УДК 004.67+005.2 DOI: 10.30837/0135-1710.2025.184.005

Н.В. ВАСИЛЬЦОВА, А.В. ПОПОВА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КІЛЬКІСНОГО ОЦІНЮВАННЯ ЗМІН У ДОВГОСТРОКОВОМУ ІТ-ПРОЄКТІ

Розглянуто основні існуючі методи управління змінами в проєктах. Встановлено, що жоден з цих методів не дозволяє кількісно оцінювати зміни, які виникають під час виконання проєкту. Як рішення задачі підвищення якості процесу управління змінами та точності оцінювання змін часу у довгострокових ІТ-проєктах запропоновано розробити інформаційну технологію кількісного оцінювання змін на основі методу, заснованого на моделі Бекхарда і Гарріса. Виконано опис архітектури розробленої інформаційної технології, її технологічний стек та особливості реалізації її математичного та програмного забезпечень.

1. Вступ

Останні десятиріччя характеризуються широким розповсюдженням довгострокових проєктів, термін виконання яких складає три і більше років. Такі проєкти є характерними й для різних галузей ІТ-сфери: розробки та впровадження інформаційних технологій (ІТ), інформаційних систем (ІС), ІТ-інфраструктури, наукових досліджень в даній галузі тощо. Довгострокові проєкти часто розбиваються на декілька етапів або фаз, кожна з яких має свої цілі, завдання та результати, що дозволяє краще контролювати прогрес і вчасно вносити необхідні корективи [1], [2].

Аналіз характеристик сучасних довгострокових ІТ-проєктів дозволяє виділити основні їхні особливості [2]:

– складність опису, спричинена великою кількістю функцій, процесів, елементів, даних та зв’язків між ними;

– наявність сукупності тісно взаємозв’язаних підсистем, кожна з яких має свої локальні цілі та задачі;

– необхідність інтеграції існуючих і повторно розроблюваних застосувань;

– різнорідність окремих груп розробників за рівнем класифікації та сформованих традицій використання інструментальних засобів;

– постійний потік змін в сучасних ІС.

Значна кількість змін при реалізації таких проєктів відбувається на операційному рівні задач, що робить необхідним якісне управління змінами на цьому рівні. Наприклад, низька якість оцінювання змін часу виконання окремих операцій проєкту може призвести до перевищення термінів виконання проєкту, перенапруження ресурсів та непередбачуваних затримок у запуску продукту на ринок. Навпаки, систематичне оцінювання змін часу виконання дозволяє при управлінні проєктами приймати обґрунтовані рішення щодо ресурсного планування, розподілу завдань та пріоритетів у випадку змін у вимогах чи умовах проєкту. Такий підхід сприяє зменшенню ризиків затримок у виконанні проєкту та забезпечує його успішне завершення в рамках встановлених термінів. Тому проведення досліджень, спрямованих на підвищення якості оцінювання змін часу виконання окремих операцій довгострокових ІТ-проєктів, є актуальним.

2. Аналіз літературних даних і постановка проблеми дослідження

Найпоширенішими методами управління змінами у проєктах є:

– метод, заснований на моделях ADKAR та ADKAR PROSCI [3], [4];

– метод прискорення впровадження (Accelerating Implementation Methodology, AIM) [5], [6];

– метод, заснований на моделі Бекхарда і Гарріса [7];

– метод, заснований на моделі Бріджеса [8];
– метод, заснований на моделі Джона Коттера [9];
– метод, заснований на моделі Кюблер-Росс [10];
– метод, заснований на моделі Курта Левіна [11].

Метод, заснований на моделі ADKAR, був створений Джефрі Хаяттом наприкінці 1990-х років. ADKAR – це скорочення, яке представляє п'ять ключових етапів, необхідних для успішного здійснення змін: «Усвідомлення» (Awareness); «Бажання» (Desire); «Знання» (Knowledge); «Здатність» (Ability); «Підкріплення» (Reinforcement) [4]. Цей метод пропонує глибокий підхід до управління змінами, зосереджений на персоналі. Передбачається, що в процесі впровадження змін персоналу необхідно чітко роз'яснити причини змін та їхню важливість, що сприяє залученню персоналу до процесу. Далі важливо навчити кожного співробітника методам впровадження змін, що дозволяє показати рівень їхніх професійних знань і навичок у процесі внесення змін. Завершальний етап передбачає закріплення змін, забезпечуючи сталість інновацій у діяльності організації [3].

Метод ADKAR PROSCI є частиною ширшого підходу до управління змінами, розробленого компанією PROSCI. Він базується на методі з використанням моделі ADKAR, але включає додаткові інструменти, методи та практики, спрямовані на ефективне впровадження змін в організації [4].

AIM є потужним та дисциплінованим методом управління організаційними змінами, включаючи трансформаційні зміни до повного повернення інвестицій. Це інтегрована система операціоналізованих принципів, стратегій, тактик, аналітики вимірювання та інструментів, підтримана сертифікаційними та навчальними програмами. AIM можна застосовувати до будь-якого виду ініціативи або проєкту. Але більшість організацій спрямовують основні ресурси та енергію на технічні та бізнес-процесні компоненти [5], [6].

Однією з переваг цього методу є можливість систематичного аналізу та передбачення можливих труднощів у процесі змін, що дозволяє організаціям ефективніше підготуватися та відповісти на них. Однак цей підхід може вимагати значних ресурсів та часу на його виконання, а також може стати складним у випадку несприятливих обставин чи непередбачуваних подій у процесі впровадження змін [6].

Метод, заснований на моделі Бекхарда і Гарріса, передбачає п'ять етапів управління змінами, спрямованих на визначення потреби в змінах, розробку стратегії їх реалізації, формування плану дій та визначення відповідальних виконавців. Основною перевагою методу, заснованому на моделі Бекхарда і Гарріса, є систематичний підхід до процесу змін, що дозволяє структурувати та керувати ними ефективно. Компанія Prахіе створила програмні застосунки для управління змінами з використанням методу, заснованому на моделі Бекхарда і Гарріса, та надає можливість проходити тренінги [7], [8].

Метод, заснований на моделі переходу Бріджеса, надає можливість організаціям та особам краще розуміти людську та особисту сторони змін та ефективно керувати ними. Метод базується на виконанні трьох етапів, які проходить кожна людина під час змін: завершення того, що є зараз; нейтральна зона; новий початок. Основною перевагою цього методу є його спрямованість на внутрішній перехід, що може забезпечити глибшу та стійкішу зміну організаційного середовища. Однак він може виявитися менш ефективним у випадку, коли потрібно швидко реагувати на зовнішні та непередбачувані зміни, оскільки фокусується на внутрішньому аспекті переходу [7].

Метод, заснований на моделі Джона Коттера, покликаний підвищити залученість персоналу до управління змінами та забезпечити їхню прийнятність для всіх працівників. Метод включає вісім таких етапів [9]: «Створення емоційної необхідності для зміни»; «Створення коаліції»; «Складання бачення»; «Розповсюдження бачення»; «Впровадження

дій»; «Формування короткострокових досягнень»; «Закріплення досягнень»; «Впровадження змін у культуру». Пропуск одного з цих етапів може призвести до проблем у внесенні змін та ускладнити процес змін.

Основна перевага методу полягає у його систематичному підході до управління змінами та акценті на залученні персоналу до процесу.

До недоліків методу, заснованому на моделі Коттера, можна віднести такі:

– реалізація всіх восьми етапів методу може вимагати значних зусиль та часу, особливо у великих організаціях або при великому масштабі змін;

– метод фокусується переважно на організаційних аспектах змін, залишаючи осторонь індивідуальні емоції та потреби працівників;

– метод не завжди враховує зовнішні впливи, такі як економічні чи політичні фактори, які можуть впливати на успішність змін.

Метод, заснований на моделі Кюблер-Росс, описує етапи зміни поведінки персоналу, включаючи: опір персоналу до змін, неусвідомлення наслідків змін, адаптацію персоналу до нових умов праці, позитивне ставлення працівників до змін та їх прийняття. Цей метод важливий для розуміння та прогнозування реакцій персоналу на зміни в організації. Однак, варто враховувати, що реакція на зміни може бути індивідуальною та не завжди відповідає цій послідовності етапів. Даний метод базується на кривій змін Кюблер-Росс, яка відображає емоційний стан співробітників (див. рис. 1). Пояснення щодо корисності цього методу розповсюджується в корпоративному світі для кращого розуміння емоційних турбот, з якими стикається співробітник через будь-яку ініціативу змін на робочому місці, таку як впровадження нового програмного забезпечення чи вдосконалення бізнес-процесів. З практичної сторони метод може бути використаний для виявлення будь-яких перешкод у ранніх етапах змінних проєктів та розробки відповідних стратегій.

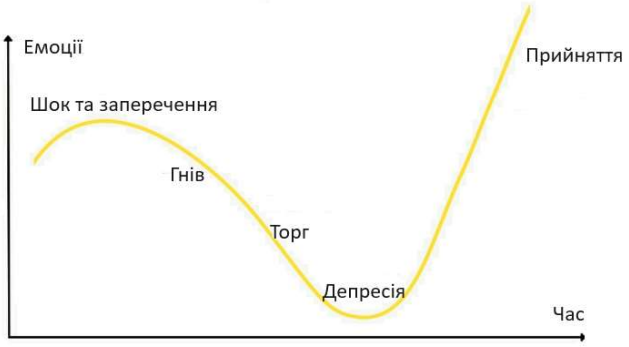


Рис.1. Крива змін Кюблер-Росс

До основних переваг методу можна віднести те, що він є простим, але ефективним для управління організаційними змінами. Недоліком методу є те, що оскільки всі співробітники реагують з різною швидкістю, вони перебувають на різних етапах, позначених на кривій змін. Це часто призводить до проблем при плануванні змін [10].

Метод, заснований на моделі Курта Левіна, розроблений в середині XX століття, залишається одним з найпопулярніших на сучасному етапі розвитку управління змінами. Цей метод відзначається своєю системністю та практичністю, що робить його ефективним для впровадження змін у різних типах організацій. Однак, він може вимагати значних ресурсів та часу на виконання, а також виявитися менш ефективним у випадку складних організаційних структур або невибіркового застосування методів управління змінами [11].

Для порівняння розглянутих методів управління змінами у проєктах у ході дослідження запропоновано основні критерії (показники). Результати порівняння методів

управління змінами за критеріями наведено в табл. 1.

Таблиця 1

Порівняння існуючих методів управління змінами у проєктах

Показник	Метод № 1	Метод № 2	Метод № 3	Метод № 4	Метод № 5	Метод № 6	Метод № 7
Простота використання	–	–	+	–	–	+	+
Складність впровадження	–	–	+	–	+	+	+
Час, необхідний на використання	+	+	–	+	+	+	–
Наявність кількісної оцінки змін	–	–	–	–	–	–	–
Гнучкість методу	–	–	+	+	–	+	–
Використання при швидких змінах	–	+	+	–	–	–	–
Можливість проходження сертифікації	+	+	–	+	+	–	–
Можливості проходження тренінгів	+	+	+	+	+	+	+
Наявність інформативного сайту	+	+	+	–	+	–	–
Наявність безкоштовних тренінгів	–	–	+	–	–	+	+
Можливість використання в складних проєктах	+	+	+	–	+	–	–
Вартість тренінгів, долари США	2000	1790	–	–	6000 (850)	–	–

У таблиці 1 прийнято такі позначення:

– метод № 1 – метод, заснований на моделях ADKAR та ADKAR PROSCI;

– метод № 2 – метод AIM;

– метод № 3 – метод, заснований на моделі Бекхарда і Гарріса;

– метод № 4 – метод, заснований на моделі Бріджеса;

– метод № 5 – метод, заснований на моделі Джона Коттера;

– метод № 6 – метод, заснований на моделі Кюблер-Росс;

– метод № 7 – метод, заснований на моделі Курта Левіна.

В процесі дослідження перелічених методів управління змінами було виявлено, що в них відсутні підходи для кількісного оцінювання змін на рівні задач. Отже, реалізація підходу з можливістю кількісного оцінювання змін є актуальною з теоретичної та прикладної точок зору.

3. Мета і задачі дослідження

Метою даного дослідження є розробка способу підвищення точності оцінювання змін часу виконання задач довгострокових ІТ-проєктів. Застосування цього способу дозволить зменшити витрати на реалізацію довгострокових ІТ-проєктів за рахунок скорочення незапланованих витрат часу на виконання окремих задач цих проєктів.

Для досягнення цієї мети у дослідженні вирішуються такі задачі:

– запропонувати підхід, який дозволить автоматизувати вирішення задачі кількісного оцінювання змін в межах існуючого методу управління змінами при реалізації ІТ-проєктів;

– розробити ІТ кількісного оцінювання змін на основі існуючого методу управління змінами;

– розробити елементи реалізації ІТ кількісного оцінювання змін.

4. Матеріали і методи дослідження

Об'єктом дослідження є процес оцінювання змін часу виконання задач при реалізації довгострокових ІТ-проєктів. Основною гіпотезою дослідження є гіпотеза про можливість підвищення якості оцінювання змін часу виконання окремих задач довгострокових ІТ-проєктів шляхом розробки та впровадження ІТ, яка дозволить автоматизувати використання одного з існуючих методів оцінювання змін.

В основу розроблюваної ІТ запропоновано покласти метод, заснований на моделі Бекхарда і Гарріса [7]. Цей метод складається з таких етапів:

– «Внутрішній організаційний аналіз»;

– «Визначення потреби в змінах»;

– «Аналіз відмінностей між поточним станом та бажаним»;

– «Планування дій»;

– «Керування переходом».

Етап «Внутрішній організаційний аналіз». Метою етапу є визначення загального ставлення до змін в організації. На цьому етапі необхідно ідентифікувати співробітників, які можуть чинити опір змінам. Зокрема, необхідно визначити будь-які зовнішні чинники, які можуть перешкоджати процесу змін.

Етап «Визначення потреби в змінах». Необхідний для того, щоб створити підставу для впровадження змін. Ключові учасники змін повинні погодитися з тим, що зміни є необхідними для успіху організації. Це погодження передбачає, що учасники змін можуть чітко сформулювати, куди вони хочуть привести організацію і чому впровадження змін допоможе наблизити організацію до бажаного стану. Необхідно також мати уявлення про наслідки, пов'язані з відмовою від впровадження змін [7].

Етап «Аналіз відмінностей між поточним станом та бажаним». Перед проведенням впровадження змін спочатку необхідно визначити, які відхилення існують між поточним станом організації і тим, яким вона повинна бути. Визначення цих відхилень важливе для того, щоб чітко сформулювати розуміння майбутнього організації.

Етап «Планування дій». Формується та передається до виконання план змін. Необхідно визначити ключових учасників процесу змін та обов'язки кожного, хто вносить зміни.

Етап «Керування переходом». Після того, як зміна буде здійснена, співробітники, що здійснюють зміни, відповідають за безперервний моніторинг прогресу змін і внесення коригувань.

Основними недоліками цього методу є:

– відсутність можливості кількісного оцінювання змін, які виникають під час виконання ІТ-проєкту;

– суб'єктивність прийняття рішень з управління змінами, які залежать від індивідуальної компетентності осіб, що приймають рішення під час виконання етапів

розглядуваного методу.

5. Результати дослідження

5.1. Визначення підходу до автоматизованого вирішення задачі кількісного оцінювання змін

Для розробки ІТ, яка дозволить автоматизувати вирішення задачі кількісного оцінювання змін у довгострокових ІТ-проєктах, пропонується застосувати дескрипторний підхід. Цей підхід базується на таких поняттях: SCRUM, спринт, задача, дескриптор, показник змін.

SCRUM – методологія управління проєктами, яка зазвичай використовується в розробці програмного забезпечення. Вона базується на ітераційному та інкрементальному підходах до розробки продукту, де команда працює у коротких циклах (спринтах) [12].

Спринт – це короткий період часу (зазвичай, від двох до чотирьох тижнів), в рамках якого проєктна команда працює над конкретним набором задач. Така практика реалізується при використанні Agile-методології розробки програмного забезпечення для підвищення ефективності та прозорості процесу розробки [12].

Задача – окреме завдання з унікальним ідентифікаційним номером, яке виконується в рамках спринту певною проєктною командою для досягнення поставленої мети або результату. Задача найчастіше оцінюється за складністю та часом виконання. Вона має такі характеристики: опис, дату створення, пріоритет, оцінку часу виконання, відповідального виконавця, статус виконання, критерії прийняття, дедлайн, коментарі та додаткові матеріали.

Дескриптор – кількісна характеристика задачі, яка надає описову інформацію про задачу в числовому форматі. Дескриптор може описувати певні аспекти задач, такі як: інформацію про використання певної технології, інформацію про команду, що буде виконувати задачу тощо.

Показник змін – числовий показник, з використанням якого можна зробити оцінку змін при їх внесенні. Цей показник виступає числовим показником оцінювання змін. Показником змін може виступати час, необхідний для виконання задачі, або різниця запланованого часу та часу, витраченого на виконання задачі.

Дескрипторний підхід, який пропонується для оцінювання змін на рівні задач ІТ-проєкту, включає виконання таких процесів:

– збір та зберігання в базі даних інформації, яка буде описувати характеристики виконуваних задач у вигляді дескрипторів;

– побудова статистичних моделей для пошуку залежності показника змін від дескриптора з метою кількісного оцінювання змін.

Умовою для використання дескрипторного підходу є наявність достатньої кількості інформації про виконувані задачі. Для довгострокових ІТ-проєктів, відмінною рисою яких є характерне накопичування інформації щодо виконаних задач в значній кількості, що дає змогу цю інформацію аналізувати з використанням статистичних методів, ця умова виконується.

При використанні дескрипторного підходу в рамках реалізації довгострокового ІТ-проєкту запропоновано таку класифікацію дескрипторів:

– технічні дескриптори;

– дескриптори особливостей задачі;

– дескриптори опису команди;

– дескриптори процесу тестування;

– дескриптори доступності веб-сторінки;

– інші.

Кількість різновидів дескрипторів, за якими проводиться класифікація, може коливатись в залежності від вимог ІТ-проєкту.

Дескриптор може мати бінарне значення: «Так», або «1», або «Істина» чи «Ні», або «0», або «Хибність». Наприклад, дескриптор зміни бібліотеки для тестування модулів для певного програмного компоненту може мати значення «1» (якщо в завданні необхідно змінити бібліотеку для тестування компоненту) або «0» (якщо ні).

Прикладами дескриптору з простим числовим значенням можуть бути дескриптор кількості рядків коду, для якого необхідно реалізувати функціональне тестування, або дескриптор кількості помилок в коді, які необхідно виправити під час виконання задачі.

При використанні дескрипторного підходу рекомендується надавати можливість внесення дескрипторів до бази даних всім членам проєктної команди.

Пропонується процес оновлення бази даних дескрипторів проводити в рамках кожного спринту та інформацію про наявність цього процесу додати до вимог готовності задач, які існують у проєкті (DoD, definition of done) [13].

При наявності бази даних дескрипторів та показників змін можна будувати статистичні моделі залежності показників змін від дескрипторів.

Пропонується будувати саме одномірну модель, виходячи з таких причин:

– простота розрахунків;

– необхідність меншої кількості даних для розробки моделі;

– відсутність проблеми, пов’язаної з мультиколінеарністю дескрипторів, які пропонуються [14].

Одномірна модель може бути достатньою для виявлення основних тенденцій і залежностей в даних, особливо на початкових етапах дослідження.

5.2. Результати розробки інформаційної технології кількісного оцінювання змін

ІТ кількісного оцінювання змін було запропоновано розробити для автоматизованого виконання методу, заснованого на моделі Бекхарда і Гарріса, із застосуванням дескрипторного підходу. Тому було прийнято рішення про застосування цієї ІТ для автоматизації тільки тих етапів методу, які безпосередньо пов’язані із збиранням, обробкою та зберіганням даних та інформації щодо змін, які виникають під час виконання довгострокового ІТ-проєкту. До цих етапів належать:

– етап «Внутрішній організаційний аналіз»;

– етап «Аналіз відмінностей між поточним станом та бажаним»;

– етап «Керування переходом».

Виходячи з цього рішення, розроблювану ІТ було запропоновано представити як послідовність таких етапів і кроків.

Етап 1. Опитування команди виконавців щодо поточного сприйняття процесу управління змінами.

Крок 1.1. Формування анкети та проведення опитування усіх співробітників, які утворюють команди виконавців ІТ-проєкту, щодо поточного ставлення до процесу управління змінами.

Крок 1.2. Обробка результатів анкетування і формування поточних оцінок рівня задоволеності співробітників процесом управління змінами.

Етап 2. Формування та зберігання дескрипторів окремих робіт ІТ-проєкту.

Крок 2.1. Визначення множини дескрипторів окремих робіт ІТ-проєкту.

Крок 2.2. Формування множин значень визначених дескрипторів.

Крок 2.3. Зберігання сформованих множин значень визначених дескрипторів.

Етап 3. Статистичний аналіз дескрипторів ІТ-проєкту.

Крок 3.1. Визначення показників змін ІТ-проєкту.

Крок 3.2. Вибір визначених дескрипторів для створення моделі змін.

Крок 3.3. Аналіз множин значень відібраних дескрипторів на наявність викидів.

Крок 3.4. Якщо результати аналізу, отримані в результаті виконання Кроку 3.3, свідчать про відсутність викидів, то побудова моделі змін із застосуванням методу найменших квадратів (ordinary least squares, OLS). В іншому випадку – побудова моделі змін із застосуванням методу регресії Хубера.

Крок 3.5. Розрахунок показників працездатності побудованої моделі змін.

Етап 4. Опитування команди виконавців щодо підсумкового сприйняття процесу управління змінами.

Крок 4.1. Формування анкети та проведення опитування усіх співробітників, які утворюють команди виконавців ІТ-проєкту щодо підсумкового ставлення до процесу управління змінами.

Крок 4.2. Обробка результатів анкетування і формування підсумкових оцінок рівня задоволеності співробітників процесом управління змінами.

Опис архітектури розроблюваної ІТ кількісного оцінювання змін у вигляді діаграми потоків даних наведено на рис. 2. Під час створення цієї діаграми було застосовано нотацію Йордона-ДеМарко. На рис. 2 не вказано назви потоків, які безпосередньо пов'язані із сховищами даних, тому що ці назви співпадають із назвами самих сховищ даних.

Для подальшої реалізації розроблюваної ІТ було визначено особливості її технологічного стеку. Під технологічним стеком тут і далі будемо розуміти набір технологій, які використовуються разом для розробки та підтримки програмного забезпечення [15].

Технологічний стек ІТ кількісного оцінювання змін має такі складові:

а) для розробки елементів «Опитування команди виконавців щодо поточного сприйняття процесу управління змінами» та «Опитування команди виконавців щодо підсумкового сприйняття процесу управління змінами» запропоновано використати сервіс Google Forms, який входить до безкоштовного пакету редакторів Google Docs компанії Google;

б) для розробки елементу «Формування та зберігання дескрипторів окремих робіт ІТ-проєкту» запропоновано використати існуючі системи управління ІТ-проєктами (наприклад, Jira [16]) та інструментальні засоби для зберігання описів дескрипторів та їхніх числових значень (наприклад, продукт Microsoft Excel, або його аналоги, або системи управління базами даних Firebase, MongoDB, PostgressSQL тощо);

в) для розробки елементу «Статистичний аналіз дескрипторів ІТ-проєкту» запропоновано розробити спеціалізований сервіс із застосуванням мови програмування Python.

5.3. Особливості реалізації інформаційної технології кількісного оцінювання змін

5.3.1. Особливості реалізації математичного забезпечення інформаційної технології

При реалізації дескрипторного підходу було запропоновано досліджувати залежність між показниками змін (шуканими показниками) та значеннями дескрипторів.

Дану залежність запропоновано представити формулою:

$$y_i = \beta_0 + \beta_1 * x_i , \quad (1)$$

де y_i – залежна змінна (показник змін); x_i – незалежна змінна (дескриптор); β_0, β_1 – параметри регресії.

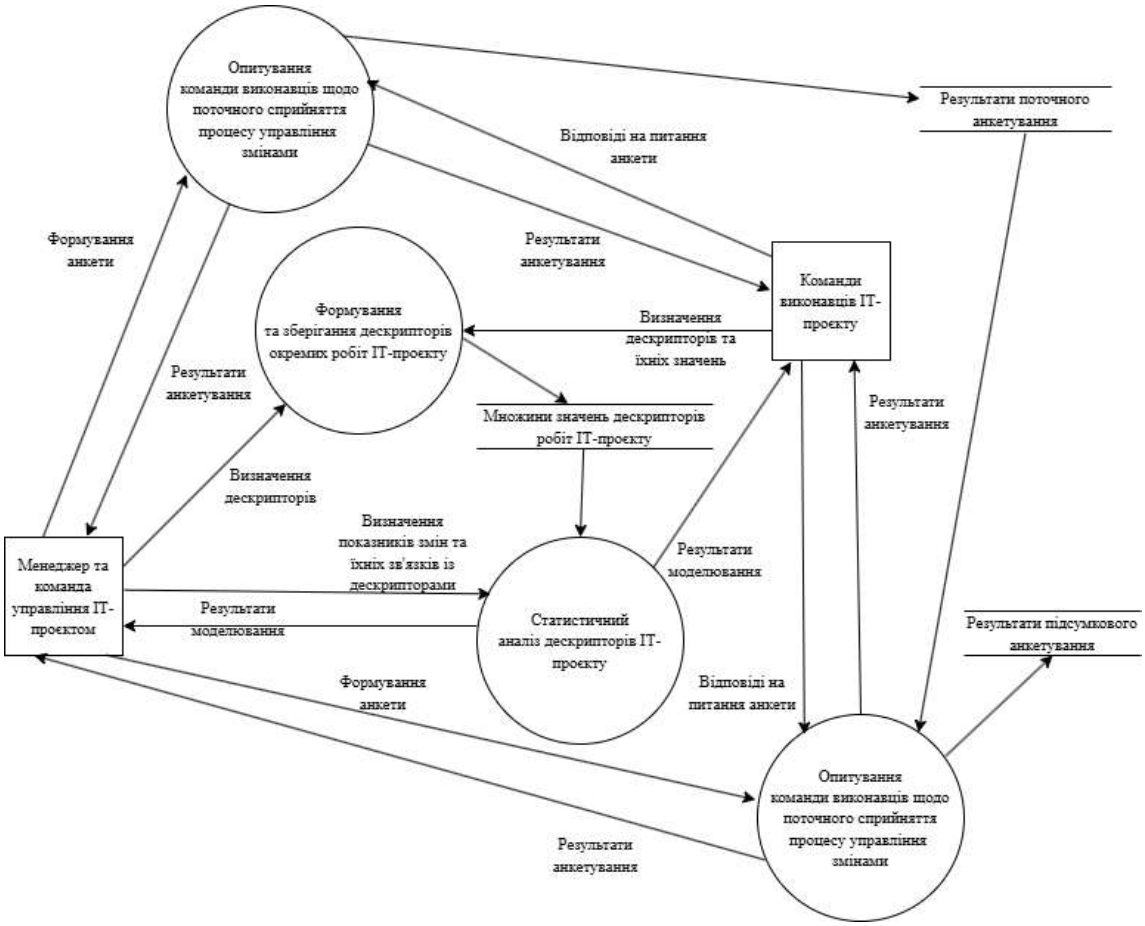


Рис. 2. Опис архітектури інформаційної технології кількісного оцінювання змін

Існує багато методів для побудови регресійної моделі (1). Проте, найчастіше використовується метод OLS. Перед використанням методу OLS необхідно перевірити виконання передумов використання регресійного аналізу. В результаті такого аналізу можуть бути виявлені викиди, які треба буде опрацювати. У випадку наявності викидів в даних, які збираються при виконанні задач проєкту, в методі, який розробляється, пропонується використовувати регресію Хубера, тому що вона має властивості, пов'язані з робастністю [17].

Після побудови статистичних моделей пропонується виконувати перевірку їхньої працездатності з використанням таких показників [17]:

– коефіцієнт детермінації (R^2);

– коефіцієнт прогнозування (Q^2);

– коефіцієнт прогнозування, розрахований за процедурою Leave-one-out cross-validation (Q^2_{LOOCV});

– стандартне відхилення (σ);

– оцінка значущості статистичної моделі за F-критерієм.

Якщо аналіз статистичних моделей надає незадовільні показники, відбувається вибір іншого дескриптору для аналізу та перерахунок статистичних моделей. У випадку задовільних показників аналізу статистичних моделей відбувається оцінювання змін з

використанням побудованої моделі.

З метою оцінювання успішності процесу управління змінами було запропоновано використовувати такі показники:

– доля (у відсотках) змін часу Ch_{Cp}, скасованих у зв'язку з неможливістю їх реалізації, за виключенням змін, що стали непотрібними за певний період;

– доля (у відсотках) змін часу, що були прийняті замовником та затверджені як успішно реалізовані за певний період часу, Ch_{Sp};

– різниця між часом, який був витрачений на виконання робіт, і оціненим часом за певний період часу, δCh_T;

– доля (у відсотках) змін, які були завершені вчасно за певний період часу, Ch_{iTp}.

Долю (у відсотках) змін, скасованих за певний період часу у зв'язку з неможливістю їх реалізації, Ch_{Cp}, за виключенням змін, що стали неактуальними, можна розрахувати за формулою:

$$Ch_{Cp} = \frac{Ch_C}{Ch} * 100\%, \tag{2}$$

де Ch_C – кількість скасованих змін за певний період часу у зв'язку з неможливістю їх реалізації, за виключенням змін, що стали неактуальними; Ch – загальна кількість змін за певний період часу.

Великі значення Ch_{Cp} свідчать про погано сплановані зміни.

Долю (у відсотках) змін, що були прийняті замовником та затверджені як успішно реалізовані, за певний період часу Ch_{Sp} можна розрахувати за формулою:

$$Ch_{Sp} = \frac{Ch_S}{Ch} * 100\%, \tag{3}$$

де Ch_S – кількість змін, що були прийняті замовником та затверджені як успішно реалізовані за певний період часу.

Велике значення Ch_{Sp} свідчить про кращий процес управління змінами.

Різницю між часом, витраченим на виконання задач довгострокового IT-проєкту, і оціненим часом за певний період часу δCh_T можна розрахувати за формулою:

$$\delta Ch_T = \frac{\sum_{i=0} (t_i^{(p)} - t_i^{(a)})}{t * i}, \tag{4}$$

де $t_i^{(p)}$ – час, запланований на виконання зміни, днів; $t_i^{(a)}$ – час, витрачений на виконання зміни, днів; t – період часу для оцінки, днів; i – кількість змін за період часу t.

Показник δCh_T вказує на те, чи виконуються зміни вчасно і відповідно до плану змін. Що менше показник, то краще організоване управління змінами.

Долю (у відсотках) змін, які були завершені вчасно за певний період часу Ch_{iTp}, можна розрахувати за формулою:

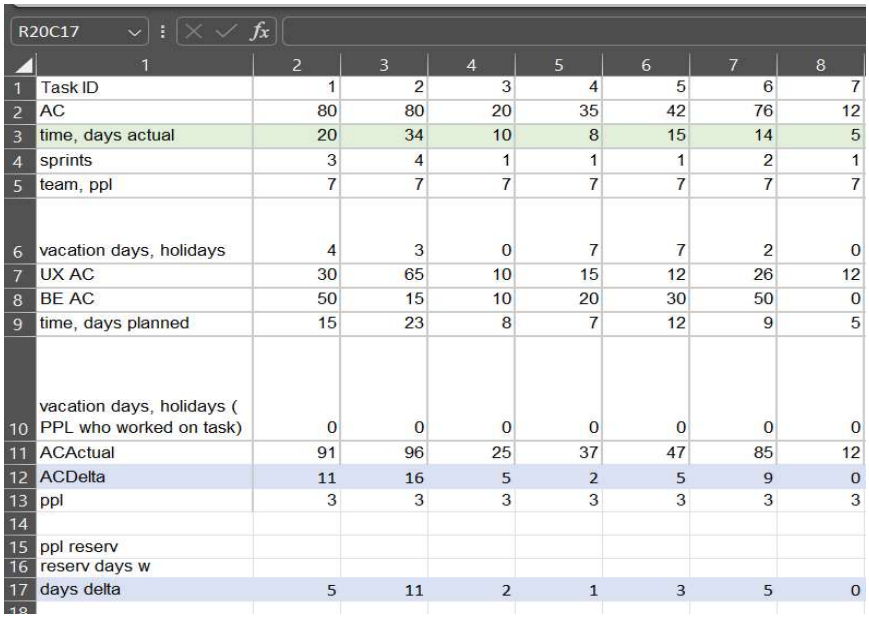
$$Ch_{iTp} = \frac{Ch_{iT}}{Ch} * 100\%, \tag{5}$$

де Ch_{iT} – кількість змін, які були завершені вчасно за певний період часу.

Велике значення Ch_{iTp} свідчить про кращий процес управління змінами та слідування запланованому графіку.

5.3.2. Особливості реалізації програмного забезпечення інформаційної технології

Для реалізації сховищ даних, де планується зберігати описи дескрипторів та їхніх числових значень, було використано продукт Microsoft Excel. Приклад описів визначених дескрипторів та значень цих дескрипторів для окремих задач довгострокового IT-проєкту «Web Constructor» наведено на рис. 3.



	1	2	3	4	5	6	7	8
1	Task ID	1	2	3	4	5	6	7
2	AC	80	80	20	35	42	76	12
3	time, days actual	20	34	10	8	15	14	5
4	sprints	3	4	1	1	1	2	1
5	team, ppl	7	7	7	7	7	7	7
6	vacation days, holidays	4	3	0	7	7	2	0
7	UX AC	30	65	10	15	12	26	12
8	BE AC	50	15	10	20	30	50	0
9	time, days planned	15	23	8	7	12	9	5
10	vacation days, holidays (PPL who worked on task)	0	0	0	0	0	0	0
11	ACActual	91	96	25	37	47	85	12
12	ACDelta	11	16	5	2	5	9	0
13	ppl	3	3	3	3	3	3	3
14								
15	ppl reserv							
16	reserv days w							
17	days delta	5	11	2	1	3	5	0

Рис. 3. Приклад зберігання описів визначених дескрипторів та їхніх значень для окремих задач IT-проєкту

Фрагмент програмного коду для обчислення значень функції щільності Гаусса наведено на рис. 4. Фрагмент програмного коду для побудови графіка обчисленої функції щільності Гаусса та перевірки отриманих результатів методом трьох сигм наведено на рис. 5. Приклад результату роботи цих фрагментів наведено на рис. 6.

```
// Функція для обчислення значень гауссівської функції щільності (PDF)
function gaussianPDF(x, mean, variance) {
  const stdDev = Math.sqrt(variance);
  return (1 / (stdDev * Math.sqrt(2 * Math.PI))) * Math.exp(-((x - mean) ** 2) / (2 * variance));
}
```

Рис. 4. Фрагмент програмного коду для обчислення значень функції щільності Гаусса

Фрагмент програмного коду створення моделі за методом OLS наведено на рис. 7. Фрагмент програмного коду створення моделі за методом регресії Хубера наведено на рис. 8.

Приклад візуального представлення моделі, побудованої з використанням наведеного на рис. 7 програмного коду, показано на рис. 9. Приклад візуального представлення моделі, побудованої з використанням наведеного на рис. 8 програмного коду, показано на рис. 10. Фрагмент програмного коду розрахунку показника працездатності моделі «Коефіцієнт детермінації» R^2 наведено на рис. 11. Фрагмент програмного коду розрахунку показника працездатності моделі

```
// Функція для побудови графіку розподілу Гаусса у вигляді гістограми та перевірки методом трьох сигм
export function plotGaussianDistribution(x, chartRef2) {
  // Підготовка даних для побудови
  const mean = math.mean(x); // Середнє значення x
  const variance = math.variance(x); // Дисперсія x
  const stdDev = Math.sqrt(variance); // Стандартне відхилення

  // Обчислення значень гауссівської функції для кожного x
  const data = x.map(value => ({
    x: value,
    y: gaussianPDF(value, mean, variance)
  }));

  // Вивід корисної інформації в консоль
  console.log(`Середнє значення (Mean): ${mean}`);
  console.log(`Дисперсія (Variance): ${variance}`);
  console.log(`Стандартне відхилення (Standard Deviation): ${stdDev}`);

  // Перевірка за методом трьох сигм
  const lowerBound = mean - 3 * stdDev;
  const upperBound = mean + 3 * stdDev;
  const outliers = x.filter(value => value < lowerBound || value > upperBound);

  console.log(`Аномальні значення (за межами 3σ): ${outliers}`);

  new Chart(chartRef2.current.getContext('2d'), {
    type: 'bar', // Використання гістограми
    data: {
      labels: x,
      datasets: [{
        label: 'Гауссівський розподіл',
        data: data.map(point => point.y),
        borderColor: 'rgba(75, 192, 192, 1)',
        backgroundColor: 'rgba(75, 192, 192, 0.2)',
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        x: {
          type: 'linear',
          position: 'bottom',
          title: {
            display: true,
            text: 'Значення X'
          }
        },
        y: {
          type: 'linear',
          position: 'left',
          title: {
            display: true,
            text: 'Густина ймовірності (Probability Density Function)'
          }
        }
      },
      plugins: {
        annotation: {
          annotations: {
            box1: {
              type: 'box',
              xMin: lowerBound,
              xMax: upperBound,
              yMin: 0,
              yMax: Math.max(...data.map(point => point.y)),
              backgroundColor: 'rgba(0, 255, 0, 0.1)',
              borderColor: 'rgba(0, 255, 0, 0.5)',
              borderWidth: 1,
              label: {
                content: '3σ Range',
                enabled: true,
                position: 'center'
              }
            }
          }
        }
      }
    }
  });
}
```

Рис. 5. Фрагмент програмного коду для побудови графіка обчисленої функції щільності Гаусса та перевірки отриманих результатів методом трьох сигм

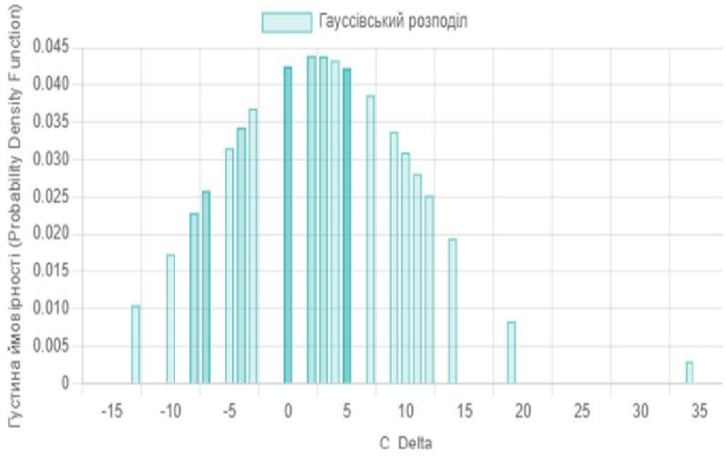


Рис. 6. Приклад візуального представлення графіка обчисленої функції щільності Гаусса та перевірки отриманих результатів методом трьох сигм

```
// Простий метод МНК (метод найменших квадратів)
export function leastSquaresRegression(x, y) {
    const n = x.length;
    const X = math.concat(math.reshape(x, [n, 1]), math.ones([n, 1]), 1);
    const theta = math.multiply(math.inv(math.multiply(math.transpose(X), X)), math.multiply(math.transpose(X), y));
    return theta;
}
```

Рис. 7. Фрагмент програмного коду створення моделі за методом найменших квадратів

«Коефіцієнт прогнозування Q^2 » наведено на рис. 12. Фрагмент програмного коду розрахунку показника працездатності моделі «Коефіцієнт прогнозування Q^2_{LOOCV} » наведено на рис. 13.

```
  while (iteration < maxIterations) {
    let updatedTheta = [0, 0];
    let totalLoss = 0;

    for (let i = 0; i < n; i++) {
        const r = y[i] - (thetaHR[0] * x[i] + thetaHR[1]);
        const loss = huberLoss(r, delta);
        const weight = Math.sqrt(delta / (r * r + delta));

        // Construct weighted matrices
        const weightedX = [X[i][0], X[i][1]]; // Select columns [0, 1] (corresponding to x and the intercept)
        const weightedY = yMatrix[i][0]; // Access element from yMatrix directly using array indexing

        // Update coefficients using weighted least squares
        updatedTheta = math.add(updatedTheta, math.multiply(weight, weightedX, weightedY));

        totalLoss += loss;
    }

    // Перевірка на збіжність за зміною загальних втрат
    if (Math.abs(totalLoss - previousLoss) < tolerance) {
        break; // Зупиняємо ітерації, якщо втрати майже не змінюються
    }

    // Оновлення коефіцієнтів регресії на основі вагованих МНК
    thetaHR = updatedTheta;
    previousLoss = totalLoss; // Оновлюємо попередні втрати для порівняння
    iteration++;
  }

  return thetaHR;
}
```

Рис. 8. Фрагмент програмного коду створення моделі за методом регресії Хубера

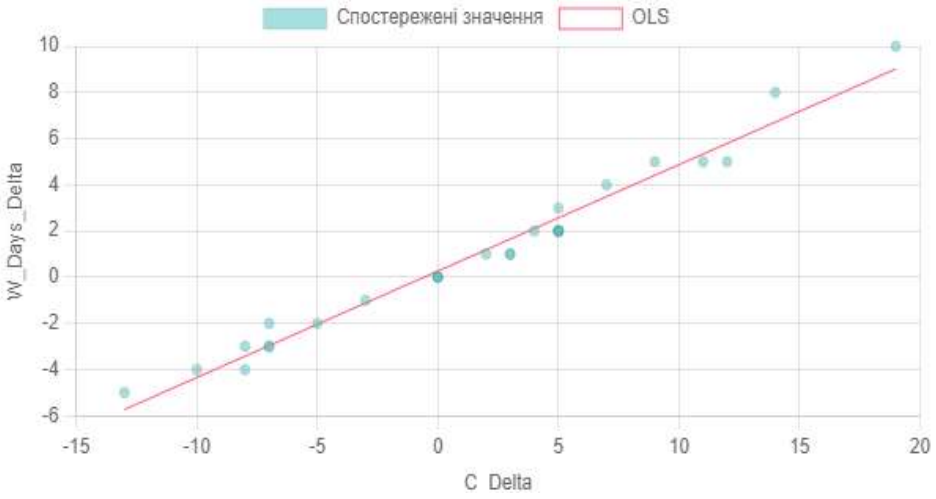


Рис. 9. Приклад візуального представлення моделі, побудованої за методом найменших квадратів (OLS)

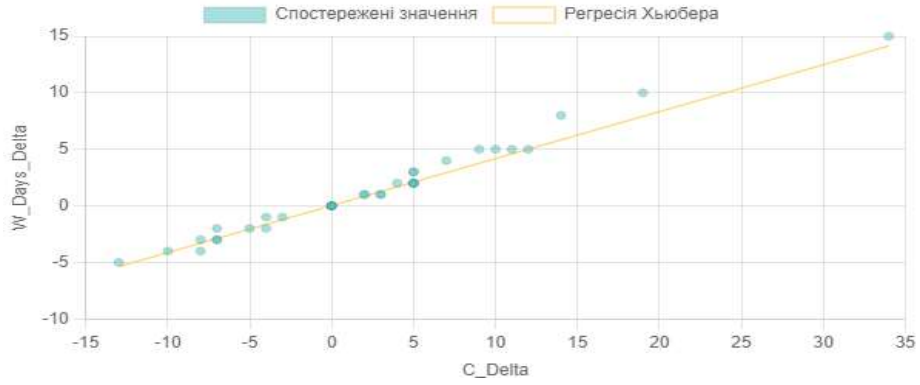


Рис. 10. Приклад візуального представлення моделі, побудованої за методом регресії Хубера

```
function calculateR2(actual, predicted) {
  const meanActual = math.mean(actual);
  const totalSumOfSquares = math.sum(actual.map(val => Math.pow(val - meanActual, 2)));
  const residualSumOfSquares = math.sum(predicted.map((val, i) => Math.pow(actual[i] - val, 2)));
  const r2 = 1 - (residualSumOfSquares / totalSumOfSquares);

  return r2;
}
```

Рис. 11. Фрагмент програмного коду розрахунку показника працездатності моделі «Коефіцієнт детермінації R^2 »

```
// Розрахунок простого Q^2 для регресії
function calculateSimpleQ2(y, yPred) {
  const meanY = math.mean(y);
  const totalSumOfSquares = math.sum(y.map(val => Math.pow(val - meanY, 2)));
  const sumSquaredErrors = math.sum(y.map((val, i) => Math.pow(val - yPred[i], 2)));
  const simpleQ2 = 1 - (sumSquaredErrors / totalSumOfSquares);

  return simpleQ2;
}
```

Рис. 12. Фрагмент програмного коду розрахунку показника працездатності моделі «Коефіцієнт прогнозування Q^2 »

```
// Розрахунок Q^2 за LOOCV для регресії
function calculateQ2LOOCV(x, y, regressionFunction, addParam) {
  const n = x.length;
  let sumSquaredErrors = 0;
  let totalSumOfSquares = 0;

  for (let i = 0; i < n; i++) {
    const xTrain = [...x.slice(0, i), ...x.slice(i + 1)];
    const yTrain = [...y.slice(0, i), ...y.slice(i + 1)];
    const xTest = x[i];
    const yTest = y[i];

    const theta = regressionFunction(xTrain, yTrain, addParam);

    // Перевірка, чи отримано коректні значення коефіцієнтів theta
    if (theta !== null && typeof theta === 'object' && theta.length >= 2) {
      const yPred = theta[0] * xTest + theta[1];
      const squaredError = Math.pow(yTest - yPred, 2);
      sumSquaredErrors += squaredError;

      const meanY = math.mean(yTrain);
      const totalSquare = Math.pow(yTest - meanY, 2);
      totalSumOfSquares += totalSquare;
    } else {
      console.error('Regression function did not return valid coefficients.');
      // Опціонально можна повернути null або інше значення у випадку помилки
      return null;
    }
  }

  const Q2 = 1 - (sumSquaredErrors / totalSumOfSquares);
  return Q2;
}
```

Рис. 13. Фрагмент програмного коду розрахунку показника працездатності моделі «Коефіцієнт прогнозування Q^2_{LOOCV} »

Фрагмент програмного коду розрахунку показника працездатності моделі «Стандартне відхилення (σ)» наведено на рис. 14.

```
// StandardDeviation
function calculateCalculatedStandardDeviation(actual, calculated, numParams) {
  const n = actual.length;
  const residuals = actual.map((val, i) => val - calculated[i]);
  const squaredResiduals = residuals.map(residual => Math.pow(residual, 2));
  const sumSquaredResiduals = math.sum(squaredResiduals);

  const sigmaCalc = Math.sqrt(sumSquaredResiduals / (n - numParams - 1));
  return sigmaCalc;
}
```

Рис. 14. Фрагмент програмного коду розрахунку показника працездатності моделі «Стандартне відхилення (σ)»

Фрагмент програмного коду розрахунку показника працездатності моделі «F-критерій» наведено на рис. 15.

```
function calculateFStatistic(R2, numParams, numDataPoints) {
  const F = (R2 / (1 - R2)) * ((numDataPoints - numParams - 1) / numParams);

  return F;
}
```

Рис. 15. Фрагмент програмного коду розрахунку показника працездатності моделі «F-критерій»

6. Обговорення результатів дослідження

Застосування дескрипторного підходу для автоматизації виконання етапів методу, заснованого на моделі Бекхарда і Гарріса, дозволило вирішити такі задачі:

а) проводити кількісне оцінювання змін на основі значень дескрипторів, які формуються в процесі виконання спринтів довгострокового ІТ-проєкту його виконавцями;

б) автоматизувати вирішення задачі кількісного оцінювання змін шляхом формування і аналізу статистичних моделей залежностей показників змін від дескрипторів.

Розроблена ІТ кількісного оцінювання змін дозволяє підвищити якість управління змінами у довгостроковому ІТ-проєкті. Ця можливість була досягнути завдяки застосуванню дескрипторного підходу та статистичних моделей кількісного оцінювання змін. Використання програмної реалізації розробленої ІТ значно спрощує виконання робіт з оцінювання змін параметрів часу виконання окремих задач довгострокового ІТ-проєкту.

На відміну від похідного методу, заснованого на моделі Бекхарда і Гарріса [7], застосування розробленої ІТ кількісного оцінювання змін дозволяє покращити показники виконання процесу оцінювання змін. Результати порівняння значень цих показників, розрахованих для похідного методу, заснованого на моделі Бекхарда і Гарріса, та після впровадження розробленої ІТ, наведено у табл. 2.

Головним обмеженням використання розробленої ІТ є необхідність експлуатації ІС та ІТ управління роботою проєктних команд, які могли б забезпечити формування похідних масивів даних для подальшого розрахунку значень обраних дескрипторів задач довгострокового ІТ-проєкту. Крім того, суттєвим недоліком розробленої ІТ є значне збільшення тривалості розрахунків моделей при збільшенні кількості дескрипторів, взятих для аналізу. Для подолання цього недоліку необхідно проводити додаткові дослідження зі зменшення часової та обчислювальної складності алгоритмів реалізації розробленої ІТ.

Таблиця 2

Показник	Застосування похідного методу, заснованого на моделі Бекхарда і Гарріса	Застосування розробленої інформаційної технології кількісного оцінювання змін	Зміна показника
Доля (у відсотках) змін часу, скасованих у зв'язку з неможливістю їх реалізації Ch_{Cp}	7	5	Зменшилася на 2
Доля (у відсотках) змін часу, що були прийняті замовником та затверджені як успішно реалізовані Ch_{Sp}	92	95	Збільшилася на 3
Різниця (в годинах) між часом, витраченим на виконання робіт, і оціненим часом, δCh_T	7	2	Зменшилася на 5
Доля (у відсотках) змін, які були завершені вчасно Ch_{iTp}	79	86	Збільшилася на 7
Оцінка ставлення співробітників до процесів управління змінами (від 1 до 10)	6,575	8,025	Поліпшилася на 14,5 %

Дослідження з розвитку розробленої ІТ кількісного оцінювання змін пропонується зосередити на визначенні можливості її застосування для оцінювання та прогнозування інших (окрім часу) параметрів змін, що виникають під час виконання довгострокового ІТ-проєкту. Для проведення таких досліджень буде необхідно також провести додаткові дослідження з визначення найкращих показників, які характеризують зміни та успішність виконання процесу оцінювання змін у ІТ-проєктах різних типів.

7. Висновки

Під час виконання даного дослідження було підвищено точність оцінювання змін часу виконання задач довгострокових ІТ-проєктів. Це підвищення було досягнуто за рахунок розробки ІТ кількісного оцінювання змін на основі існуючого методу управління змінами при реалізації ІТ-проєктів. Як похідний метод оцінювання змін було обрано метод, заснований на моделі Бекхарда і Гарріса. Розроблену ІТ було адаптовано для використання на рівні окремих задач довгострокового ІТ-проєкту із застосуванням дескрипторного підходу. З метою оцінки змін часу виконання задач було запропоновано використовувати статистичні моделі за методом OLS або методом регресії Хубера.

Під час розробки ІТ було визначено особливості дескрипторного підходу для оцінювання змін у довгостроковому ІТ-проєкті. Базуючись на похідному методі управління змінами, заснованому на моделі Бекхарда і Гарріса, було визначено основні етапи та кроки розробленої ІТ та створено опис її архітектури у вигляді діаграми потоків даних. Розглянуто основні особливості реалізації елементів математичного та програмного забезпечення розробленої ІТ.

Встановлено, що застосування розробленої ІТ кількісного оцінювання змін дозволяє покращити значення показників успішності виконання процесу оцінювання змін та оцінки рівня задоволеності команди виконавців від участі у цьому процесі.

Перелік посилань:

1. Сахно Є. Ю., Сідін Е. П., Корнієць К. Є. Моделювання ефективності реалізації довгострокових проектів. Науковий вісник Полісся, 2015. №2 (2), С. 87–94. URL : https://doi.org/10.25140/10.25140/2410-957620152 (2)87-94

2. Сьоме видання Настанови до зводу знань з управління проєктами (Настанова PMBOK) та Стандарт з управління проєктами Project Management Institute, PMI, 2022. 275 с. URL: https://pmiukraine.org/pmbok7/

3. Adkar model. URL: https://www.maxzosim.com/adkar-model/ (дата звернення: 05.04.2024).

4. Adkar Prosci. URL: https://www.prosci.com/ methodology/adkar (дата звернення: 05.04.2024).

5. Aim change management methodology. URL: https://www.imaworldwide.com/aim-change-management-methodology (дата звернення: 05.04.2024).

6. Aim change management methodology. URL: https://www.aim.com.au/leadership-strategy/courses/change-management-embrace-evolve-thrive (дата звернення: 05.04.2024).

7. Beckhard & Harris Change Process. URL: https://praxie.com/beckhard-harris-change-process-online-tools-templates-web-software/ (дата звернення: 05.04.2024).

8. Bridges Transition Model. URL: https://wmbridges.com/about/what-is-transition/ (дата звернення: 07.04.2024).

9. The 8 Steps for Leading Change. URL: https://www.kotterinc.com/methodology/8-steps/ (дата звернення: 07.04.2024).

10. The Kübler Ross Change Curve in the Workplace 2024. URL: https://whatfix.com/blog/kubler-ross-change-curve/ (дата звернення: 08.04.2024).

11. Lewin's 3-Stage Model of Change Theory: Overview. URL: https://whatfix.com/blog/lewins-change-model/ (дата звернення: 15.04.2024).

12. Malhotra V. Single Reference Guide for Scrum Certification (Professional Scrum Master I (PSM I) and Professional Scrum Product Owner I (PSPO I) Certification): Vishal Malhotra, 2020. 169 p.

13. Larman C., Vodde B. Large-Scale Scrum: More with LeSS. Pearson Education, 2016. 368 p.

14. Recent Advances in Total Least Squares Techniques and Errors-in-variables Modeling / editor: Van Huffel S. Philadelphia, 1997. 377 p.

15. Як вибрати правильний технологічний стек для вашого проекту [Електронний ресурс]. URL: https://redstone.agency/blog/yak-vybraty-pravylnyi-tekhnolohichnyi-stek-dlia-vashoho-proektu/ (дата звернення: 10.12.2024).

16. Doar M. Practical JIRA Administration. O'Reilly Media, 2011. 71 p.

17. Diego O., Essam H. H., Salvador H. Metaheuristics in Machine Learning Theory and Applications: CRC Press, 2021. 769 p.

Надійшла до редколегії 10.12.2024 р.

Васильцова Наталія Володимирівна, кандидат технічних наук, доцент, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: nataliia.vasyltsova@nure.ua, ORCID: https://orcid.org/0000-0002-4043-487X (науковий керівник здобувачки вищої освіти Попової Анастасії Вікторівни).

Попова Анастасія Вікторівна, здобувачка вищої освіти, група УПГІТм-22-3, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: anastasiia.popova1@nure.ua