

І. СОЛОВЕЙ, О. ВОРОЧЕК

## УПРОВАДЖЕННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ В ПРОЦЕСИ АВТОМАТИЗОВАНОГО ПРОГНОЗУВАННЯ ПОКАЗНИКІВ ПРОЄКТІВ ІЗ РОЗРОБЛЕННЯ ПРОГРАМНИХ СИСТЕМ

**Предметом дослідження** є процеси автоматизованого прогнозування показників проєктів із розроблення програмних систем, що зазвичай підлягають оцінюванню, а також методи й моделі штучного інтелекту, які можуть бути застосовані для генерації базових шаблонів дорожніх карт і післяопераційних переліків робіт та альтернативних оцінок залежно від контексту. **Мета роботи** – дослідження можливості впровадження та ефективності методів штучного інтелекту у створенні системи для автоматизованого прогнозування альтернативних оцінок програмного продукту. У статті розв'язуються такі **завдання**: визначення етапів, пов'язаних з оцінюванням альтернатив у життєвому циклі проєкту з розроблення програмного продукту; вивчення проблем прогнозування та основних факторів, що впливають на кінцеві показники; дослідження методів прогнозування, що можуть бути впроваджені для реалізації багатоваріантного оцінювання проєкту з розроблення програмного продукту. Стаття присвячена визначенню концептуальних засад створення систем автоматизованого оцінювання та прогнозування на підставі аналізу ефективності обраних моделей машинного навчання. Застосовуються такі **методи**: оцінювання та прогнозування трудовитрат у проєктах із розроблення програмного забезпечення, машинного та глибокого навчання й оцінювання їх ефективності для вирішення проблеми прогнозування. **Досягнуті результати**: визначено концептуальні засади створення систем автоматизованого оцінювання та прогнозування на підставі аналізу ефективності обраних моделей машинного навчання, сфери застосування методів штучного інтелекту в процесі оцінювання показників проєктів із розроблення програмного забезпечення; оцінено показники продуктивності різних моделей машинного навчання за певними параметрами оцінки моделі, які характеризують точність прогнозів; запропоновано концептуальну архітектуру програмного засобу генерації дорожніх карт проєкту з використанням мовної моделі *GPT*. **Висновки**: використання методів машинного та глибокого навчання може підвищити точність прогнозів основних показників проєкту, забезпечити можливість гнучкої генерації різних альтернативних варіантів шаблонів дорожніх карт і післяопераційних переліків робіт, що зробить процес планування та управління більш ефективним і прозорим за умови високого рівня невизначеності вимог до проєкту.

**Ключові слова**: оцінювання проєкту; програмне забезпечення; машинне навчання; генеративні моделі.

### Вступ

У наш час у сфері ІТ-індустрії розробляється все більше й більше проєктів, стартапів і різноманітних продуктів, але далеко не всі з них досягають успіху, тобто завершуються виведенням на ринок продукту або послуги вчасно й у межах бюджету, надають очікувану цінність і відповідають вимогам якості. Основною причиною невдачі проєктів є невідповідна оцінка основних показників розроблення і, як наслідок, вартості. Оскільки оцінка вартості програмного забезпечення є природно складною, люди часто помиляються в прогнозуванні абсолютних результатів. Жодні два проєкти не є однаковими: кожен унікальний за своєю метою, за контекстом виконання та ключовими вимогами. Розв'язання проблем під час розроблення завжди призводить до змін обсягу робіт, часу реалізації, вартості тощо. Унаслідок різноманіття ризиків у проєкті завжди

будуть "невідомі", які можна виявити лише тоді, коли вони виникають.

Крім того, учасники проєктів (зацікавлені особи та сторони, причетні сторони, стейкхолдери) не є однаковими. Кожний з них має власний набір знань, досвіду, цінностей, очікувань, ставлення до ризику та здатності адаптуватися. І цей контекст також змінюється як під час виконання одного проєкту, так і послідовної реалізації різних завдань.

Оцінювання – це ітеративний процес; точність оцінки коливається від дуже приблизної на етапі ідеї та до максимально точної після завершення, і одним із найважливіших завдань менеджера та інших зацікавлених осіб є отримання найбільш вірогідних оцінок на будь-якому етапі життєвого циклу. Занижений бюджет може переконати керівництво розробити нові системи, що згодом перевищать їх бюджет і не досягнуть очікуваної вигоди. Багато вартих уваги проєктів скасовуються через

перевитрати внаслідок неякісних і нереальних кошторисів. У такий спосіб компанія втрачає прибуток і зазнає репутаційних ризиків; неефективною стає майбутня співпраця з клієнтом, а продукт так і не виходить у використання.

З іншого боку, завищені витрати можуть переконати керівництво не розробляти потенційно корисні системи. Коли оцінювачі прогнозують нереально високі витрати, що перевищують максимально допустимі для виправдання нової системи із значними перевагами, керівництво, як правило, відмовляється це схвалювати й втрачає вигоди.

Ще одним питанням, яке має вирішуватись під час оцінювання, – це передбачення впливу різних підходів до розроблення програмного продукту на отримані оцінки основних показників проекту.

В оцінюванні часу, вартості та обсягів робіт необхідно брати до уваги також різні можливі способи реалізації тих чи інших складників програмних систем. Так, наприклад, повторне використання компонентів може як збільшити, так і зменшити трудомісткість і вартість розроблення продукту. Отже, прогнозування застосовується для оцінювання того, якими будуть ці витрати. Прогнози ніколи не можуть бути абсолютно точними, але дають гарну уяву про те, на що чекати, якщо застосовуються відповідні техніки. Деякі техніки прогнозування менш точні, але використовуються для початку планування. Коли стає більш доступною інформація, можуть впроваджуватися більш точні техніки прогнозування. Мета цієї роботи полягає в дослідженні методів прогнозування, що можуть застосовуватися у створенні програмного забезпечення для автоматизованого прогнозування оцінок імовірних варіацій реалізації та модифікування програмного продукту.

### Аналіз останніх досліджень і публікацій

Загальновідомо, що кількість ІТ-проектів завершується, так і не досягаючи заявлених цілей, і тому вважаються неуспішними. Це пояснюється неефективним управлінням проектом, поганим оцінюванням витрат, низькими вимогами тощо [1].

Нещодавня стаття *SaaSList* показала, що один із шести ІТ-проектів перевищує витрати на 200%, а ІТ-проекти з бюджетом щонайменше \$ 1 млн мають на 50% більшу ймовірність невдачі в досягненні бізнес-цілей [2].

Оцінки процесу інженерії програмного забезпечення відіграють важливу роль у розрахунку вартості проекту, водночас оцінка зусиль на розроблення є найважливішим складником для визначення вартісних показників. Протягом останніх десятиліть було проведено доволі значну кількість різноманітних досліджень, що відтворюють складну природу моделі оцінювання та прогнозування ймовірних змін у досягнутих показниках завдяки варіаціям підходів до розроблення та модифікування елементів програмного продукту.

Оцінювання зусиль щодо створення програмного забезпечення (ОЗППЗ, або *SDEE – Software Development Efforts Estimation*) – це процес, що використовують керівники проектів або розробники програмного забезпечення для прогнозування трудовитрат, необхідних для створення системи програмного забезпечення.

Найбільш відомі методи оцінювання різних показників у проектах, пов'язаних із розробленням програмного забезпечення, можна класифікувати таким чином:

#### а) алгоритмічні методи

– вартість конструктивної моделі (COCOMO) [3]. Це підхід до визначення вартості програмного забезпечення, що використовує математичні формули та розрахунки для оцінювання вартості проекту. Він дає приблизну кількість необхідних зусиль, а також розклад проекту програмного забезпечення;

– функціональний точковий аналіз [4]. Оцінює показники системи з функціонального погляду, таким чином розв'язуючи проблеми, пов'язані з технологічною залежністю в життєвому циклі розроблення. Ефективність у програмній інженерії досягається завдяки комплексному аналізу застосунків у три етапи. Перший етап аналізу функціональних точок стосується визначення форм транзакцій, які мають бути здійснені в програмних застосунках. По-друге, інженери оцінюють компоненти програмної системи. Нарешті, процес передбачає оцінювання загальних характеристик системи;

– модель Путнама [5]. Забезпечує простий і надійний спосіб прогнозування витрат на програмне забезпечення. Він упроваджується з метою розрахунку зусиль і часу, необхідних для завершення роботи над програмним забезпеченням, на основі заданого розміру проекту;

#### б) неалгоритмічні методи

– експертне оцінювання. Під час оцінювання витрат здебільшого покладаються на експертизу та

досвід експерта. Це залежить від його предметних знань, а не від історичних показників. Досвідчений фахівець несе відповідальність за оцінку вартості програмного забезпечення на основі того факту, що він має достатні знання, які гарантують максимально точну оцінку вартості;

– оцінювання "згори вниз" [6]. Зазначений підхід до оцінювання витрат зосереджується на визначенні вартості проекту на основі глобальних його властивостей загалом і використання або алгоритмічних (наприклад, модель Патнема), або неалгоритмічних методів. Потім оцінка пропорційно розбивається на різні компоненти;

– оцінювання "знизу вгору" [7]. Повна протилежність попереднього методу. У ньому визначається вартість кожного компонента програмного забезпечення, а потім кінцевий результат досягається за допомогою поєднання цих елементів для отримання загальної оцінки вартості проекту;

– оцінювання ціни до виграшу [8]. У зазначеному підході оцінка програмного проекту прямо пропорційна бюджету замовника. Більше уваги приділяється фінансовим можливостям клієнта, ніж функціональності програмного забезпечення, і проект коштує стільки, скільки замовник має на нього витратити;

*в) моделі, орієнтовані на навчання*

– штучні нейронні мережі. Це один з основних підходів, що використовуються в секторі моделей машинного навчання. Як випливає з назви, це зазвичай надихається нейронною частиною системи мозку з наміром імітувати розумний живий організм. Він складається з двох шарів – вхідного та вихідного; усередині шарів є прихований шар, що містить блоки, основною метою яких є призначення вагових коефіцієнтів для даних, які надходять із вхідних показників. Ці ваги призначаються даним випадково;

– генетичні алгоритми [9]. Визначаються як адаптивні та евристичні алгоритми пошуку, що є предметом теорії природного відбору Дарвіна. Це чи не найактивніша царина досліджень, розроблених за допомогою метаевристички, натхненної природою. Генетичний алгоритм є одним із методів програмного обчислення в процесі оцінювання вартості програмного забезпечення, до того ж його основна роль полягає в зміні певних параметрів класичних методів, таких як підхід СОСОМО, для більш точного прогнозування вартості проекту;

– нечітка логіка. Розгортається для прийняття рішень, завдяки чому її можна реалізувати з різними розмірами та можливостями, починаючи від

невеликих мікроконтролерів і завершуючи розробленням програмного забезпечення на основі значних робочих станцій;

– баєсівські мережі [10]. Застосовують графічні моделі для подання наборів змінних у поєднанні з їх умовними залежностями через спрямований ациклічний графік. Іншими словами, задіяний баєсівський висновок для виконання ймовірнісних обчислень; вони спрямовані на моделювання умовної залежності під час розроблення оцінки вартості програмного забезпечення, яка зазвичай позначається ребрами в орієнтованому графі;

– регресія опорних векторів [11]. Це концепція набору пов'язаних методів навчання під наглядом, які зазвичай упрощуються для аналізу даних у поєднанні з розпізнаними шаблонами. Зазначена модель, застосована в оцінюванні вартості програмного забезпечення, бере набір вхідних показників, а потім дає прогноз кожного вхідного;

– дерево регресії [12]. Поширений підхід, що використовується в оцінюванні вартості програмного забезпечення на основі низки факторів та їх наслідків. Модель, як правило, має форму дерева з різними внутрішніми вузлами та призначена для перевірки атрибута;

– за аналогами. Один із найефективніших підходів в оцінюванні вартості програмного забезпечення завдяки його видатній продуктивності та здатності обробляти складні набори даних. Модель використовує порівняння як основну форму предмета для зіставлення проекту програмного забезпечення, що розглядається, з минулими проектами, які мають попередні відомі характеристики, графік і зусилля.

Цікаво, що, незважаючи на широкий спектр методів і моделей оцінювання, нині найбільш уживаними залишаються методи оцінювання "згори вниз", "знизу вгору", за аналогами та їх різновиди.

Оцінювання за аналогами – це техніка для визначення різноманітних параметрів проекту та показників масштабу. Цей тип оцінювання може бути корисним у разі наявності досвіду проектів в одній сфері, який можна застосувати до іншої галузі. Параметри проекту, які можна виміряти, передбачають його вартість, бюджет, обсяг та очікувану тривалість. Показники проекту, які можна оцінити за допомогою цієї техніки, коливаються залежно від розміру, вагомості та складності завдання. Оцінки отримуються способом зіставлення поточної діяльності з діяльністю, яка мала місце раніше, і проведення порівнянь пропорційно цьому.

Техніка часто впроваджується для оцінювання розміру певного параметра, коли інформація щодо цього параметра в межах поточного проєкту обмежена або не доступна до пізнішої дати. Оцінювання за аналогами здебільшого є формою експертного судження, яке є найбільш надійним не лише тоді, коли попередні дії схожі на поточну діяльність фактично, але також традиційно є найбільш ефективним, коли члени команди, що готує оцінювання, мають високий рівень технологічної експертизи та здатні проводити часткові аналогії на підставі досвіду та/або загальнодоступної інформації.

Оцінювання "знизу вгору" є надзвичайно корисною технікою в управлінні проєктами, оскільки дає змогу отримати більш точний показник конкретного компонента роботи. У процесі такого оцінювання кожне завдання розбивається на більш дрібні складники. Потім розробляються індивідуальні кошториси, щоб визначити, що конкретно необхідно для задоволення вимог кожного з цих менших компонентів роботи. Оцінки для менших окремих компонентів потім агрегуються з метою розроблення більшого оцінювання для всього завдання загалом. У цьому разі оцінка для всього завдання найчастіше є набагато точнішою, оскільки дає змогу ретельно розглянути кожну меншу частину, а потім об'єднати ці ретельно продумані оцінки, а не лише отримати одну велику оцінку, яка зазвичай не настільки детально розглядатиме всі окремі компоненти завдання.

Основною проблемою окреслених підходів є те, що вони не передбачають гнучкого механізму багатоваріантного оцінювання. Якщо необхідно розглянути різні варіанти етапів виконання проєкту з розроблення програмного продукту, це призводить до необхідності проведення повного циклу оцінювання для кожного варіанта.

Щодо передбачення наслідків повторного використання коду або внесення змін (модифікацій) у процес розроблення, то найбільш уживаним залишається баєсівський підхід.

Варто зазначити, що наявні на ринку комерційні засоби оцінювання, які застосовуються в реальних проєктах зі створення програмного забезпечення, наразі майже не використовують широкі можливості відповідних моделей і методів, фактично підміняючи процес прогнозування процесом оцінювання, у найкращому випадку визначаючи певний ступінь точності показників.

Найбільш поширеним підходом в абсолютній більшості систем є впровадження методу

*PERT* [13], що може бути в таких програмних продуктах, як, наприклад, *Microsoft Project* ([www.microsoft.com/project](http://www.microsoft.com/project)). У більшості продуктів основна увага зосереджена на автоматизації побудови календарних графіків проєкту на підставі оцінок, розрахованих у ручному режимі, та на аналізі прогресу певних показників проєкту щодо запланованих значень. Прикладами таких систем є *Wrike* ([www.wrike.com](http://www.wrike.com)), *ClickUp* ([www.clickup.com](http://www.clickup.com)), *Zoho Projects* ([www.zoho.com/projects/](http://www.zoho.com/projects/)), *Jira* ([www.atlassian.com/software/jira](http://www.atlassian.com/software/jira)), *Trello* ([www.trello.com](http://www.trello.com)), *Smartsheet* ([www.smartsheet.com](http://www.smartsheet.com)) тощо.

Проведений аналіз продемонстрував, що проблеми ефективного засобу визначення та прогнозування альтернативних оцінок основних показників проєктів, пов'язаних із розробленням програмного забезпечення, не існує.

### Мета й завдання роботи

---

Мета цієї статті полягає в дослідженні засад створення системи для автоматизованого прогнозування альтернативних оцінок програмного продукту.

Основними завданнями є такі:

- визначення етапів, пов'язаних з оцінюванням альтернатив у життєвому циклі проєкту з розроблення програмного продукту;
- висвітлення проблем прогнозування та основних факторів, що впливають на кінцеві показники;
- дослідження методів прогнозування, що можуть бути використані для реалізації багатоваріантного оцінювання проєкту з розроблення програмного продукту.

### Матеріали й методи

---

У цьому дослідженні припускається, що проєкт із розроблення програмного забезпечення, для якого аналізується процедура оцінювання та прогнозування показників трудовитрат, тривалості або вартості, має типовий життєвий цикл відповідно до моделі, яку запропонував Інститут управління проєктами (PMI). Ця модель містить п'ять основних фаз: ініціація, планування, виконання, продуктивність та контроль і завершення проєкту.

Аналіз основних фаз дає змогу виявити місця в життєвому циклі, що потенційно передбачають прийняття рішень за наявності кількох альтернатив і завдань, які необхідно виконати. Визначаються

підходи до оцінювання, що типово впроваджуються на кожному етапі, та наявні обмеження.

Оскільки кожен з етапів має свої особливості, то й алгоритмічний і математичний апарати, які можуть бути використані на кожному етапі, імовірно, відрізняються. У цьому разі аналізується множина факторів, що впливають на оцінки.

Якщо поглянути на життєвий цикл (рис. 1), то можна побачити, що принаймні три його етапи здатні передбачати альтернативне оцінювання залежно від застосованого контексту та, як наслідок, викликають потребу спрогнозувати основні показники й на їх підставі прийняти рішення щодо подальших етапів проєкту.

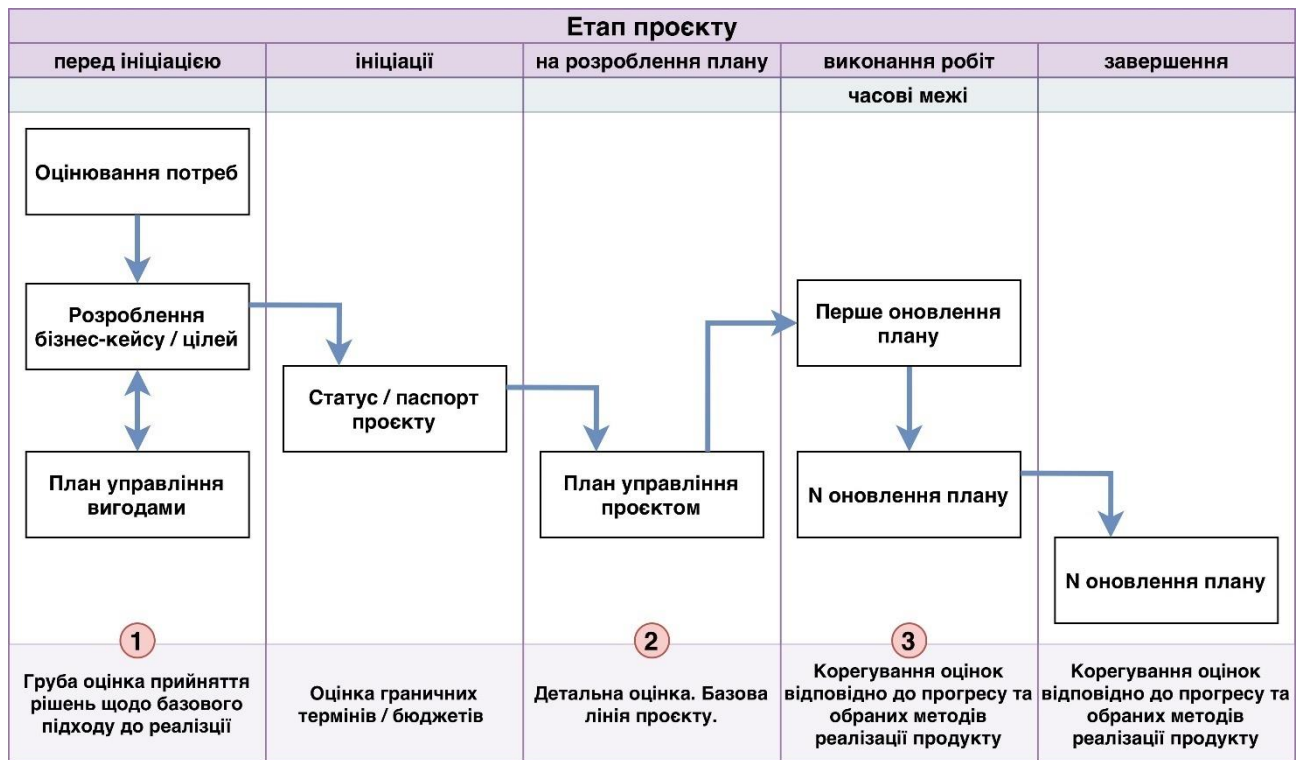


Рис. 1. Оцінювання проєкту залежно від етапу

Так, на етапі концепції (1), особливо якщо проєкт пов'язаний з новою розробкою, основною проблемою є відсутність чіткої інформації щодо структури продукту. Як наслідок, оцінки, якими оперують особи, які приймають рішення, мають низький ступінь вірогідності та точності. Фактично на цьому етапі відсутня можливість побудувати післяопераційний перелік робіт (*WBS, Work breakdown structure*), що призводить до неможливості застосування наявних методів оцінювання, таких як "знизу вгору". З іншого боку, для формування концепції, за умови наявності історичного досвіду виконання подібних проєктів, а також детального визначення контексту системи (як бізнесового, так і технічного), отримання прогнозів (якісних – таких що відтворюють потенційні варіанти реалізації проєкту; кількісних – прогнозних оцінок трудовитрат, вартості, часу залежно від контексту) можливе.

Крім того, зважаючи на сучасний стан розвитку технологій, на цьому етапі доцільно використовувати засоби генеративного штучного інтелекту для формування альтернативних маршрутних карт проєктів з огляду на контекстні характеристики. Необхідно зауважити, що всі прогнозовані оцінки, отримані на цьому етапі, мають низьку точність, але саме вони визначають граничні значення вартості й термінів, які на етапі ініціації фіксуються в статуті проєкту.

На етапі планування (2) проводиться детальна декомпозиція робочого процесу. Природно, що у формуванні оцінок цього рівня менеджери проєктів зосереджуються на діяльності, безпосередньо пов'язаній з основними процесами програмної інженерії. Деталізація післяопераційного переліку робіт може бути різного рівня, але здебільшого в базовому плані проєкту вже береться до уваги

архітектура продукту та дії, які потрібно виконати для забезпечення якісного результату, – такі як робота з вимогами, проектування, розроблення, тестування. Типово виникають питання щодо доцільності повторного використання компонентів, розроблених у минулому, або сторонніх компонентів; можливості розроблення власними силами або передавання робіт на субпідряд; призначення ресурсів на роботи з огляду на баланс продуктивності / якості / вартості тощо. Фактично особа, яка укладає детальний план, має справу з екстремально багатокритеріальним завданням, розв'язання якого на практиці зводиться до формування єдиної детальної маршрутної карти проєкту. Її якість залежить тільки від кваліфікації менеджера, який оцінює, і скоріш за все, вона не є оптимальною.

Упровадження методів машинного навчання (*ML*) на цьому етапі не тільки забезпечить можливість генерації більш точних післяопераційних переліків робіт, але й різних альтернативних декомпозицій, зважаючи на глибокий контекст і наявні припущення та обмеження.

На етапі виконання робіт, окрім проблем, властивих попередньому етапу, також додається необхідність змінити попередні оцінки та передбачити показники варіантів реалізації окремих компонентів для забезпечення вимог щодо термінів і бюджетів основних етапів проєкту. У цьому разі може бути потрібне повернення до оцінок будь-якої попередньої ітерації, уточнення прогнозів різного рівня ієрархії післяопераційного переліку робіт.

З огляду на вищезазначене у створенні програмної системи автоматизованого оцінювання та прогнозування основних показників проєкту з розроблення програмного забезпечення особлива увага має приділятися визначенню методів глибокого (*DL*) та машинного навчання (*ML*), ефективних для прогнозування обсягу зусиль, термінів виконання та вартості різних варіантів реалізації визначеного функціоналу.

Аналіз можливості застосування *ML*-методів оцінювання та прогнозування передбачав розроблення концептуального прототипу програмного продукту оцінювання показників проєктів, пов'язаних з інженерією програмного забезпечення, що надає інструменти прогнозування альтернативних оцінок багатоваріантного процесу розроблення.

Передбачалося, що базовими методами оцінювання в продукті впроваджуватимуться за аналогіями та "знизу вгору", й адаптованість

до альтернатив забезпечуватиметься за допомогою використання *DL* та *ML* [14].

Основними функціональними можливостями такого продукту, пов'язаними безпосередньо з оцінюванням, мають бути:

- формування контексту проєкту (новий проєкт чи модифікація наявного, маршрутна карта розроблення продукту, елементи, реалізація яких передбачає альтернативні рішення);
- вибір технології проєкту та визначення залежностей оцінок проєкту від властивостей застосування відповідних технологій;
- вибір ролей і визначення альтернативних оцінок компетентності / продуктивності / вартості доступних людських ресурсів на кожен роль;
- визначення додаткових необхідних типів ресурсів, що відрізняються від людських;
- генерація альтернативних маршрутів проєкту та відповідних переліків робіт;
- розрахунок прогнозних оцінок для кожного з варіантів.

Існує три основних сценарії використання прототипу, пов'язаних із досліджуваними процесами: попередня генерація узагальненої структури проєкту відповідності контексту з прогнозуванням передпроектних оцінок трудовитрат (1), вартості та термінів реалізації (2); оцінювання під час побудови базового плану проєкту (*project baseline*); оцінювання та прогнозування змін у процесі реалізації (3). *Service Blueprint* для перших двох сценаріїв наведено на рис. 2 і 3.

Контекст у межах цієї роботи – це певний набір параметрів, за допомогою яких може здійснюватися відбір відповідних наявних дорожніх карт або післяопераційних переліків робіт, а для моделювання контексту можуть використовуватися різні підходи, наприклад метаконтекстний обмін інформацією [15].

У межах цього підходу система оперує двома видами знань: контекстними та онтологічними. Семантика того чи іншого об'єкта визначається як  $S(Sk, So) = So^{Sk}$ , де  $Sk$  – контекстні знання про об'єкти;  $So$  – онтологічні описи. Контекстні знання такої системи описують властивості об'єктів предметної галузі, тоді як онтологічні знання описують взаємозв'язки між об'єктами та вплив цих зв'язків на формування семантики об'єктів. Фактично онтологічний складник системи формально описує знання про можливі схеми концептуального опису дорожніх карт / структур *WBS* / опису робіт.

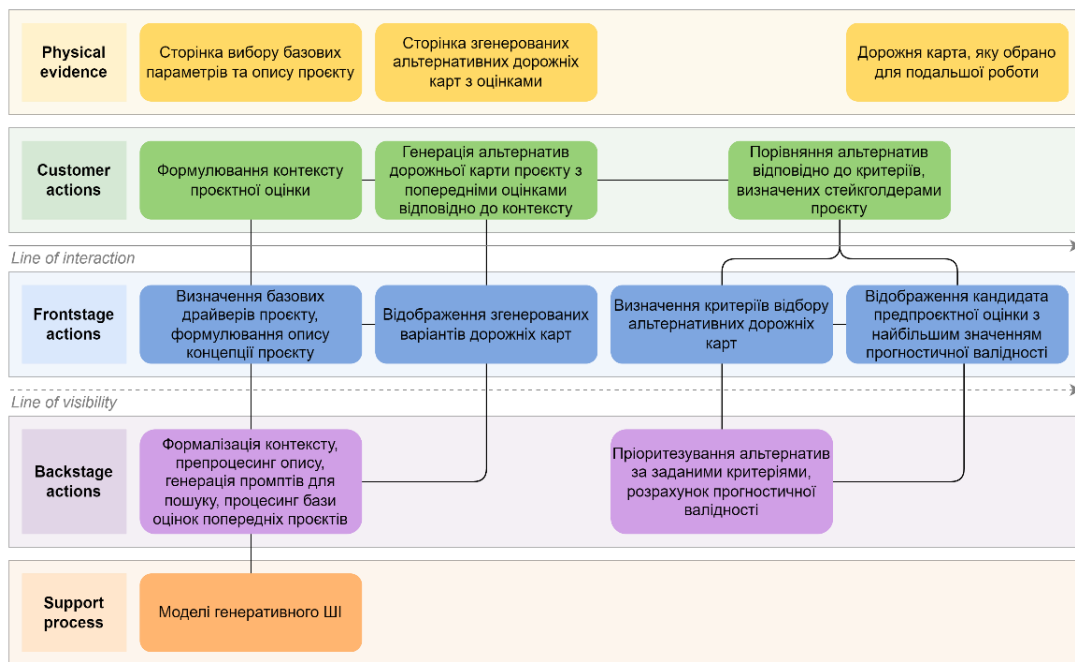


Рис. 2. Сценарій попередньої генерації узагальноної структури проекту та прогнозування передпроектних оцінок

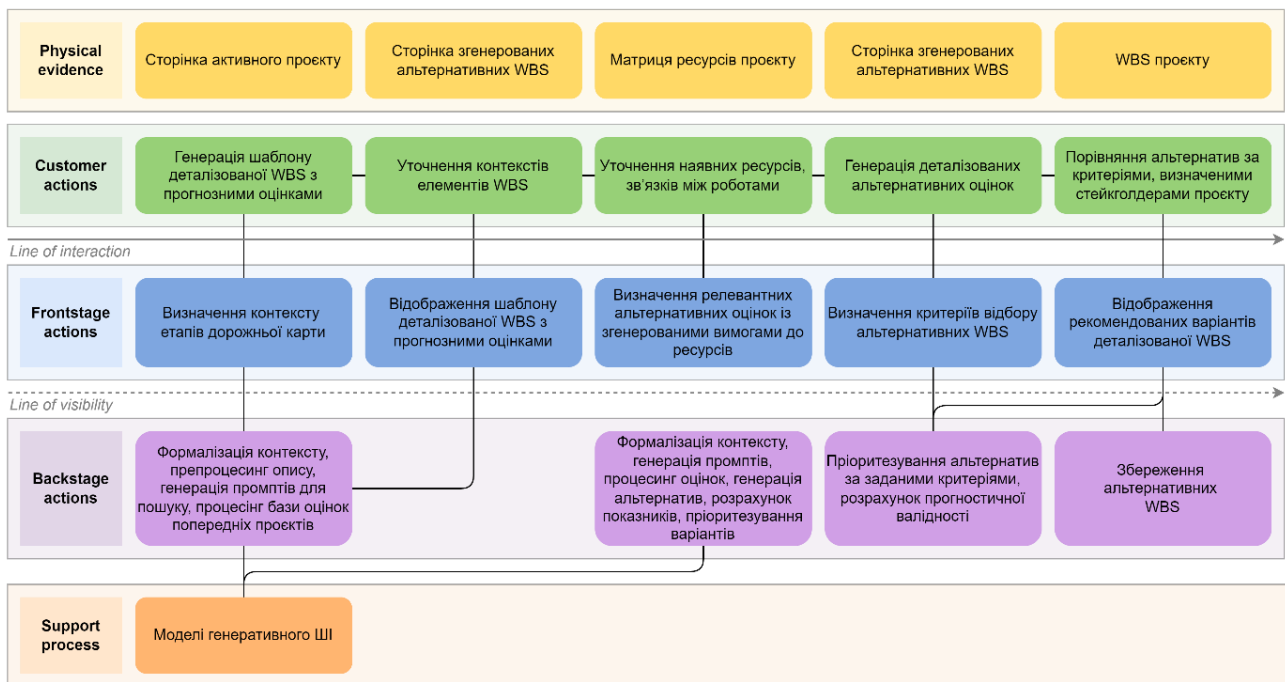


Рис. 3. Сценарій оцінювання показників у процесі побудови базового плану проекту

Препроцесинг описів полягає в обробленні текстових визначень проекту / продукту та у формуванні на їх основі структурованих формальних виразів, що уточнюють параметри контексту.

Щодо процесингу бази попередніх проектів у дослідженні передбачається, що система зберігає попередні оцінки в модульному вигляді,

супроводжуючи їх формальними контекстами, для забезпечення можливості повторного використання як у повному вигляді, так і окремих елементів.

Можливості моделей генеративного ШІ дають змогу отримувати прогнози навіть за умов відсутності історичної бази оцінок. Для ефективного застосування модель має бути донаведена. Сучасний стан розвитку

сфери управління проєктами пропонує значну кількість наявних датасетів з оцінками.

Обчислення показника прогностичної валідності є також одним із важливих аспектів створення цієї системи, оскільки саме цей параметр має суттєвий вплив на прийняття рішень щодо використання того чи іншого варіанта отриманих оцінок.

Прогнозовані оцінки в цьому сценарії ґрунтуються на наявних значеннях попередніх проєктів та загальному показнику продуктивності компанії-розробника, якщо він вказаний як контекстний параметр і використовується як фактор оцінювання.

Для другого сценарію властиві дві схеми оцінювання: побудова детального плану на базі *WBS* з повністю новим кодом та *WBS*, частина елементів якої пов'язана з повторним використанням. У першому випадку методи й алгоритми фактично поглиблюють згенеровану карту, альтернативні оцінки можливі завдяки знанням про наявні ресурси. У другому випадку маємо справу з розгалуженням оцінок структурних елементів різного рівня та прогнозуванням кінцевих показників для кожного з варіантів.

Прогнозовані оцінки базуються на історичній інформації та обчислюються, зважаючи на коефіцієнти продуктивності, доступності та вартості потенційних виконавців для ролей проєкту.

Особливістю третього сценарію є те, що зміни в розробленні можуть потребувати перегляду оцінок (не тільки обраного елемента *WBS*, а й інших, пов'язаних із ним). Має бути спрогнозовано вплив різних варіантів доданих змін на показники проєкту для поточної структури, а також наново обчислено майбутні етапи з огляду на поточну продуктивність проєкту та згенеровані оцінки. На цьому етапі мають обчислюватися коефіцієнти часової та фінансової ефективності поточного процесу, а також продуктивність виконання робіт кожного з елементів післяопераційного переліку робіт як драйверів показників.

Як уже згадувалося вище, під час дослідження особливостей створення системи оцінювання та прогнозування показників проєктів із розроблення програмного забезпечення передбачалося використання методів, моделей та алгоритмів глибокого й машинного навчання, що порівнювалися за різними доступними наборами даних за допомогою восьми параметрів.

У процесі перевірки гіпотези можливості використання методів машинного навчання для

автоматизованого оцінювання робіт було вирішено дослідити три типові моделі: дерева рішень, наївний баєсівський класифікатор та багатошаровий перцептрон.

Дерева рішень (ДР) – це непараметричні контрольовані алгоритми навчання, що застосовуються для розв'язання проблем класифікації та регресії. Метою використання цього алгоритму може бути підготовка моделі прогнозування цільових змінних способом вивчення правил прийняття рішень, отриманих на підставі характеристик даних. Наївний баєсівський класифікатор (НБК) – це класифікатор, що на підставі теореми Баєса визначає ймовірність належності елемента вибірки до певного класу за умови припущення незалежності змінних. Багатошаровий перцептрон (БШП) – це штучна нейронна мережа, контрольований алгоритм навчання. Мережа містить нейрони, розподілені по трьох шарах (вхідні, приховані та вихідні). Дані подаються з нейронів вхідного рівня, прогнозуються нейронами вихідного рівня, рівень абстракції забезпечується прихованими шарами.

У дослідженні розглядалися ці моделі як представники різних типів моделей навчання для перевірки загальної гіпотези щодо можливості їх застосування.

У статті порівнюються результати різних моделей машинного навчання на трьох загальнодоступних наборах даних за допомогою восьми параметрів. Параметри оцінювання моделі містять такі показники: середня абсолютна помилка (САП), середньоквадратична помилка (СКВП), коренева середньоквадратична помилка (КСКВП) та коефіцієнт детермінації (*R*-квадрат). Середня точність оцінки у визначенні зусиль розроблення програмного забезпечення вимірюється за допомогою середньої величини відносної похибки (СВВП), відносної середньої величини помилки (ВВСП), медіанної величини відносної помилки (МВВП) та точності прогнозування.

1. Середня абсолютна похибка (САП, англ. *MAE*) – це середня сума абсолютних похибок.

$$САП = \frac{1}{n} \sum_{i=1}^n |y - \bar{y}_i|, \quad (1)$$

де  $y - \bar{y}_i$  – помилка прогнозування, абсолютна помилка – модуль помилки прогнозування, а САП – середнє суми помилок. У зазначеному рівнянні  $n$  відповідає за загальну кількість даних,  $y_i$  є фактичним значенням, а  $\bar{y}_i$  – передбачуваним значенням.



2. Середньоквадратична помилка (СКвП, англ. *MSE*) – це середнє значення квадратів помилок у наборі даних.

$$СКвП = \frac{1}{n} \sum_{i=1}^n (y - \bar{y}_i)^2. \quad (2)$$

3. Коренева середньоквадратична помилка (КСКвП, англ. *RMSE*) – це міра стандартного відхилення передбачених помилок.

$$КСКвП = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - \bar{y}_i)^2}. \quad (3)$$

4. *R*-квадрат – це метрика відповідності моделі актуальним даним. Це статистичний показник того, наскільки добре апроксимуються фактичні дані в контексті регресії. Формула для обчислення *R*-квадрата має такий вигляд:

$$R^2 = 1 - \frac{ЗалСК}{ЗагСК}, \quad (4)$$

$$ЗВП = \frac{|ДійсніТрудовитрати - ПрогнозованіТрудовитрати|}{ДійсніТрудовитрати}, \quad (5)$$

$$СВВП = \frac{1}{n} \sum_{i=1}^n \left( \frac{|ДійсніТрудовитрати - ПрогнозованіТрудовитрати|}{ДійсніТрудовитрати} \right) \quad (6)$$

або 
$$СВВП = \frac{1}{n} \sum_{i=1}^n (ЗВП)_i. \quad (7)$$

6. Медіанна величина відносної помилки (МВВП – англ. *MdMRE*).

$$МВВП = \text{медіана}(ЗВП)_i. \quad (8)$$

$$ЗПВ = \frac{|ДійсніТрудовитрати - ПрогнозованіТрудовитрати|}{ПрогнозованіТрудовитрати}, \quad (9)$$

$$ВСВП = \frac{1}{n} \sum_{i=1}^n \left( \frac{|ДійсніТрудовитрати - ПрогнозованіТрудовитрати|}{ПрогнозованіТрудовитрати} \right)_i \quad (10)$$

або

$$ВСВП = \frac{1}{n} \sum_{i=1}^n (ЗПВ)_i. \quad (11)$$

8. Точність прогнозування. Середній відсоток оцінок, що перебували в межах *N* % від фактичних значень.

$$\text{Прогн}(N) = \frac{\text{КількістьВідповідних}}{\text{ЗагальнаКількість}} \times 100\%. \quad (12)$$

Аналіз проводився з використанням таких датасетів, як

*China* (<https://zenodo.org/records/268446>),

*Finnish* ([https://figshare.com/articles/dataset/Finnish\\_Effort\\_Estimation\\_Dataset/1334271](https://figshare.com/articles/dataset/Finnish_Effort_Estimation_Dataset/1334271)) та

*Kitchenham* (<https://zenodo.org/records/268457>).

де сума квадратів помилок між початковими значеннями *y* та  $\bar{y}_i$  відома як залишкова сума квадратів (*ЗалСК*). Загальна сума квадратів (*ЗагСК*) вимірює суму квадратів помилок між оригінальним значенням *y* і сумою всіх *y*. Значення *R*-квадрата можуть коливатися від 0 до 1. Якщо значення *R*-квадрата близьке до 1 або дорівнює 1, моделі надається перевага. Якщо *R*-квадрат негативний, це свідчить про відсутність асоціації між даними та моделлю. Цей показник також відомий як коефіцієнт детермінації.

5. Середня величина відносної похибки (СВВП, англ. *MMRE*). Використовує значення відносної помилки (ЗВП – англ. *MRE*), щоб визначити середнє значення відносної помилки. Відносна помилка (ЗВП – англ. *MRE*) обчислюється за такою формулою:

7. Відносна середня величина помилки (ВСВП – англ. *MMER*). Відносно використовує значення помилки (ЗПВ – англ. *MER*), що обчислюється за такою формулою:

## Результати досліджень та їх обговорення

Дослідження виявило низку проблем, пов'язаних із створенням системи автоматизованого оцінювання основних показників проекту та прогнозування впливу на ці показники різних модифікацій як самого продукту, для якого будуються оцінки, так і альтернативних процесів у межах життєвого циклу.

Як уже згадувалося вище, важливим питанням була перевірка продуктивності різних моделей машинного навчання за певними параметрами оцінки моделі. Ці показники характеризують точність прогнозів, що можна отримати. Порівняльні результати для дерев рішень, наївного баєсівського класифікатора та багаточарового перцептрона

наведено в табл. 1. Як бачимо, найвища точність передбачення для датасету *China* властива БШП, для *Finnish* – НБК, для *Kemerer* – ДР. Детальний аналіз досягнутих результатів продемонстрував, що показники загалом залежать від типу даних,

але найбільш придатними для використання в системах прогнозування зусиль на розроблення є ймовірнісні моделі та моделі, основані на нейронних мережах.

Таблиця 1. Показники продуктивності моделей для обраних датасетів

Модель	Прогн (25)	Прогн (50)	САП	СВВП	ВСВП	МВВП	R-квадрат	СКвП	КСКвП
Датасет <i>China</i>									
ДР	22.66%	46.66%	0.0366	1.0713	0.5456	0.5011	0.6409	0.0065	0.0807
НБК	21.33%	47.33%	0.0415	0.9399	0.3359	0.4545	0.6239	0.0068	0.0826
БШП	27.33%	49.33%	0.0357	0.9481	0.3734	0.5025	0.7015	0.0054	0.0735
Датасет <i>Finnish</i>									
ДР	19.67%	45.90%	0.0511	1.3536	0.5668	0.5188	-0.1736	0.0123	0.1109
НБК	23.77%	45.90%	0.0362	1.2171	-2.4677	0.5560	0.6766	0.0034	0.0582
БШП	17.21%	44.26%	0.0403	1.3961	0.0086	0.5225	0.6444	0.0037	0.0611
Датасет <i>Kemerer</i>									
ДР	0%	60%	0.0801	0.2071	0.3295	0.1047	-0.1736	0.0083	0.0911
НБК	20%	40%	0.1046	0.8173	0.8792	0.4589	0.6766	0.0156	0.1247
БШП	20%	40%	0.0703	0.3848	0.3641	0.2510	0.6444	0.0057	0.0757

Дослідження також показало, що в розробленні подібних систем необхідно брати до уваги вплив сетів даних, які використовуються для навчання моделі, а саме специфіку проєктів датасету, їх тип, складність, кількість проєктів у сеті, атрибути тощо.

Поза межами дослідження також залишилися методології оцінювання, основані не на традиційних показниках зусиль (наприклад, у людино-годинах),

а тих, що часто застосовуються в *Agile*-проєктах (наприклад, сторі поінт). Як наслідок, питання ефективності моделей штучного інтелекту для роботи із зазначеними типами проєктів залишається дискусійним.

Другим результатом дослідження став спроектований для демонстрації можливості використання генеративного штучного інтелекту *GPT*-засіб, архітектура якого зображена на рис. 5.

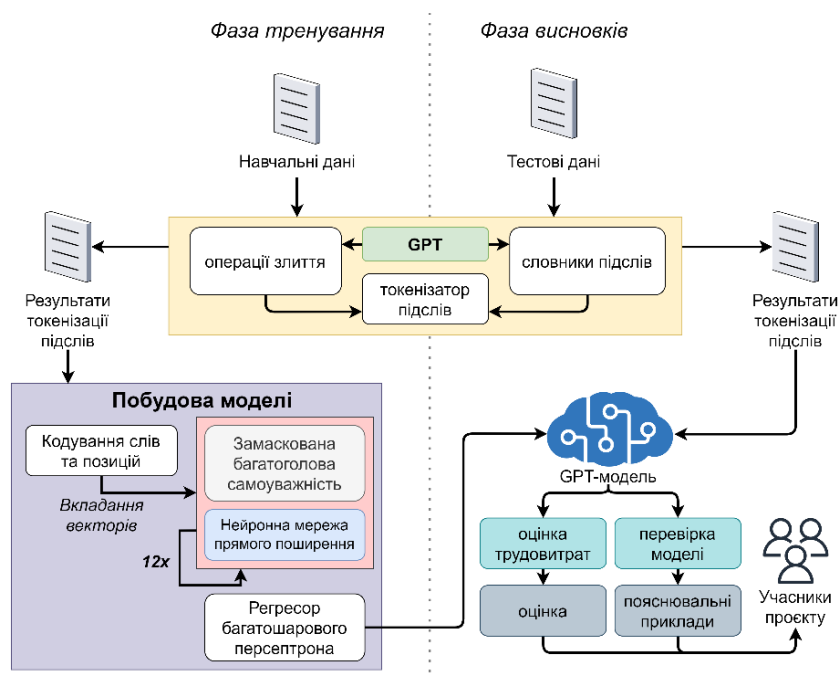


Рис. 5. Огляд архітектури програмного засобу генерації дорожніх карт і відповідних оцінок

Концептуально програмний засіб працює таким чином. Спочатку проводиться токенизація підслів на основі попередньо навченої моделі *GPT*, а далі створюється модель генерації на основі архітектури *GPT*. Для кожного результату токенизації підслова ця модель кодує слова й позиції, створюючи цим вектор вбудовування кожного слова й позиції в результат. Далі вектор подається в архітектуру *GPT*, що містить блоки декодера. Після оброблення декодером вихідний вектор подається в багатосаровий перцептрон для оцінювання елемента дорожньої карти цього результату.

Використання зазначеного засобу дозволило протестувати можливості генеративного ШІ для формування передпроектних оцінок. Важливо зауважити, що відкритим питанням для обговорення залишається вибір мовної моделі (*BERT*, *Gemini*, *LLaMA*) та їх ефективність у розв'язанні поставленого завдання.

### Висновки й перспективи подальшого розвитку

Запропоноване в статті дослідження присвячено концептуальним засадам створення програмних систем автоматизованого обчислення показників проєктів із розроблення програмного забезпечення та прогнозування впливу на показники різних підходів до проєкту. Особливу увагу приділено аналізу можливості використання моделей штучного інтелекту як найбільш трендового підходу в сучасній інженерії систем оброблення інформації та прийняття рішень.

Досягнуті результати підтвердили гіпотезу про те, що методи машинного та глибокого навчання

доцільно впроваджувати для підвищення об'єктивності оцінок проєктів. Саме вони забезпечують можливість отримувати значущі та правильні характеристики заданих у процесі генерації результатів. Проблемним питанням водночас залишається побудова ефективних інструментів для оброблення складних даних та їх високорозмірних варіацій.

Ще одним напрямом подальших досліджень є створення універсального датасету, оскільки через відносно незначні розміри та специфічність наявних датасетів може значно збільшитися час на навчання моделі та виникнути суттєві похибки в прогнозах. Це особливо важливо для сфер діяльності, що мають надзвичайно мінливі контекстні умови, зокрема програмна інженерія, і датасети не є репрезентативними для всього діапазону продуктових та проєктних характеристик. Це призводить до необхідності постійної підтримки тренувального набору відомостей у актуальному стані, долучення нових даних, оновлення алгоритмів, визначення повторюваності та наявності інформації в датасеті тощо.

Цікавим також залишається питання моделювання контексту оцінювання, перспективним виглядає використання принципів метаконтекстного обміну інформацією та адаптації зазначеного підходу до оброблення неструктурованих даних. Цей підхід також може бути запроваджений як інструмент усунення різномірності тренувальних наборів даних.

Необхідно наголосити, що методи машинного та глибокого навчання чуттєві до якості інформації, яка зберігається безпосередньо в датасетах. Унаслідок цього перспективними є дослідження, пов'язані з теорією якості даних.

### Список літератури

1. Lauesen, S. IT project failures, causes and cures. *IEEE Access*. 2020. Vol. 8. P. 72059–72067. DOI: <https://doi.org/10.1109/ACCESS.2020.2986545>
2. SaasList. The State of Project Management in 2023 [42 Statistics]. 2023. URL: <https://saaslist.com/blog/project-management-statistics/> (дата звернення: 15.04.2023).
3. Gupta R. G., Dumka A., Mazumdar B. D. Software Cost Estimation: A Comparative Analysis. 2024 *International Conference on Computer, Electrical & Communication Engineering (ICCECE)*. 2024. P. 1–8. DOI: <https://doi.org/10.1109/ICCECE58645.2024.10497286>
4. Nesma. What is Function Point Analysis (FPA) and what are function points? 2015. URL: <https://nesma.org/faq/function-point-analysis-fpa-function-points/> (дата звернення: 17.04.2024).
5. Brar P., Nandal D. A Systematic Literature Review of Machine Learning Techniques for Software Effort Estimation Models. 2022 *Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*. 2022. P. 494–499. DOI: <https://doi.org/10.1109/CCICT56684.2022.00093>
6. Milošević D. *Project Management ToolBox. Tools and Techniques for the Practicing Project Manager*. Wiley, Hoboken, New Jersey, 2003. ISBN: 9780471208228. 584 p.

7. Wolverton R. W. The Cost of Developing Large-Scale Software. *IEEE Transactions on Computers*. 1974. Vol. C-23. No. 6. P. 615–636. DOI: <https://doi.org/10.1109/T-C.1974.224002>
8. APMP. Competitive Price To Win. 2023. URL: <https://www.apmp.org/assets/BoK-PTW-M-v4.pdf> (дата звернення: 19.04.2024).
9. Affenzeller M., Wagner S., Winkler S., Beham A. *Genetic Algorithms and Genetic Programming. Modern Concepts and Practical Applications*. CRC Press, Boca Raton, Florida. 2009. ISBN: 9781420011326. 379 p.
10. Kim A., Lee D. Dynamic Bayesian network-based situational awareness and course of action decision-making support model. *Expert Systems with Applications*. 2024. Vol. 252, Part A. 124093 p. DOI: <https://doi.org/10.1016/j.eswa.2024.124093>
11. Chong L. W., Rengasamy D., Wong Y. W., Rajkumar R. K. Load prediction using support vector regression. *TENCON 2017 – 2017 IEEE Region 10 Conference*. 2017. P. 1069–1074. DOI: <https://doi.org/10.1109/TENCON.2017.8228016>
12. Elish, M. O. Improved estimation of software project effort using multiple additive regression trees. *Expert Systems with Applications*. 2009. Vol. 36, No. 7. P. 10774–10778. DOI: <https://doi.org/10.1016/j.eswa.2009.02.013>
13. Yunning Z., Xixi S. Research on Improved PERT Model in Analysis of Schedule Risk of Project. *2010 International Conference on E-Business and E-Government*. 2010. P. 2768–2771. DOI: <https://doi.org/10.1109/ICEE.2010.699>
14. Cunnama L. (nee Shillington), Sinanovic E., Ramma L., Foster N., Berrie L., Stevens W., Molapo S., Marokane P., McCarthy K., Churchyard G., Vassall A. Using Top-down and Bottom-up Costing Approaches in LMICs: The Case for Using Both to Assess the Incremental Costs of New Technologies at Scale. *Health economics*. 2016. Vol. 25. P. 53–66. DOI: <https://doi.org/10.1002/hec.3295>
15. Biletskiy, Y., Campeanu, C., Dudar, Z., Vorochek, O. Meta-context mediation to attain semantic interoperability. *2004 2nd International IEEE Conference on 'Intelligent Systems'*. 2004. Vol. 1. P. 238–243. DOI: <https://doi.org/10.1109/IS.2004.1344674>

## References

1. Lauesen, S. (2020), "IT project failures, causes and cures", *IEEE Access*, Vol. 8, P. 72059–72067. DOI: <https://doi.org/10.1109/ACCESS.2020.2986545>
2. SaasList (2023), "The State of Project Management in 2023 [42 Statistics]", available at: <https://saaslist.com/blog/project-management-statistics/> (last accessed 15.04.2023).
3. Gupta, R. G., Dumka, A., Mazumdar, B. D. (2024), "Software Cost Estimation: A Comparative Analysis", *2024 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, P. 1–8, DOI: <https://doi.org/10.1109/ICCECE58645.2024.10497286>.
4. Nesma (2023), "What is Function Point Analysis (FPA) and what are function points?" available at: <https://nesma.org/faq/function-point-analysis-fpa-function-points/> (last accessed 17.04.2024).
5. Brar, P., Nandal, D. (2022), "A Systematic Literature Review of Machine Learning Techniques for Software Effort Estimation Models", *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, P. 494–499, DOI: <https://doi.org/10.1109/CCICT56684.2022.00093>
6. Milošević, D. (2003), *Project Management Toolbox. Tools and Techniques for the Practicing Project Manager*, Wiley, Hoboken, New Jersey, 584 p. ISBN: 9780471208228
7. Wolverton, R. W. (1974), "The Cost of Developing Large-Scale Software", *IEEE Transactions on Computers*, Vol. C-23, No. 6, P. 615–636. DOI: <https://doi.org/10.1109/T-C.1974.224002>
8. APMP (2023), "Competitive Price To Win", available at: <https://www.apmp.org/assets/BoK-PTW-M-v4.pdf> (last accessed 19.04.2024).
9. Affenzeller, M., Wagner, S., Winkler, S., Beham, A. (2009), *Genetic Algorithms and Genetic Programming. Modern Concepts and Practical Applications*, CRC Press, Boca Raton, Florida, 379 p. ISBN: 9781420011326.
10. Kim, A., Lee, D. (2024), "Dynamic Bayesian network-based situational awareness and course of action decision-making support model", *Expert Systems with Applications*, Vol. 252, Part A. 124093 p., DOI: <https://doi.org/10.1016/j.eswa.2024.124093>.
11. Chong, L. W., Rengasamy, D., Wong, Y. W., Rajkumar, R. K. (2017), "Load prediction using support vector regression", *TENCON 2017 – 2017 IEEE Region 10 Conference*, 1069–1074 p. DOI: <https://doi.org/10.1109/TENCON.2017.8228016>
12. Elish, M. O. (2009), "Improved estimation of software project effort using multiple additive regression trees", *Expert Systems with Applications*, Vol. 36, No. 7, P. 10774–10778, DOI: <https://doi.org/10.1016/j.eswa.2009.02.013>
13. Yunning, Z., Xixi, S. (2010), "Research on Improved PERT Model in Analysis of Schedule Risk of Project", *2010 International Conference on E-Business and E-Government*, P. 2768–2771. DOI: <https://doi.org/10.1109/ICEE.2010.699>
14. Cunnama, L. (nee Shillington), Sinanovic, E., Ramma, L., Foster, N., Berrie, L., Stevens, W., Molapo, S., Marokane, P., McCarthy, K., Churchyard, G., Vassall, A. (2016), "Using Top-down and Bottom-up Costing Approaches in LMICs: The Case for Using Both to Assess the Incremental Costs of New Technologies at Scale", *Health economics*, Vol. 25, P. 53–66. DOI: <https://doi.org/10.1002/hec.3295>

15. Biletskiy, Y., Campeanu, C., Dudar, Z., Vorochek, O. (2004), "Meta-context mediation to attain semantic interoperability", *2004 2nd International IEEE Conference on Intelligent Systems*, Vol. 1, P. 238–243, DOI: <https://doi.org/10.1101/IS.2004.1344674>

Надійшла (Received) 09.05.2024

#### Відомості про авторів / About the Authors

**Соловей Ілля Владиславович** – Харківський національний університет радіоелектроніки, здобувач вищої освіти факультету комп'ютерних наук, Харків, Україна; e-mail: [illia.solovei@nure.ua](mailto:illia.solovei@nure.ua); ORCID ID: <https://orcid.org/0009-0005-5715-2755>

**Ворочек Ольга Григорівна** – кандидат технічних наук, Харківський національний університет радіоелектроніки, доцент кафедри програмної інженерії, Харків, Україна; e-mail: [olga.vorochek@nure.ua](mailto:olga.vorochek@nure.ua); ORCID ID: <https://orcid.org/0000-0002-9054-9894>

**Solovei Illia** – Kharkiv National University of Radio Electronics, Higher Education Seeker at the Faculty of Computer Science, Kharkiv, Ukraine.

**Vorochek Olga** – PhD (Engineering Sciences), Kharkiv National University of Radio Electronics, Associate Professor at the Department of Software Engineering, Kharkiv, Ukraine.

## IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE METHODS TO THE PROCESSES OF AUTOMATED METRICS FORECASTING FOR SOFTWARE SYSTEMS DEVELOPMENT PROJECTS

The **subject matter** of the article is the process of automated forecasting of project metrics for software development projects that are typically subject to evaluation. It also covers AI methods and models that can be used to generate basic roadmap templates and operational work lists, as well as alternative estimates depending on the context. The **goal** of the work is to study the foundations of creating a system for automated predicting of alternative evaluations of a software product. The following **tasks** were solved in the article: determining the stages of evaluation related to the assessment of alternatives in the life cycle of a software development project; investigating the problems of predicting and the main factors affecting the final indicators; exploring predicting methods that can be used to implement multivariate assessment of a software development project. The following **methods** are used: methods for evaluating and predicting labor costs in software development projects, machine and deep learning, and assessing their effectiveness for solving the prediction problem. The following **results** were obtained: the conceptual foundations for creating automated evaluation and prediction systems based on the analysis of the effectiveness of selected machine learning models were determined, the areas of application for artificial intelligence methods in the process of evaluating software development project indicators were identified, the performance indicators of various machine learning models were assessed based on certain model evaluation parameters that characterize prediction accuracy; a conceptual architecture of a project roadmap generation software tool based on the GPT language model was proposed. **Conclusions:** the use of machine and deep learning methods can improve the accuracy of predictions for key project indicators, provide the possibility of flexible generation of various alternative roadmap templates and operational work lists, making the planning and management process more efficient and transparent under conditions of high uncertainty of project requirements.

**Keywords:** project evaluation; software; machine learning; generative models.

#### Бібліографічні описи / Bibliographic descriptions

Соловей І. В., Ворочек О. Г. Упровадження методів штучного інтелекту в процесі автоматизованого прогнозування показників проєктів із розроблення програмних систем. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024. № 2 (28). С. 153–1165. DOI: <https://doi.org/10.30837/2522-9818.2024.2.153>

Solovei, I., Vorochek, O. (2024), "Implementation of artificial intelligence methods to the processes of automated metrics forecasting for software systems development projects", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (28), P. 153–165. DOI: <https://doi.org/10.30837/2522-9818.2024.2.153>