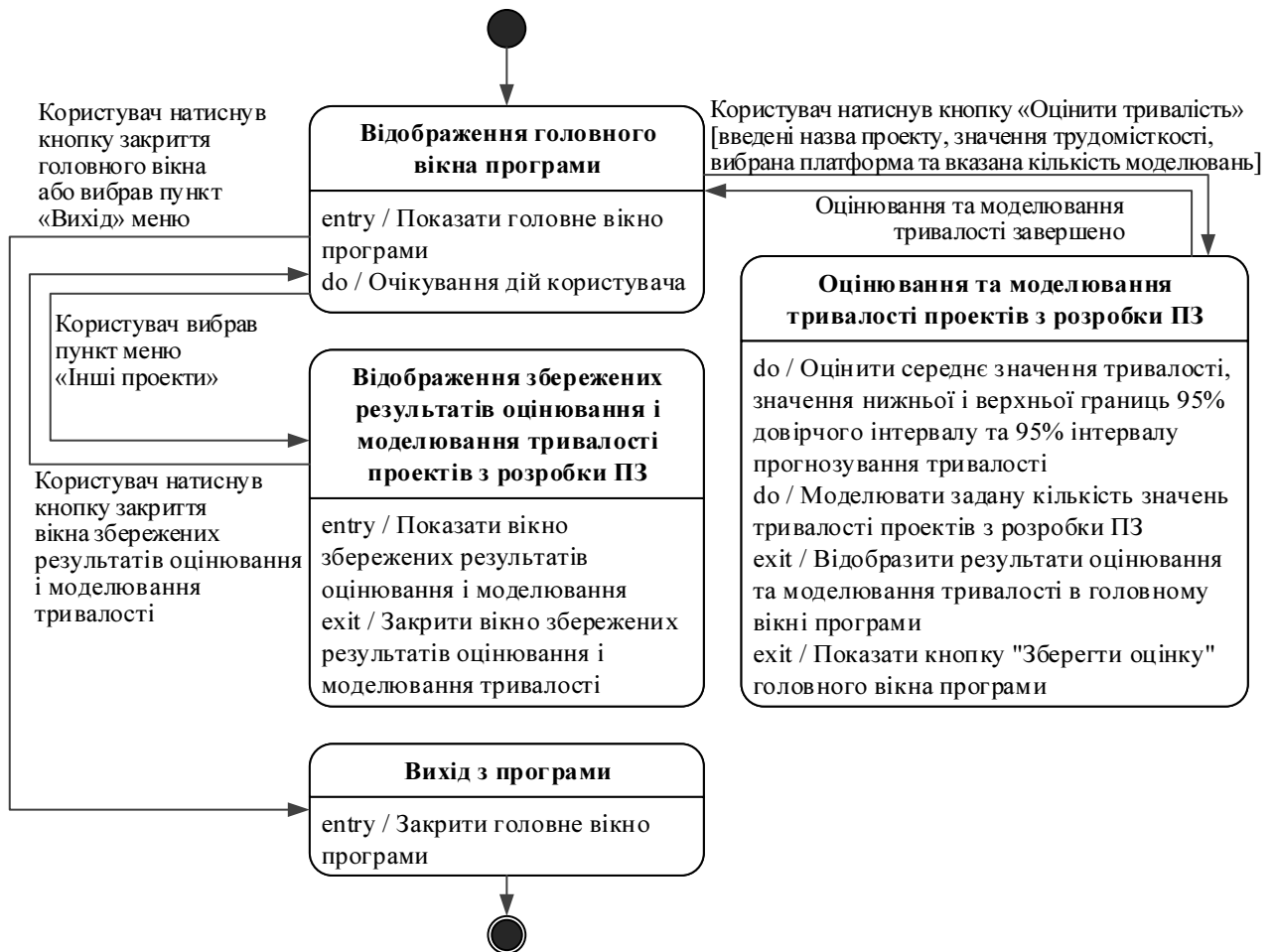


Рис. 4.4. Діаграма станів комп'ютерної програми з візуальним інтерфейсом «Оцінювання тривалості розробки програмного забезпечення»

Технічне забезпечення інформаційної технології

Технічним забезпеченням інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення є ПК в такій мінімальній конфігурації:

- процесор Intl Pentium (2 GHz);
- ОЗП 1 Gb;
- ПЗП 10 Гб;
- монітор 800x600 точок;
- клавіатура;
- маніпулятор «миша».

Методичне забезпечення інформаційної технології

Методичним забезпеченням інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення є методика оцінювання тривалості проектів з розробки програмного забезпечення з використанням відповідного програмного забезпечення – комп'ютерної програми з візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62], розробленої в рамках даного дослідження. Настанова користувача цієї комп'ютерної програми приведена в додатку Б, а вигляд головного вікна програми показаний на рис. 4.5.

Параметер	Оцінка
Назва	Проект 1
Платформа	Мейнфрейми (MF)
Трудомісткість	1000 людино-годин
Оцінювання тривалості:	
Оцінка тривалості	6.5 місяців
95% Довірчий інтервал	[5.9, 7.1] місяців
95% інтервал передбачення	[2.1, 19.2] місяців
Моделювання тривалості:	
1.	6.1 місяців
2.	3.6 місяців
3.	11.4 місяців
4.	18.9 місяців
5.	11.3 місяців

Рис. 4.5. Головне вікно комп'ютерної програми з візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62]

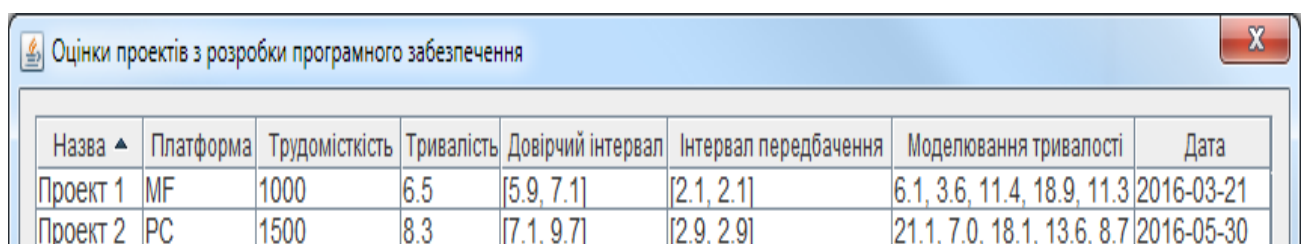
Методика оцінювання тривалості проектів з розробки програмного забезпечення

1. Аналітик збирає вхідні данні для оцінювання тривалості проекту з розробки програмного забезпечення: назву, платформу, оцінку трудомісткості.
2. Аналітик вмикає ПК та чекає завантаження операційної системи.
3. Аналітик запускає комп'ютерну програму з візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62] (далі – програму).
4. Аналітик вводить зібрані вхідні данні у відповідні поля головного вікна програми (рис. 4.5).
5. Аналітик вводить необхідну кількість моделювань у відповідне поле головного вікна програми (рис. 4.5).
6. Аналітик натискає на кнопку «Оцінити тривалість» в головному вікні програми.
7. Програмою завантажуються параметри математичних моделей з бази даних.
8. Програмою обчислюється та виводиться на екран оцінка середнього значення тривалості, значення нижньої і верхньої границь 95% довірчого інтервалу отриманої оцінки середнього значення тривалості та значення нижньої і верхньої границь 95% інтервалу прогнозування (рис. 4.5) з використанням нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від їх трудомісткості
9. Також програма виведе задану кількість змодельованих значень тривалості проектів з розробки програмного забезпечення.
10. При необхідності аналітик зберігає результати оцінювання і моделювання в базу даних, натиснувши на кнопку «Зберегти оцінку», яка з'явилася в головному вікні програми після виконання попереднього пункту (рис. 4.5).
11. Аналітик закриває програму, вибравши пункт меню «Вихід» вверху головного вікна програми або натиснувши кнопку закриття головного вікна програми, яка знаходиться серед елементів управління вікном операційної системи.

12. При необхідності, аналітик вимикає ПК, використовуючи відповідні елементи управління операційної системи.

Для перегляду результатів оцінювання і моделювання виконується наступна послідовність дій:

1. ОПР вмикає ПК та чекає завантаження операційної системи.
2. ОПР запускає комп'ютерну програму з візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62] (далі – програму).
3. ОПР вибирає пункт меню «Інші проекти» вверху головного вікна програми, після чого буде показано вікно перегляду збережених результатів оцінювання і моделювання тривалості (рис. 4.6).
4. У відкритому вікні ОПР переглядає збережені результати оцінювання і моделювання тривалості проектів з розробки програмного забезпечення. Результати оцінювання і моделювання використовуються при складанні контракту з замовником програмного забезпечення.
5. ОПР закриває вікно перегляду збережених результатів оцінювання натиснувши кнопку закриття цього вікна, яка знаходиться серед елементів управління вікном операційної системи.
6. ОПР закриває програму, вибравши пункт меню «Вихід» вверху головного вікна програми або натиснувши кнопку закриття головного вікна програми, яка знаходиться серед елементів управління вікном операційної системи.
7. При необхідності, ОПР вимикає ПК, використовуючи відповідні елементи управління операційної системи.



Назва ▲	Платформа	Трудомісткість	Тривалість	Довірчий інтервал	Інтервал передбачення	Моделювання тривалості	Дата
Проект 1	MF	1000	6.5	[5.9, 7.1]	[2.1, 2.1]	6.1, 3.6, 11.4, 18.9, 11.3	2016-03-21
Проект 2	PC	1500	8.3	[7.1, 9.7]	[2.9, 2.9]	21.1, 7.0, 18.1, 13.6, 8.7	2016-05-30

Рис. 4.6. Вікно перегляду збережених результатів оцінювання і моделювання тривалості проектів з розробки програмного забезпечення

Послідовність дій інформаційної технології

Діаграма станів інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення показана на рис. 4.7.



Рис. 4.7. Діаграма станів інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

Для виконання оцінювання тривалості проектів з розробки програмного забезпечення виконується наступна послідовність дій:

1. Аналітик збирає вхідні данні для оцінювання тривалості проекту з розробки програмного забезпечення: назву, платформу, оцінку трудомісткості.
2. Аналітик вмикає ПК та чекає завантаження операційної системи.

3. Аналітик запускає комп'ютерну програму з візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62] (далі – програму).
4. Аналітик вводить зібрані вхідні данні у відповідні поля головного вікна програми (рис. 4.5).
5. Аналітик вводить необхідну кількість моделювань у відповідне поле головного вікна програми (рис. 4.5).
6. Аналітик натискає на кнопку «Оцінити тривалість» в головному вікні програми.
7. Програмою з бази даних завантажуються параметри негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення (2.4) для вибраної платформи; параметри нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості (3.9) для вибраної платформи; параметри рівнянь нижньої та верхньої границь довірчого інтервалу (3.11) та інтервалу прогнозування (3.12) нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості для вибраної платформи.
8. Програмою, з використанням нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості, обчислюється та виводиться на екран оцінка середнього значення тривалості програмного забезпечення.
9. Програмою, з використанням рівнянь нижньої та верхньої границь довірчого інтервалу (3.11) та інтервалу прогнозування (3.12) нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості, обчислюються та виводяться на екран значення нижньої і верхньої границь 95% довірчого інтервалу отриманої оцінки середнього значення тривалості та значення нижньої і верхньої границь 95% інтервалу прогнозування.

- 10.Програмою, з використанням негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення (2.4) та методу виключень, моделюється та виводиться на екран задана кількість значень тривалості проектів з розробки програмного забезпечення.
- 11.При необхідності збереження результатів оцінювання і моделювання в базу даних, аналітик натискає на кнопку «Зберегти оцінку» програми, після чого програма зберігає в базу даних результати оцінювання і моделювання тривалості проектів з розробки програмного забезпечення.
- 12.Аналітик закриває програму, вибравши пункт меню «Вихід» вверху головного вікна програми або натиснувши кнопку закриття головного вікна програми, яка знаходиться серед елементів управління вікном операційної системи.
- 13.При необхідності, аналітик вимикає ПК, використовуючи відповідні елементи управління операційної системи.

Для перегляду результатів оцінювання і моделювання виконується наступна послідовність дій:

1. ОПР вмикає ПК та чекає завантаження операційної системи.
2. ОПР запускає комп'ютерну програму із візуальним інтерфейсом "Оцінювання тривалості розробки програмного забезпечення" [62] (далі – програму).
3. ОПР вибирає пункт меню «Інші проекти» вверху головного вікна програми.
4. Програма завантажує з бази даних збережені результати оцінювання і моделювання тривалості проектів з розробки програмного забезпечення.
5. Програмою відображається вікно перегляду збережених результатів оцінювання і моделювання тривалості з завантаженими даними (рис. 4.6).
6. У відкритому вікні програми ОПР переглядає збережені результати оцінювання і моделювання тривалості проектів з розробки програмного забезпечення. Результати оцінювання і моделювання використовуються при складанні контракту з замовником програмного забезпечення.

7. ОПР закриває вікно перегляду збережених результатів оцінювання натиснувши кнопку закриття цього вікна, яка знаходиться серед елементів управління вікном операційної системи.
8. ОПР закриває програму, вибравши пункт меню «Вихід» вверху головного вікна програми або натиснувши кнопку закриття головного вікна програми, яка знаходиться серед елементів управління вікном операційної системи.
9. При необхідності, ОПР вимикає ПК, використовуючи відповідні елементи управління операційної системи.

Впровадження інформаційної технології

Інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення було впроваджено у виробничі процеси ТОВ «Макротел» (акт впровадження від 21.05.2016), ТОВ «Вебкодерс» (акт впровадження від 07.11.2016). Впровадження запропонованої інформаційної технології переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення дозволило підвищити достовірність оцінювання тривалості проектів для платформ PC, mid-range, mainframe. Також, інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення було впроваджено в навчальний процес кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування імені адмірала Макарова шляхом використання в робочій навчальній програмі з дисципліни «Емпіричні методи програмної інженерії» (акт впровадження від 02.11.2016). Акти впровадження результатів дисертаційної роботи наведені в додатку А.

4.3 Методика статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення

Для математичних моделей, що були побудовані в розділах 1 і 2, та використані при створенні інформаційної технології з переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення в підрозділі 4.2, існує можливість уточнення параметрів моделей після отримання нових емпіричних даних про завершені проекти, що дозволить підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення за рахунок використання уточнених параметрів математичних моделей. Для цього, з певною періодичністю (наприклад, кожні шість місяців), статист має виконувати статистичну обробку емпіричних даних з врахуванням нових емпіричних даних тривалості і трудомісткості проектів з розробки програмного забезпечення: спочатку провести попередню обробку вказаних емпіричних даних, після чого визначити нові параметри негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення (2.4), параметри нелінійної регресійної моделі (3.9) і параметри рівнянь нижньої та верхньої границь довірчого інтервалу (3.11) та інтервалу прогнозування (3.12) нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від їх трудомісткості за методикою, приведеною далі в цьому підрозділі. Моделі з уточненими параметрами в подальшому використовуються при оцінюванні тривалості проектів з розробки програмного забезпечення у відповідній інформаційній технології з переробки інформації, для чого уточнені параметри потрібно внести в базу даних.

Приведемо методику статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.

При уточненні параметрів математичних моделей тривалості проектів з розробки програмного забезпечення в якості випадкової величини приймаються

емпіричні значення тривалості проектів з розробки програмного забезпечення (одиниця вимірювання – місяці). При уточненні параметрів нелінійних регресійних моделей і параметрів рівнянь довірчого інтервалу та інтервалу прогнозування додатково використовуються емпіричні значення трудомісткості проектів з розробки програмного забезпечення (одиниця вимірювання – людино-години).

Методика статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення

1. Групування емпіричних даних за платформою, для якої розроблялося програмне забезпечення.

Тривалість проектів з розробки програмного забезпечення значною мірою залежить від платформи, для якої розробляється програмне забезпечення. Основними платформами є PC, mid-range, mainframe [10, 11]. Тому емпіричні дані тривалості та трудомісткості проектів з розробки програмного забезпечення групуються за платформою, для якої розроблялося програмне забезпечення. Дії, вказані в наступних пунктах даної методики потрібно виконувати окремо для кожної з платформ, для якої доступні нові емпіричні дані.

2. Розрахунок статистичних параметрів вибірок емпіричних даних.

Для вибірок емпіричних значень тривалості та трудомісткості проектів з розробки програмного забезпечення розраховуються такі параметри: кількість значень n , вибіркове середнє \bar{x} за формулою (2.6), дисперсія D за формулою (2.7), асиметрія A за формулою (2.8), ексцес ε за формулою (2.9).

3. Перевірка можливості застосування розподілу та перетворення Джонсона сім'ї S_B .

По діаграмі Джонсона (рис. 2.2), за значеннями знайдених в п. 2 оцінок асиметрії у квадраті A^2 та ексцесу ε , виконується перевірка можливості застосування розподілу та перетворення Джонсона сім'ї S_B (2.4): точка з координатами (A^2, ε) повинна знаходитися нижче лінії S_L до лінії критичної області.

4. Уточнення параметрів негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення.

Для уточнення параметрів негаусівської ймовірнісної моделі тривалості проектів з розробки програмного забезпечення в результаті рішення задачі (2.14) знаходяться параметри γ_D , η_D , φ_D , λ_D розподілу Джонсона сім'ї S_B для вибірок емпіричних значень тривалості та трудомісткості проектів з розробки програмного забезпечення.

5. Нормалізація емпіричних даних.

Виконується нормалізація емпіричних даних тривалості проектів з розробки програмного забезпечення з використанням відповідного для вибраної в п. 3 сім'ї нормалізуючого перетворення Джонсона (2.15) зі знайденими в п. 4 параметрами γ_D , η_D , φ_D , λ_D . Додатково, в результаті рішення задачі (2.14), знаходяться параметри γ_E , η_E , φ_E , λ_E нормалізуючого перетворення Джонсона (2.15) для нормалізації емпіричних даних трудомісткості тривалості проектів з розробки програмного забезпечення. Виконується перевірка нормалізованих даних на відповідність нормальному закону розподілу. В якості критерію згоди використовується критерій Пірсона χ^2 .

6. Уточнення параметрів нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів.

Використовуючи (3.6), за нормалізованими в п. 5 емпіричними значеннями тривалості і трудомісткості проектів з розробки програмного забезпечення, знаходяться значення уточнених параметрів b_0 і b_1 нелінійної регресійної моделі тривалості проектів з розробки програмного забезпечення в залежності від нормалізованих значень трудомісткості цих проектів. Крім того, знайдені в п. 5 значення параметрів γ_D , η_D , φ_D , λ_D та γ_E , η_E , φ_E , λ_E також є уточненими параметрами даної нелінійної регресійної моделі.

7. Уточнення параметрів рівнянь границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів.

Використовуючи (3.7) та (3.8), за нормалізованими в п. 5 емпіричними значеннями тривалості і трудомісткості проектів з розробки програмного забезпечення, знаходяться уточнені значення параметрів $s_{z_D}^2$, \bar{z}_E , S_{z_E} рівнянь границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів. Крім того, знайдені в п. 6 значення параметрів b_0 і b_1 , і знайдені в п. 5 значення параметрів γ_D , η_D , ϕ_D , λ_D та γ_E , η_E , ϕ_E , λ_E також є уточненими параметрами даної нелінійної регресійної моделі.

8. Збереження уточнених параметрів математичних моделей тривалості проектів з розробки програмного забезпечення в базу даних.

Необхідно зберегти в базу даних для даної платформи нове значення n ; зберегти уточнені в п. 5 значення параметрів γ_D , η_D , ϕ_D , λ_D та γ_E , η_E , ϕ_E , λ_E ; зберегти уточнені в п. 6 значення параметрів b_0 і b_1 ; зберегти уточнені в п. 7 значення параметрів $s_{z_D}^2$, \bar{z}_E , S_{z_E} . Для цього в базі даних автоматизованої системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення потрібно виконати наступний запит:

```
UPDATE duration_models SET
```

```
`d_gamma`= $\gamma_D$ , `d_nu`= $\eta_D$ , `d_phi`= $\phi_D$ , `d_lambda`= $\lambda_D$ ,
```

```
`e_gamma`= $\gamma_E$ , `e_nu`= $\eta_E$ , `e_phi`= $\phi_E$ , `e_lambda`= $\lambda_E$ ,
```

```
`b0`= $b_0$ , `b1`= $b_1$ , `zd_sse`= $s_{z_D}^2$ , `ze_avg`= $\bar{z}_E$ , `ze_S`= $S_{z_E}$ , `n`=n
```

```
WHERE `platform`="КОД_ПЛАТФОРМИ";,
```

де КОД_ПЛАТФОРМИ = $\begin{cases} PC, & \text{для платформи PC;} \\ MR, & \text{для платформи mid - range;} \\ MF, & \text{для платформи mainframe.} \end{cases}$

4.4 Висновки до розділу 4

В даному розділі було створено інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення із застосуванням моделей, побудованих з використанням нормалізуючого перетворення Джонсона.

1. Розроблено архітектуру автоматизованої інформаційної системи переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення, визначено функціональні підсистеми та функціональні ролі цієї автоматизованої інформаційної системи.
2. Створено інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення, яка дозволяє підвищити достовірність оцінювання тривалості цих проектів. Основою побудови пропонованої інформаційної технології послужили математичні моделі, розроблені автором в даному дослідженні.
3. Створено інженерну методику статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.

Результати даного розділу були опубліковані в [61, 85].

ВИСНОВКИ

В дисертаційній роботі було вирішене науково-практичне завдання підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення.

В процесі проведення дисертаційних досліджень одержані наступні результати:

1. Проаналізовано сучасний стан інформаційних технологій переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення; досліджено особливості, переваги та недоліки існуючих моделей та методів оцінювання тривалості проектів з розробки програмного забезпечення; визначено задачі, які необхідно вирішити для досягнення мети дослідження – підвищення достовірності оцінювання тривалості проектів з розробки програмного забезпечення.
2. Удосконалено негаусівську ймовірнісну модель тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe за рахунок застосування щільності ймовірності Джонсона, що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. Ця ймовірнісна модель краще апроксимує розподіл емпіричних даних тривалості проектів з розробки програмного забезпечення, ніж бета-розподіл за критерієм Пірсона χ^2 .
3. Зроблено порівняння перетворення у вигляді десяткового логарифму, перетворення Бокса-Кокса та перетворення Джонсона для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення. Вибрано нормалізуюче перетворення Джонсона сім'ї S_B для нормалізації емпіричних даних тривалості та трудомісткості проектів з розробки програмного забезпечення.

4. Отримали подальший розвиток нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів (модель ISBSG) для платформ PC, mid-range, mainframe за рахунок застосування нормалізуючого перетворення Джонсона сім'ї S_B що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. За критеріями R^2 , SSE та $PRED(25\%)$ побудовані рівняння нелінійної регресії на основі перетворення Джонсона є кращими, ніж відповідні рівняння на основі перетворення у вигляді десяткового логарифму. Відносна похибка для побудованих нелінійних регресійних моделей тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів не перевищує 15%.
5. Побудовано рівняння нижньої та верхньої границь довірчих інтервалів нелінійних регресій тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe за рахунок застосування нормалізуючого перетворення Джонсона сім'ї S_B , що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. В залежності від платформи ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення Джонсона є на 10-13% меншою, ніж ширина 95% довірчого інтервалу нелінійної регресії на основі перетворення у вигляді десяткового логарифму.
6. Побудовано рівняння нижньої та верхньої границь інтервалів прогнозування нелінійних регресій тривалості проектів з розробки програмного забезпечення для платформ PC, mid-range, mainframe за рахунок застосування нормалізуючого перетворення Джонсона сім'ї S_B , що дозволяє підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення. В залежності від платформи ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення

Джонсона є на 10-25% меншою, ніж ширина 95% інтервалу прогнозування нелінійної регресії на основі перетворення у вигляді десяткового логарифму.

7. Розроблено архітектуру автоматизованої інформаційної системи та створено інформаційну технологію переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення із застосуванням математичних моделей, побудованих з використанням нормалізуючого перетворення Джонсона сім'ї S_B . Розроблено відповідне програмне забезпечення – комп'ютерну програму з візуальним інтерфейсом «Оцінювання тривалості розробки програмного забезпечення».
8. Створено інженерну методику статистичної обробки емпіричних даних для оцінювання тривалості проектів з розробки програмного забезпечення.
9. Практичне значення одержаних результатів дослідження підтверджується їх використанням в ТОВ «Макротел» та «Вебкодерс», що дозволило підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення на вказаних підприємствах. Також результати дослідження впроваджено у навчальний процес кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування ім. адмірала Макарова при викладанні дисципліни «Емпіричні методи програмної інженерії».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДеМарко, Т. Вальсируя с Медведями: управление рисками в проектах по разработке программного обеспечения [Текст] / Т. ДеМарко, Т. Листер. – Компания p.m.Office, 2005. – 196 с. – ISBN 5-902681-03-0
2. Temnenco, V., Software Estimation, Enterprise [Электронный ресурс] / V. Temnenco // Wide, IBM The Rational Edge. – Vol. June 2007. – Режим доступа: <http://www.ibm.com/developerworks/rational/library/jun07/temnenco/index.html>
3. Ройс, У. Управление проектами по созданию программного обеспечения. Унифицированный подход [Текст] / У. Ройс. – Москва : "Лори", 2002. – 424 с. – ISBN 5-85582-156-0.
4. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch [Электронный ресурс] // S. Hastie, S. Wojewoda. – InfoQ, 2015. – Режим доступа : <https://www.infoq.com/articles/standish-chaos-2015>.
5. Chow, A. W. Engaging a corporate community to manage technology and embrace innovation [Text] / A. W. Chow, B. D. Goodman, J. W. Rooney, C. D. Wyble. // IBM Systems Journal. – Vol. 46, No. 4, 2007. – P. 639 - 650.
6. Guide to the Software Engineering Body of Knowledge [Text] – Los Alamitos, CA : IEEE Computer Society, 2004. – 204 p. – ISBN 0-7695-2330-7.
7. McConnell, S. Rapid Development: Taming Wild Software Schedules [Text] / S. McConnell. – Redmond, WA : Microsoft Press, 1996. – 660 p. – ISBN 1556159005.
8. A guide to the project management body of knowledge. Fifth edition [Text] – Project Management Institute, 2013. – 589 p. – ISBN 978-1-935589-67-9.

9. Boehm, B. W. Software engineering economics [Text] / B. W. Boehm. – Englewood Cliffs, NJ : Prentice Hall, 1981. – 768 p. – ISBN 0-13-822122-7.
10. Oligny, S. An empirical assessment of project duration models in software engineering [Text] / S. Oligny, P. Bourque, A. Abran. // In Proc. 8th European Software Control and Metrics Conference ESCOM. – Berlin, 1997.
11. Oligny, S. Exploring the relation between effort and duration in software engineering projects [Text] / S. Oligny, P. Bourque, A. Abran, B. Fournier // In proc. of the World Computer Congress, Aug. 2000. – Pp. 175-178.
12. Tayntor, C. B. Project Management Tools and Techniques for Success [Text] / C. B. Tayntor. – CRC Press, 2010. – 290 p. – ISBN 978-1439816318.
13. Vetrici, M. Software Project Duration Estimation Using Metrix Model [Text] / M. Vetrici. – Informatica Economica Journal. – Vol. 3, No. 47, 2008. – Pp. 87-91.
14. Vetrici, M. Knowledge based project duration estimation for workflow based document management software projects [Text] / M. Vetrici, C. Ioniță. – Economy Informatics. – No. 1-4, 2008. – Pp. 122-125.
15. Vetrici, M. Distributed collaborative software development process improvement using criticality analysis [Text] / M. Vetrici, C. Tomozei. – Economy Informatics. – Vol. 9, No. 1, 2009. – Pp. 71-78.
16. Boehm, B. W. Software Cost Estimation with COCOMO II. [Text] / B. W. Boehm. – Upper Saddle River, NJ : Prentice Hall PTR, 2000. – 586 p. – ISBN 0-13-026692-2.
17. Jørgensen, M. A review of studies on expert estimation of software development effort [Text] / M. Jørgensen // The Journal of Systems and Software. – Vol. 70, No. 1-2, 2004.
18. Callahan, J. Reducing software product development time [Text] / J. Callahan, B. Moreton // International Journal of Project Management. – Vol. 19, No.1, January 2001.

19. Wiegers, C. Stop Promising Miracles [Text] / C. Wiegers // Software Development Magazine. – Vol. 8, No.2, February 2000. – P. 49-54.
20. Bittner, K. Managing Iterative Software Development Projects [Text] / K. Bittner, I. Spence. – Addison-Wesley Professional, 2006. – 640 p.
21. Кульдин, С. П. Генетический подход к проблеме оценки сроков и трудоемкости разработки программного обеспечения с заданными требованиями к качеству [Текст] / С. П. Кульдин // Прикладная информатика. – №5. – 2010. – С. 30-42.
22. Колдовский, В. Разработка ПО: оценка результата [Текст] / В. Колдовский // Компьютерное обозрение. – №34 (553). – 2006.
23. Putnam, L. H. A general empirical solution to the macrossoftware sizing and estimating problem [Text] / L. H. Putnam // IEEE Transactions on Software Engineering. – Vol. 4, No. 2, July 1978. – Pp. 345-361.
24. Hill, P R. Practical Software Project Estimation: A Toolkit for Estimating Software Development and Duration [Text] / P R. Hill. – McGraw-Hill Professional, 2010.
25. Maheswari, J. U. Project Scheduling using Dependency Structure Matrix [Text] / J. U. Maheswari, K. Varghese. // International Journal of Project Management. – Vol. 23, No. 3, April 2005.
26. Zahedi, F. Managing Information Technology Projects [Text] / F. Zahedi, G. McLeod, D. Smith. – Course Technology, 1996. – 386 p. – ISBN 9780760049440.
27. Kishore, S. Software requirements and estimation [Text] / S. Kishore, R. Naik. – New Delhi : Tata McGraw-Hill Pub. Co., 2001. – 392 p. – ISBN 9780070403123.
28. Lee, D. Norden-Raleigh Analysis: A Useful Tool for EVM in Development Projects [Text] / D. Lee // The Measurable News, March 2002. – Pp. 21–24.

29. Barry, E.J. Software Project Duration and Effort: An Empirical Study [Text] / E. J. Barry, T. Mukhopadhyay, S. A. Slaughter // Information Technology and Management. – Vol. 3, Issue 1, January 2002. – Pp. 113-136.
30. Ayal, M. Forecasting Project Duration as a Function of Scope Changes and Delay [Text] / M. Ayal, S. Globerson // Project Management Institute Proceedings of the Annual Seminars/Symposium, 2002. – Project Management Institute, 2002.
31. Tsui, F. F. Managing Software Projects [Text] / F. F. Tsui. – Jones & Bartlett Learning, 2004. – 337 p. – ISBN 978-0763725464.
32. Ayal, M. The effect of scope changes on project duration extensions [Text] / M. Ayal // Extended Abstract of a PhD Dissertation. – Tel Aviv University, 2005. – 9 p.
33. Sentas, P. Survival analysis for the duration of software projects [Text] / P. Sentas, L. Angelis // In 11th IEEE International Software Metrics Symposium, September 2005. – IEEE, 2005. – Pp. 10-15.
34. Connor, A. M. Using historical data in stochastic estimation of software project duration [Text] / A. M. Connor, S. G. MacDonell // Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications. – Wellington : NACCQ, 2006. – Pp. 53-59.
35. McConnell, S. Software Estimation: Demystifying the Black Art [Text] / S. McConnell. – Microsoft Press, 2006. – 308 p. – ISBN 978-0735605350.
36. Galorath, D. D. Software sizing, estimation, and risk management: when performance is measured performance improves / D. D. Galorath, M. W. Evans. – CRC Press, 2006. – 576 p. – ISBN 978-1420013122.

37. Connor, A.M. Probabilistic estimation of software project duration [Электронный ресурс] / A.M. Connor // Journal of Applied Computing and Information Technology. – NACCQ, 2007. – Vol. 11, Issue. 1, 2007. – Режим доступа: <https://aut.researchgateway.ac.nz/bitstream/handle/10292/7090/27015234.pdf?sequence=15>.
38. Ebert, C. Software Measurement. Establish - Extract - Evaluate - Execute [Text] / C. Ebert, R. Dumke. – Springer, 2007. – 561 p. – ISBN 978-3540716495.
39. Sentas, P. A statistical framework for analyzing the duration of software projects [Text] / P. Sentas, L. Angelis, I. Stamelos // Empirical Software Engineering. – Vol. 13, Issue 2, April 2008. – Pp. 147–184.
40. Ambler, S., Software development project success survey 2008 [Text] / S. Ambler. – Dr. Dobbs Journal, 2009.
41. Buglione, L. Estimation tools and techniques [Text] / L. Buglione, C. Ebert. IEEE Software, Vol. 23, No. 3, 2011. – Pp. 15-18.
42. The IFPUG Guide to IT and software measurement [Text] / IFPUG. – CRC Press, 2012. – 848 p. – ISBN 978-1439869307.
43. Suelmann, H. Putnam's effort-duration trade-off law: is the software estimation problem really solved? [Text] / H. Suelmann // 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, Rotterdam, 2014. – IEEE, 2014. – Pp. 79-84.
44. Ходаков, В. Е. Система информационного обслуживания руководителей предприятий [Текст] / В. Е. Ходаков – Киев : Техніка, 1992. – 200 с.
45. Андон, Ф. И. Логические модели интеллектуальных информационных систем [Текст] / Ф. И. Андон, А. Е. Яшунин, В. А. Резниченко. – НАН Украины. Институт программных систем. – Киев : Наук. думка, 1999. – 396 с.

46. Димо, О. Б. Підвищення ефективності управління проектами розробки програмного забезпечення з відкритим вихідним кодом [Текст] : Дис... канд. техн. наук: 05.13.22 / Олександр Борисович Димо, НУК ім. адм. Макарова – Миколаїв, 2007. – 226 с.
47. Приходько, С. Б. Інтервальне оцінювання математичного сподівання часу затримок виконання програмних проектів на основі перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Вестник ХНТУ. – Херсон : ХНТУ, 2010. – №2 (38). – С. 402-404 – ISBN 5-7763-2514-5.
48. Приходько, С. Б. Інтервальне оцінювання статистичних моментів часу затримок виконання програмних проектів на основі перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Збірник наукових праць НУК. – Миколаїв : НУК, 2010. – № 2 (431). – С. 118-124. – ISSN 2313-0415.
49. Приходько, С. Б. Програмне забезпечення для оцінки ризиків при управлінні програмними проектами [Електронний ресурс] / С. Б. Приходько, А. В. Пухалевич // Електронне Видання «Вісник НУК» №4, 2010р. Режим доступу : <http://ev.nuos.edu.ua/content/10psbypp>.
50. Приходько, С. Б. Вибір нормалізуючого перетворення для оцінки довірчого інтервалу математичного сподівання часу затримок виконання програмних проектів [Текст] / С. Б. Приходько, А. В. Пухалевич, Н. В. Негара // Збірник наукових праць НУК. – Миколаїв : НУК, 2011. – № 3 (438). – С. 124-129. – ISSN 2313-0415.
51. Приходько, С. Б. Порівняння ймовірнісних моделей тривалості робіт в проектах з розробки програмного забезпечення [Текст] / С. Б. Приходько, А. В. Пухалевич // Восточно-Европейский журнал передовых технологий. – Харьков : Технологический центр, 2012. – № 1/13 (55). – С. 39-41. – ISSN 1729-3774.

52. Інформаційна картка НДР «Оцінювання тривалості програмних проектів на основі негаусовських стохастичних моделей», Державний реєстраційний номер 0111U005719 [Рукопис] / С. Б. Приходько, А. В. Пухалевич // Український інститут науково-технічної і економічної інформації, 2012.
53. Інформаційна картка НДР «Побудова нелінійної регресійної моделі тривалості робіт на основі нормалізуючих перетворень для управління часом в програмних проектах», Державний реєстраційний номер 0113U000199 [Рукопис] / С. Б. Приходько, А. В. Пухалевич // Український інститут науково-технічної і економічної інформації, 2015.
54. Приходько, С. Б. Розробка нелінійної регресійної моделі тривалості програмних проектів на основі нормалізуючого перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Радіоелектронні і комп'ютерні системи. – Харків : Харківський авіаційний інститут, 2012. – № 4 (56) – С. 90-93. – ISSN 1814-4225.
55. Приходько, С. Б. Розробка нелінійних регресійних моделей тривалості програмних проектів на основі перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Збірник наукових праць НУК. – Миколаїв : НУК, 2014. – № 2 (2014). – С. 76-80. – ISSN 2313-0415.
56. Приходько, С. Б. Confidence interval estimation of PC software project duration regression based on Johnson transformation [Текст] / С. Б. Приходько, А. В. Пухалевич // Радіоелектронні і комп'ютерні системи. – Харків : Харківський авіаційний інститут, 2014. – № 2 (66). – С. 104-107. – ISSN 1814-4225.
57. Pardoe, Iain. Applied regression modeling [Text] / Iain Pardoe. – Wiley, 2012. – 325 p.
58. Seber, George A.F. Nonlinear Regression [Text] / George A.F. Seber, C.J. Wild. – John Wiley & Sons, Inc., 2003. – 792 p.

59. Приходько, С. Б. Метод побудови нелінійних рівнянь регресії на основі нормалізуючих перетворень [Текст] / С. Б. Приходько // Тези доповідей міждерж. наук.-методич. конф. “Проблеми математичного моделювання” (Дніпродзержинськ, 13-15 червня 2012 р.) – Дніпродзержинськ : ДДТУ, 2012. – С. 31-33.
60. Приходько, С. Б. Доверительный интервал нелинейной регрессии времени восстановления работоспособности устройств терминальной сети [Текст] / С. Б. Приходько, Л. Н. Макарова // Восточно-Европейский журнал передовых технологий. – Харьков : Технологический центр, 2014. – № 3/4 (69). – С. 26-31. – ISSN 1729-3774.
61. Приходько, С. Б. Інформаційна технологія для оцінювання тривалості проектів з розробки програмного забезпечення [Текст] / С. Б. Приходько, А. В. Пухалевич // Проблеми інформаційних технологій. – Херсон, 2016. – № 1 (019). – С. 81-87. – ISSN 1998-7005.
62. Свідоцтво про реєстрацію авторського права на твір №67509. Комп'ютерна програма "Оцінювання тривалості розробки програмного забезпечення" [Текст] / С. Б. Приходько, А. В. Пухалевич – Заявка №68190 від 02.07.2016; видано 31.08.2016. – 1 с.
63. Johnson, R. A. Applied Multivariate Statistical Analysis [Text] / R. A. Johnson, D. W. Wichern. – Pearson Prentice Hall, 2007. – 800 p.
64. Sungbin, C. An exploratory project expert system for eliciting correlation coefficient and sequential updating of duration estimation [Text] / C. Sungbin // Expert Systems with Applications. – Vol. 30, No. 4, 2006. Pp. 553-560
65. Box, G. E. P. An analysis of transformations [Text] / G. E. P. Box, D. R. Cox // Journal of the Royal Statistical Society. Series B (Methodological). – Vol. 26. No. 2, 1964. – Pp. 211-252.
66. Орлов, А. И. Прикладная статистика [Текст] / А. И. Орлов. – Москва : "Экзамен", 2004. – 656 с.

67. Поллард, Дж. Справочник по вычислительным методам статистики [Текст] / Дж. Поллард; пер. с англ. В. С. Занадворова; под ред. и с предисл. Е. М. Четыркина. – Москва : Финансы и статистика, 1982. – 344 с.
68. Кендалл, М. Теория распределений [Текст] / М. Кендалл, А. Стюарт // Пер. с англ., под ред. А. Н. Колмогорова. – Москва : Наука. Гл. ред. физ.-мат. лит., 1966. – 588 с.
69. Приходько, С. Б. Інтервальне оцінювання параметрів стохастичних диференціальних систем на основі модифікації узагальненого методу моментів [Текст] / С. Б. Приходько. // Матеріали XIII Міжнародної конференція з автоматичного управління, м. Вінниця, 25-28 вересня 2006 р. – Вінниця : УНІВЕРСУМ-Вінниця, 2007. – С. 69-75.
70. Приходько, С. Б. Інтервальне оцінювання статистичних моментів негаусівських випадкових величин на основі нормалізуючих перетворень [Текст] / С. Б. Приходько. // Математичне моделювання. – Дніпродзержинськ, 2011. – № 1 (24). – С. 9-13.
71. Приходько, С. Б. Аналитические зависимости для выбора семейств распределения Джонсона [Текст] / С. Б. Приходько, Л. Н. Макарова // Комп'ютерні науки: освіта, наука, практика: матеріали Міжнародної науково-технічної конференції. – Миколаїв: НУК, 2014. – С.152-154.
72. Johnson, N. L. System of Frequency Curves Generated by Methods of Translation [Text] / N. L. Johnson // *Biometrika*, 1949, Vol. 36, No. 1/2 (Jun., 1949). – Pp. 149-176.
73. Хан, Г. Статистические модели в инженерных задачах. Пер. с англ. [Текст] / Г. Хан, С. Шапиро. – М. : Мир, 1969. – 396 с.

74. Бостанджиян, В. А. Распределение Пирсона, Джонсона, Вейбулла и обратное нормальное. Оценивание их параметров [Текст] / В. А. Бостанджиян. – Черноголовка : Редакционно-издательский отдел ИПХФ РАН, 2009. – 240 с.
75. Коваленко, І. І. Сучасні методи статистичного аналізу даних: Навчальний посібник [Текст] / І. І. Коваленко, С. Б. Приходько, Л. О. Латанська. – Миколаїв : НУК, 2011. – 192 с.
76. Кобзарь, А. И. Прикладная математическая статистика. Для инженеров и научных работников [Текст] / А. И. Кобзарь – М. : ФИЗМАТЛИТ, 2006. – 816 с.
77. Приходько, С. Б. Аналитическая зависимость для выбора распределения Джонсона семейства SL [Текст] / С. Б. Приходько, Л. Н. Макарова // Вестник ХНТУ. – Херсон : ХНТУ, 2012. – №2 (45). – С. 101-104.
78. Приходько, С. Б. Інтервальне оцінювання математичного сподівання тривалості програмних проектів [Текст] / С. Б. Приходько, А. В. Пухалевич // Тези доповідей ІХ Міжнародної науково-практичної конференції «Сучасні інформаційні технології в економіці та управлінні підприємствами, програмами та проектами». – Харків, 2011. – С. 214-216.
79. Приходько, С. Б. Інтервальне оцінювання тривалості проектів з розробки програмного забезпечення [Текст] / С. Б. Приходько, А. В. Пухалевич // Управління проектами: стан та перспективи: Матеріали 7-ї міжнародної науково-практичної конференції. – Миколаїв : НУК, 2011. – С. 257-259. – ISBN 978-966-321-190-9.
80. Пухалевич, А. В. Розробка нелінійної регресійної моделі тривалості програмних проектів / А. В. Пухалевич // Макаровські читання. Матеріали Всеукраїнського форуму молодих науковців, 23-24 травня 2013 р. – Миколаїв : НУК, 2013. – С. 122-123.

81. Приходько, С. Б. Побудова регресійної моделі тривалості проектів з розробки програмного забезпечення / С. Б. Приходько, А. В. Пухалевич // Комп'ютерні науки: освіта, наука, практика: Матеріали Міжнародної науково-технічної конференції. – Миколаїв : НУК, 2012. – С. 195-198.
82. Приходько, С. Б. Розробка нелінійних регресійних моделей тривалості програмних проектів на основі нормалізуючого перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Макаровські читання. Матеріали Всеукраїнського форуму молодих науковців, 22-23 травня 2014 р. – Миколаїв : НУК, 2014. – С. 72.
83. Prykhodko, S. B. Developing PC Software Project Duration Model based on Johnson transformation [Text] / S. B. Prykhodko, A. V. Pukhalevich // Proceedings of the 12th International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science TCSET'2014, Lviv-Slavske, Ukraine. – Lviv : Polytechnic National University, 2014. – Pp. 114-116.
84. Pukhalevich, A. V. Confidence interval estimation of regression model of embedded software project duration based on Johnson transformation [Text] / A. V. Pukhalevich // Theoretical and Applied Aspects of Cybernetics. Proceedings of the 4th International Scientific Conference of Students and Young Scientists. – Kyiv : Bukrek, 2014. – Pp 314-320. – ISBN 978-966-399-620-2.
85. Приходько, С. Б. Створення інформаційної технології для оцінювання тривалості розробки програмного забезпечення [Текст] / С. Б. Приходько, А. В. Пухалевич // Прикладна геометрія та інформаційні технології в моделюванні об'єктів, явищ і процесів: Матеріали всеукраїнської науково-практичної конференції (19-21 жовтня 2016 р., м. Миколаїв). – Миколаїв : МНУ імені В.О. Сухомлинського, 2016. – С. 107-108.

86. Conte, S. D. Software Engineering Metrics and Models [Text] / S. D. Conte, H. E. Dunsmore, V. Y. Shen. – Menlo Park, CA : Benjamin/Cummings Publishing Company, Inc., 1986.
87. Yan, X. Linear regression analysis : theory and computing [Text] / X. Yan, X. G. Su. – Singapore : World Scientific Publishing Co. Pte. Ltd., 2009. – 349 p. – ISBN 978-981-283-410-2.
88. Comparative Estimating Tool [Электронный ресурс] // International Software Benchmarking Standards Group. – Режим доступа : <http://isbsg.org/product/comparative-estimating-tool/>.
89. Microsoft Project 2016 Standard [Электронный ресурс] // Microsoft Corporation. – Режим доступа : <https://products.office.com/uk-ua/Project/project-standard-desktop-software>.
90. ДСТУ 2226-93. Автоматизовані системи. Терміни та визначення [Текст]. – Надано чинності 1994-07-01. – Київ : Держстандарт України, 1994. – 92 с.
91. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения [Текст]. – Введен 1992-01-01. – Москва : Стандартинформ, 2009. – 16 с.
92. Калашян, А. Н. Структурные модели бизнеса: DFD-технологии [Текст] / А.Н. Калашян, Г. Н. Калянов. – Москва : Финансы и статистика, 2003. – 256 с.
93. Вендров, А. М. CASE-технологии. Современные методы и средства проектирования информационных систем [Текст] / А. М. Вендров. – Москва : Финансы и статистика, 1998. – 176 с.
94. Калянов, Г. Н. CASE-технологии. Консалтинг при автоматизации бизнес-процессов [Текст] / Г. Н. Калянов. – Москва : Горячая линия – Телеком, 2000. – 320 с.

95. Booch, G. The unified modeling language user guide (2nd Edition) [Text] / G. Booch, J. Rumbaugh, I. Jacobson. – Addison-Wesley Professional, 2005. – 496 p. – ISBN 978-0321267979.
96. Rumbaugh, J. The unified modeling language reference manual (2nd Edition) [Text] / J. Rumbaugh, I. Jacobson, G. Booch. – Addison-Wesley Professional, 2005. – 721 p.
97. Unified modeling language infrastructure specification version 2.4.1 [Електронний ресурс] / Object Management Group, 2011. – Режим доступу: <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>.
98. Unified modeling language superstructure specification version 2.4.1 [Електронний ресурс] / Object Management Group, 2011. – Режим доступу: <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/>.
99. ДСТУ 2481-94. Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення [Текст]. – Надано чинності 1995-01-01. – Київ : Держстандарт України, 1994. – 72 с.
100. A Dictionary of Physics (7 ed.) [Text] / Edited by J. Law, R. Rennie. – Oxford University Press, 2015. – 672 p. – ISBN 978-0198714743.
101. Словарь по кибернетике (2-е издание) [Текст] / Под редакцией В. С. Михалевича. – Киев : Главная редакция УСЭ им. М. П. Бажана, 1989. – 751 с. – ISBN 5-88500-008-5.
102. ГОСТ 7.0-99. Система стандартов по информации, библиотечному и издательскому делу. Информационно-библиотечная деятельность, библиография. Термины и определения [Текст]. – Введен 2000-07-01. – Москва : Издательство стандартов, 1999. – 23 с.
103. Крайников, А. В. Вероятностные методы в вычислительной технике: Учебное пособие для вузов по спец. ЭВМ [Текст] / А. В. Крайников, Б. А. Курдинов, А. Н. Лебедев [и др.]; Под ред. А. Н. Лебедева и Е. А. Чернявского. – Москва : Высшая школа, 1986. – 312 с.

ДОДАТКИ

ДОДАТОК А
АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ
РОБОТИ

ЗАТВЕРДЖУЮ
 Директор ТОВ «Макротел»
 С.В. Бикова
 2016 р.



А К Т

впровадження результатів дисертаційної роботи
 Пухалевича Андрія Володимировича

Ми, що нижче підписалися, директор ТОВ «Макротел» С.В. Бикова та оператор з обробки інформації ТОВ «Макротел» Н.В. Негара, цим Актом засвідчуємо, що результати дисертаційної роботи Пухалевича Андрія Володимировича, а саме:

- негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення;
- нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів;
- рівняння нижньої та верхньої границь довірчих інтервалів нелінійних регресій тривалості проектів з розробки програмного забезпечення;
- рівняння нижньої та верхньої границь довірчих інтервалів прогнозування нелінійних регресій тривалості проектів з розробки програмного забезпечення;
- інформаційна технологія переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

було впроваджено при управлінні проектами з розробки програмного забезпечення, що розроблялися ТОВ «Макротел» в 2016 році.

Впровадження зазначених вище результатів дозволило підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення, що розроблялися ТОВ «Макротел» в 2016 році.

Директор
 ТОВ «Макротел»



С.В. Бикова

Оператор з обробки інформації
 ТОВ «Макротел»



Н.В. Негара

ЗАТВЕРДЖУЮ

Виконавчий директор ЗАТ «Вебкодерс»

 M. Stjernelykke

«07» 11 2016 р.

А К Т

впровадження результатів дисертаційної роботи
Пухалевича Андрія Володимировича

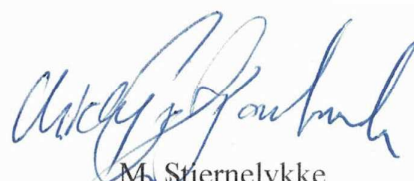
Ми, що нижче підписалися, виконавчий директор ЗАТ «Вебкодерс» М. Stjernelykke та головний проектний менеджер ЗАТ «Вебкодерс» Т. Frost, цим Актом засвідчуємо, що результати дисертаційної роботи Пухалевича Андрія Володимировича, а саме:

- негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення;
- нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів;
- рівняння нижньої та верхньої границь довірчих інтервалів нелінійних регресій тривалості проектів з розробки програмного забезпечення;
- рівняння нижньої та верхньої границь довірчих інтервалів прогнозування нелінійних регресій тривалості проектів з розробки програмного забезпечення;
- інформаційна технологія переробки інформації для оцінювання тривалості проектів з розробки програмного забезпечення

було впроваджено при управлінні проектами з розробки програмного забезпечення, що розроблялися ЗАТ «Вебкодерс» в 2015-2016 роках.

Впровадження зазначених вище результатів дозволило підвищити достовірність оцінювання тривалості проектів з розробки програмного забезпечення, що розроблялися ЗАТ «Вебкодерс» в 2015-2016 роках.

Виконавчий директор
ЗАТ «Вебкодерс»



M. Stjernelykke

Головний проектний менеджер
ЗАТ «Вебкодерс»



T. Frost

ЗАТВЕРДЖУЮ

Ректор

Національного університету кораблебудування
імені адмірала Макарова,

д.т.н., професор



Рижков С.С.

2016 р.

А К Т

впровадження результатів дисертації А.В. Пухалевича
на здобуття наукового ступеня кандидата технічних наук

Ми, що підписалися нижче, директор навчально-наукового інституту комп'ютерних і інженерно-технологічних наук (НН ІКІТН), завідувач кафедри інформаційних управляючих систем та технологій (ІУСТ), д.т.н., професор К.В. Кошкін, професор кафедри програмного забезпечення автоматизованих систем (ПЗАС), д.т.н., професор І.І. Коваленко, склали цей акт у тому, що результати дисертаційної роботи А.В. Пухалевича, а саме:

- негаусівська ймовірнісна модель тривалості проектів з розробки програмного забезпечення;
- нелінійні регресійні моделі тривалості проектів з розробки програмного забезпечення в залежності від трудомісткості цих проектів;
- рівняння нижньої та верхньої границь довірчих інтервалів нелінійної регресії та інтервалів прогнозування нелінійної регресії тривалості проектів з розробки програмного забезпечення;

було впроваджені в навчальний процес кафедри ПЗАС шляхом використання в робочій навчальній програмі з дисципліни "Емпіричні методи програмної інженерії" в 2015/16 навчальному році.

Директор НН ІКІТН,
завідувач кафедри ІУСТ,
д.т.н., професор

К.В. Кошкін

Професор кафедри ПЗАС,
д.т.н., професор

І.І. Коваленко

ДОДАТОК Б
НАСТАНОВА КОРИСТУВАЧА КОМП'ЮТЕРНОЇ ПРОГРАМИ
«ОЦІНЮВАННЯ ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ»

ЗМІСТ

1	Вступ	175
2	Загальні відомості про програму	176
2.1	Позначення і найменування програми	176
2.2	Мови програмування, на яких написана програма.....	176
2.3	Призначення програми	176
2.4	Можливості програми.....	176
2.5	Обмеження області застосування програми.....	176
3	Умови застосування програми	177
3.1	Умови, необхідні для виконання програми	177
3.2	Вимоги до технічних засобів, необхідних для функціонування програми.....	177
3.3	Програмне забезпечення, необхідне для функціонування програми	177
4	Опис програми	177
4.1	Загальний вигляд стартового вікна програми	177
4.2	Результати оцінювання та моделювання тривалості проектів з розробки програмного забезпечення.....	178

1 ВСТУП

Ця настанова призначена для ознайомлення користувача з технічними характеристиками і функціональними можливостями комп'ютерної програми «Оцінювання тривалості розробки програмного забезпечення».

Дана програма призначена для точкового та інтервального оцінювання тривалості проектів з розробки програмного забезпечення для платформ РС (personal computers), MR (mid-range), MF (main frame) в залежності від трудомісткості, а також моделювання тривалості проектів з розробки програмного забезпечення для вказаних платформ. Програма є незалежним програмним продуктом, що має графічний інтерфейс, який дозволяє введення даних, перегляд результатів оцінювання тривалості проектів з розробки програмного забезпечення та збереження отриманих результатів в базу даних.

2 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ПРОГРАМУ

2.1 Позначення і найменування програми

Найменування програми - «Оцінювання тривалості розробки програмного забезпечення». Позначення програми - "DurationEstimate".

2.2 Мови програмування, на яких написана програма

Програма написана на мові програмування Java.

2.3 Призначення програми

Оцінювання тривалості проектів з розробки програмного забезпечення для платформ PC (personal computers), MR (mid-range), MF (main frame) в залежності від трудомісткості та моделювання тривалості проектів з розробки програмного забезпечення для вказаних платформ.

2.4 Можливості програми

Програма виконує наступні функції:

- Оцінювання середнього значення тривалості проекту з розробки програмного забезпечення з використанням нелінійної регресійної моделі в залежності від трудомісткості;
- Оцінювання 95% довірчого інтервалу середнього значення тривалості проекту з розробки програмного забезпечення;
- Оцінювання інтервалу прогнозування тривалості проекту з розробки програмного забезпечення;
- Моделювання тривалості в межах знайденого інтервалу прогнозування;
- Збереження результатів оцінювання і моделювання в базу даних;
- Перегляд результатів раніше проведених оцінювань тривалості проектів з розробки програмного забезпечення.

2.5 Обмеження області застосування програми

- Область застосування програми обмежена сферою розробки програмного забезпечення.

3 УМОВИ ЗАСТОСУВАННЯ ПРОГРАМИ

3.1 Умови, необхідні для виконання програми

Спеціальні умови для виконання програми відсутні.

3.2 Вимоги до технічних засобів, необхідних для функціонування програми

Рекомендовані вимоги:

- ПК з процесором не нижче Core II Duo;
- Обсяг оперативної пам'яті - не менше 1 Гб;
- Необхідний обсяг на жорсткому диску - не менше 50 Мб.

3.3 Програмне забезпечення, необхідне для функціонування програми

Операційна система, в якій встановлена версія Java Runtime Environment (JRE) не нижче версії Java 1.6.

4 ОПИС ПРОГРАМИ

4.1 Загальний вигляд стартового вікна програми

Загальний вигляд стартового вікна комп'ютерної програми DurationEstimate представлений на рис. 1.

Стартове вікно містить 3 поля для ведення вхідних даних:

- Текстове поле «Назва проекту» – дозволяє вказати назву проекту при необхідності подальшого збереження результатів оцінювання і моделювання в базу даних;

- Текстове поле «Трудомісткість» – дозволяє вибрати значення трудомісткості, яке буде основою для оцінювання тривалості проектів з розробки програмного забезпечення;

- Випадаючий список «Платформа» – дозволяє вибрати платформу, для якої розроблюється програмне забезпечення: "Персональні комп'ютери (PC)", "Платформи середнього розміру (MR)", "Мейнфрейми (MF)";

– Текстове поле «Кількість моделювань» – призначене для вказування необхідної кількості змодельованих значень тривалості проектів з розробки програмного забезпечення.

Параметер	Оцінка
1. Вкажіть значення трудомісткості	(ціле число)
2. Виберіть платформу ПЗ	
3. Натисніть кнопку "Розрахувати"	

Рисунок 1 – Загальний вид стартового вікна комп'ютерної програми DurationEstimate

4.2 Результати оцінювання та моделювання тривалості проектів з розробки програмного забезпечення

Після введення вхідних даних у вказані поля стартового вікна користувачу потрібно натиснути кнопку «Розрахувати». Результати оцінювання тривалості проектів з розробки програмного забезпечення (Оцінка тривалості, Довірчий інтервал нелінійної регресії, Інтервал прогнозування) та необхідна кількість змодельованих значень тривалості проектів з розробки програмного забезпечення будуть показані в таблиці в нижній частині вікна програми (рис. 2).

Оцінювання тривалості розробки програмного забезпечення

Інші проекти Вихід

Назва проекту: Проект 1

Трудомісткість: 1000 *людино-годин*

Платформа: Мейнфрейми (MF)

Кількість моделювань: 5

Зберегти оцінку Оцінити тривалість

Параметер	Оцінка
Назва	Проект 1
Платформа	Мейнфрейми (MF)
Трудомісткість	1000 людино-годин
Оцінювання тривалості:	
Оцінка тривалості	6.5 місяців
95% Довірчий інтервал	[5.9, 7.1] місяців
95% Інтервал передбачення	[2.1, 19.2] місяців
Моделювання тривалості:	
1.	6.1 місяців
2.	3.6 місяців
3.	11.4 місяців
4.	18.9 місяців
5.	11.3 місяців

Рисунок 2 – Результати оцінювання та змодельовані значення тривалості проектів з розробки програмного забезпечення

При необхідності результати оцінювання і моделювання можна зберегти в базу даних, натиснувши на кнопку «Зберегти оцінку». Для перегляду раніше збережених результатів оцінювання і моделювання (рис. 3) потрібно вибрати пункт меню «Інші проекти» вверху програми.

Оцінки проектів з розробки програмного забезпечення

Назва	Платформа	Трудомісткість	Тривалість	Довірчий інтервал	Інтервал передбачення	Моделювання тривалості	Дата
Проект 1	MF	1000	6.5	[5.9, 7.1]	[2.1, 2.1]	6.1, 3.6, 11.4, 18.9, 11.3	2016-03-21
Проект 2	PC	1500	8.3	[7.1, 9.7]	[2.9, 2.9]	21.1, 7.0, 18.1, 13.6, 8.7	2016-05-30

Рисунок 3 – Перегляд збережених результатів оцінювання і моделювання тривалості проектів з розробки програмного забезпечення