



**M. M. Seniv**

Національний університет "Львівська політехніка", м. Львів, Україна.

## ЗАСІБ ДЛЯ ПІДБОРУ МЕТОДОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З УРАХУВАННЯМ МЕТРИК ПРОЄКТУ

Проаналізовано найпоширеніші підходи до створення засобів підтримки прийняття рішень для процесу підбору оптимальної методології розроблення програмного забезпечення (ПЗ). З'ясовано, що на сьогодні для досягнення цієї мети використовуються різні методи, переважно багатокритеріальні методи прийняття рішень, але мало застосовуються методи машинного навчання, також у процесі прийняття рішень не повною мірою враховуються метрики проєкту. Визначено, що на цей час існує велика кількість методологій розроблення програмного забезпечення (водоспадна (Waterfall), швидке розроблення додатків (Rapid Application Development), скрам (Scrum), канбан (Kanban) та ін.), які оптимізують процеси у команді та значно полегшують і пришвидшують розроблення програмного забезпечення, але разом з цим, з огляду на різноманітність цих методологій і проєктів, постає питання правильного підбору методології до конкретного проєкту на підставі даних про нього, оскільки кожна методологія розроблення програмного забезпечення розрахована на різні типи команд, кількість їх членів, різні типи проєктів та їхню складність. Виділено найбільш значущі метрики проєкту, такі як: розмір та кваліфікація команди, тип проєкту та бюджетування, тривалість виконання проєкту, ризик зміни вимог, локація команди та інтенсивність комунікації зі замовником. Спроєктовано, реалізовано та відтестовано програмний засіб підбору методології розроблення програмного забезпечення з урахуванням наведених вище метрик проєкту та використанням алгоритму random forest, для програмної реалізації якого було обрано мову програмування – Swift 5, середовище розроблення – XCode 10.1, для збереження даних використовується Google Firebase Realtime Data Base. Розроблено засіб у вигляді клієнт-серверного додатку з логічною частиною на стороні клієнта та з базою даних на базі Google Cloud Firestore Data Base, а для зберігання результатів, введених користувачем розробленого засобу, вибрано нереляційну базу даних Google Firebase Database. Здійснено дослідження залежності ймовірності прогнозу відповідності методології проєкту та швидкодії роботи від обсягу даних в Data Set, а також порівняння ефективності використання алгоритмів Random Forest та Linear Regression для задачі вибору методології розроблення ПЗ.

**Ключові слова:** інженерія ПЗ; життєвий цикл розроблення ПЗ; методи машинного навчання; алгоритм "випадковий ліс"; системи підтримки прийняття рішень.

### Вступ

Процес створення ПЗ, як і будь-яка інша інтелектуальна діяльність, базується на людських судженнях і висновках, тобто є творчим. Унаслідок спроби автоматизувати цей процес мають тільки обмежений успіх. CASE-засоби можуть допомогти в реалізації деяких етапів процесу розроблення ПЗ, але мало допомагають на тих етапах, де істотним є фактор творчого підходу до розроблення [6]. У процесі розроблення програмного забезпечення однією з основних задач є вибір методології розроблення ПЗ. *Методологія розроблення ПЗ* – це система принципів, сукупність ідей, понять, характеристик, методів і засобів, які визначають стиль розроблення ПЗ [5]. Від вибору методології розроблення залежить успішність реалізації програмного продукту, тому цей крок є дуже важливим. Але через те, що методологій існує дуже багато (негнучкі – водоспадна (Waterfall), швидке розроблення додатків (Rapid Application Development); гнучкі – скрам (Scrum), канбан (Kanban)

та ін.), для менеджерів та розробників стає справжнім викликом визначити таку, яка б найкраще відповідала проєктній задачі та колективу розробників. На практиці часто буває так, що для реалізації того чи іншого проєкту вона вибирається неправильно, внаслідок це може негативно впливати на час реалізації проєкту, збільшення вартості розроблення ПЗ, зниження його надійності [8] та призводити до інших непередбачуваних ризиків. Здебільшого в одній команді використовується тільки одна методологія розроблення програмного забезпечення, що призводить до втрати продуктивності під час розроблення специфічних та нетипових проєктів, які не підходять для розроблення за цією методологією. Різні типи програмних проєктів потребують різних підходів, оскільки кожна категорія проєктів має різні пріоритети і цілі. Досі не сформульовано чітких стандартизованих критеріїв підбору методології. Тому під час вибору методології часто не враховують певні фактори, які потім виявляються важливими. Відповідно, актуальною науковою задачею є розроблення

### Інформація про авторів:

**Сенів Максим Михайлович**, канд. техн. наук, доцент, кафедра програмного забезпечення. Email: [max1sudden@gmail.com](mailto:max1sudden@gmail.com);

<https://orcid.org/0000-0003-1044-4628>

**Цитування за ДСТУ:** Сенів М. М. Засіб для підбору методології розроблення програмного забезпечення з урахуванням метрик проєкту. Науковий вісник НЛТУ України. 2021, т. 31, № 3. С. 120–125.

**Citation APA:** Seniv, M. M. (2021). Suite for selection of software development methodology with reference to project metrics. *Scientific Bulletin of UNFU*, 31(3), 120–125. <https://doi.org/10.36930/40310318>

засобів вибору методології розроблення ПЗ, зокрема, з використанням методів машинного навчання та можливою урахування метрик проекту.

*Об'єкт дослідження* – процес вибору методології розроблення ПЗ.

*Предмет дослідження* – засоби вибору методології розроблення програмного забезпечення з урахуванням метрик проекту.

*Мета роботи* – розроблення ефективного та гнучкого засобу для вибору методології розроблення програмного забезпечення з використанням методу машинного навчання *random forest* та з урахуванням метрик проекту.

Для досягнення зазначеної мети визначено такі основні завдання дослідження: застосувати метод машинного навчання *random forest* для задачі вибору методології ПЗ; виділити найбільш значущі для вибору методології розроблення метрики проекту; спроектувати архітектуру програмного засобу, розробити структуру бази даних, реалізувати та відтестувати програмний засіб вибору методології ПЗ та перевірити його роботу на реальних ІТ-проєктах.

*Наукова новизна отриманих результатів дослідження* – вперше розроблено засіб для вибору методології розроблення ПЗ з використанням методу машинного навчання *random forest* та з урахуванням найбільш значущих для вибору методології розроблення метрик проекту.

*Практична значущість результатів дослідження* – можливе використання розробленого програмного засобу для підвищення ефективності процесу розроблення ПЗ у частині автоматизації надання рекомендацій керівникам ІТ-проєктів.

*Аналіз останніх досліджень та публікацій.* На сьогодні для автоматизації та оптимізації вибору методології розроблення ПЗ використовують різноманітні підходи. Один з них – це експертні системи, які базуються на правилах [4]. Тут враховуються такі критерії, як розмір додатку, ризику, складність проєкту, надійність, час, розмір команди й експертиза, і базуючись на цих характеристиках може бути запропонована водоспадна методологія, спіральна модель, інкрементна модель, XP, Scrum або RAD. Експертна система [4] використовує модульну архітектуру, засновану на правилах. Анкета (Questionnaire) складається з різних питань щодо характеристик проєкту: тип системи, розмір системи, рівень можливих ризиків, складність, надійність, тощо. Експерти мають можливість оновлювати чи додавати будь-яке питання з цього сховища. "Rule repository" підтримується як набір правил "якщо...то", він надає рекомендації відповідно до характеристик проєкту. "Set of facts" містить факти про рекомендації щодо різних можливих значень у правилах. Відповіді, надані користувачем, розташовуються у відповідних правилах "rule repository", які використовуються "rule engine" для порівняння "set of facts", структурування та відображення рекомендації користувачеві через модуль відображення ("SDLC recommendation display module") [4]. Основним недоліком такого типу систем є складність заповнення бази знань. Варто, щоб при підборі враховувалось якомога більше наявних методологій розроблення ПЗ, також варто враховувати багато різних критеріїв

залежно від характеристик проєкту. Якщо намагались зробити базу знань якомога повнішою, то передбачити всі деталі буде надзвичайно складно, особливо враховуючи, що часто думки експертів розходяться. Також у такому типі систем користувач не може змінити пріоритетність критеріїв.

Інший підхід описано у роботі [5], де запропоновано підхід до вирішення проблеми вибору Agile методології для малих і середніх проєктів, базуючись на методі багатокритеріального аналізу SMARTER. Запропонований метод підбору методології складається з таких етапів [5]:

1. Визначення набору критеріїв: пропонується 13 критеріїв, пов'язаних з налаштуванням роботи над проєктом, складністю проєкту та управлінням змінами;
2. Встановлення альтернативних рішень: вибір обмежують чотири Agile-методологіями: DSDM, SCRUM, XP2 та Crystal;
3. Побудова матриці оцінок: оцінка методологій відносно критеріїв базується на кількості наукових робіт, в яких зазначається, що певне значення критерію підходить певній методології розроблення ПЗ;
4. Визначення відносної важливості критеріїв та обчислення значення ваг критеріїв за допомогою методу ROC (Rank Order Centroid);
5. Багатоатрибутне значення функції кожної з альтернатив задається агрегацією функцій.

Як результат – альтернативи впорядковуються від найкращої до найгіршої [5].

Також у роботі [2] для підбору практик для організації процесу розроблення ПЗ пропонується використовувати метод PAPRIKA (Potentially All Pairwise Rankings of all possible Alternatives). Попарно оцінюються 31 практика відносно 11 критеріїв. Засіб опитує користувача, і на підставі відповідей формує список практик, які рекомендує використовувати для розроблення проєкту. Метод PAPRIKA заснований на тому, що користувачі висловлюють свої уподобання щодо відносної важливості критеріїв чи атрибутів, які становлять інтерес для прийнятого рішення чи вибору, шляхом попарного порівняння (ранжування) альтернатив [2].

А в роботі [7] описано спосіб підбору техніки тестування проєкту, використовуючи метод аналізу ієрархій АНР та метод TOPSIS. TOPSIS базується на ідеї, що найоптимальніша альтернатива має найкоротшу відстань від позитивного ідеального рішення і найдовшу від негативного ідеального рішення. АНР використовується для обчислення ваг критеріїв. АНР використовує коефіцієнт відносної узгодженості для верифікації узгодженості ваг критеріїв.

У роботі [3] описано засіб, де використано метод вибору методології управління проєктом на підставі нечітких уявлень. У методі застосована анкета з питаннями, які стосуються кількості людей, залучених у проєкт, досвіду роботи замовника з командою, оцінки компетентності команди проєкту менеджером проєкту, звітності за проєктом, ймовірності появи ризикових подій. Для кожної ситуації, заданої в анкеті, за допомогою опитування експертів визначені функції приналежності всіх розглянутих методологій управління проєктами, тобто їх застосовність до конкретної ситуації. Відповідно до відповідей на питання анкети за проєктом, формуються функції приналежності оцінки проєкту за кож-

ним його параметром. Для всіх розглянутих методологій розраховуються їх сумарні зважені відстані від оцінки проекту за анкетною при використанні відстаней Хеммінга і Евкліда. Обирається той підхід, для якого обчислені відстані є мінімальними [3].

З наведеного вище можна зробити висновки, що на сьогодні для автоматизації вибору методології розроблення ПЗ використовуються різні методи, в основному це багатокритеріальні методи прийняття рішень, але не застосовуються методи машинного навчання, також у процесі прийняття рішень не повною мірою враховуються метрики проекту.

## Результати дослідження та їх обговорення

Для вибору методології розроблення ПЗ було використано алгоритм Random Forest [1], який загалом описується так: нехай навчальна вибірка складається з  $N$  прикладів, розмірність простору ознак дорівнює  $M$ , і заданий параметр  $m$ . Усі дерева комітету будуються незалежно один від одного за такою процедурою [1]:

- Генеруємо випадкову підвибірку з повторенням розміром  $n$  з навчальної вибірки.
- Будуємо дерево рішень, яке класифікує приклади цієї підвибірки, причому в ході створення чергового вузла дерева будемо вибирати ознаку, на підставі якої проводиться розбиття, не з усіх  $M$  ознак, а тільки з  $m$  випадково вибраних.
- Дерево будується до повного вичерпання підвибірки і не піддається процедурі відсікання.

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить об'єкт, який класифікується до одного з класів, і перемагає клас, за який проголосувало найбільше число дерев. Оптимальне число дерев підбирається у такий спосіб, щоб мінімізувати помилку класифікатора на тестовій вибірці. У разі її відсутності, мінімізується оцінка помилки out-of-bag: частка прикладів навчальної вибірки, неправильно класифікованих комітетом, якщо не враховувати голоси дерев на прикладах, що входять в їх власну навчальну підвибірку [1].

Також визначено найбільш значущі метрики проекту, на підставі яких і буде здійснений його аналіз та підбрана методологія розроблення ПЗ. Запропоновано такі метрики (та їхні значення), які будуть враховуватися у розрахунку найбільш прийнятної методології розроблення ПЗ:

1. Тип проекту (розроблення за нуля; продовження розроблення);
2. Кількість людей на проекті: (мала кількість (2-5 осіб); середня кількість (5-15 осіб); велика кількість (понад 15 осіб));
3. Гнучкість бюджетування (фіксована ціна; погодинна оплата);
4. Кваліфікація команди (низька кваліфікація; середня кваліфікація; висока кваліфікація);
5. Тривалість виконання проекту (1-4 тижні; 1-3 місяці; 4-12 місяців; більше 1 року);
6. Ризик зміни вимог (високий (вимоги часто змінюються); середній (вимоги можуть змінюватись, але не часто); відсутній (усі вимоги суворо визначені наперед і змінюватись не будуть));
7. Локація команди (локально; на відстані; мікс);
8. Інтенсивність комунікації зі замовником (щоденна, два дні на тиждень, щотижня, два рази на місяць).

Для програмної реалізації засобу було обрано мову програмування – Swift 5, середовище розроблення – XCode 10.1, для збереження даних використовується

Google Firebase Realtime Data Base. Засіб розроблено як клієнт-серверний додаток з логічною частиною на стороні клієнта та з базою даних на базі Google Cloud Firestore Data Base. На рис. 1,а зображено діаграму розгортання засобу вибору методології розроблення програмного забезпечення. Для зберігання результатів, введених користувачем розробленого засобу, було вибрано нереляційну базу даних Google Firebase Database, оскільки це є віддалена база даних і дає змогу зберігати дані окремо від пристрою. На рис. 1,б зображено графічне подання одного зі записів.

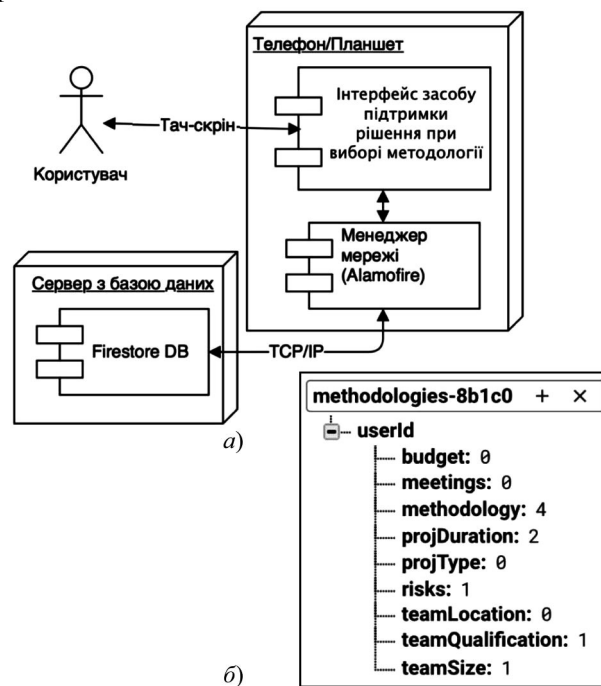


Рис. 1. Діаграма розгортання (а), графічне подання бази даних (б)

На рис. 2 зображено мокапи інтерфейсу користувача для засобу вибору методології розроблення програмного забезпечення. Після інсталяції програми на мобільний користувачу запропоновано ввести таку інформацію про проект:

- назва проекту;
- тип проекту;
- чисельність команди;
- гнучкість бюджетування;
- кваліфікація команди;
- тривалість проекту;
- ризик зміни вимог.

Треба зазначити, що користувачу потрібно заповнити всі запропоновані поля – в іншому разі додаток не розпочне підбір методології розроблення програмного забезпечення. Також для зручності використання було вирішено не давати користувачу вводити дані з клавіатури, а використати випадючі вікна для вводу користувачем даних.

На рис. 3,а,б зображено процес вводу інформації про проект, на підставі якої і буде визначено найбільш прийнятну методологію розроблення ПЗ.

Після того як користувач заповнить усі запропоновані йому поля, програма обрахує дані і покаже найкращий результат на екрані. На рис. 3,в зображено вивід результату на екран користувача. Після заповнення всіх даних програмний засіб видає назву та короткі відомості про найоптимальнішу для цього проекту методологію розроблення ПЗ. Окрім цього, засіб дає змогу переглянути історію роботи з різними проектами, можли-

вість залишити відгук про вибрану методологію розроблення програмного забезпечення для кожного з проектів для наповнення бази знань. На рис. 4 зображе-

но екран відображення історії підбору методології розроблення програмного забезпечення користувачем.

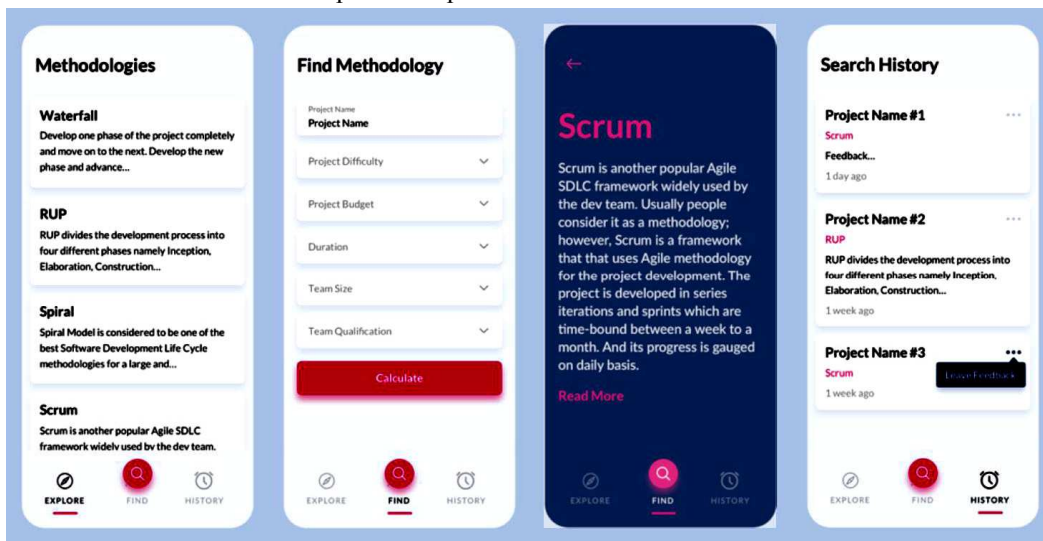


Рис. 2. Приклади інтерфейсу користувача програмного засобу

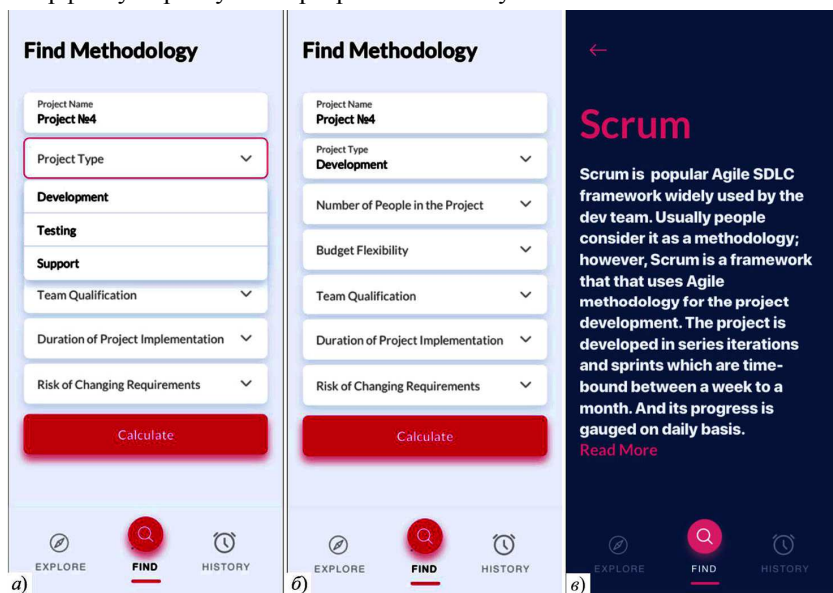


Рис. 3. Зовнішній вигляд екрана користувача для вводу даних (а, б) та екран користувача з підбраною методологією (в)

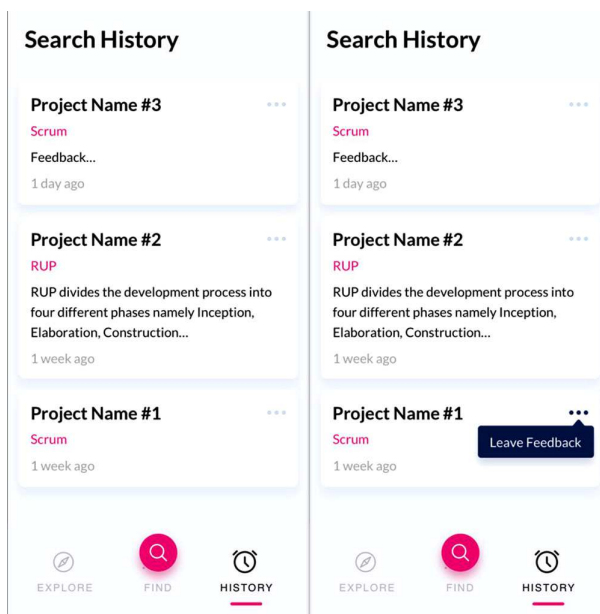


Рис. 4. Екран відображення історії підбору

Для верифікації роботи розробленого засобу та алгоритму random forest було здійснено дослідження залежності ймовірності прогнозу відповідності методології проекту та швидкодії роботи від обсягу даних у Data Set. Для моніторингу швидкодії розробленого ПЗ було використано стандартний клас мови Swift Timer Class, який надає набір методів і властивостей, які призначені для точного вимірювання витраченого часу. Результати верифікації наведено у табл. 1.

Табл. 1. Результати верифікації розробленого засобу (random forest)

Кількість даних в Data Set	Ймовірність прогнозу	Час роботи, с
100	0,83	0,21
200	0,82	0,33
300	0,85	0,35
400	0,85	1,38
500	0,91	1,40
1000	0,90	2,20
1500	0,94	3,20

Як не важко помітити, прийнятна ефективність прогнозу (ймовірність більше 0,9) наявна за мінімаль-

ної кількості даних в Data Set більше 500. Графічно ці залежності подано на рис. 5. Варто також зазначити, що імовірність правильного визначення методології розроблення має прямо пропорційну залежність від достовірності даних у Data Set.

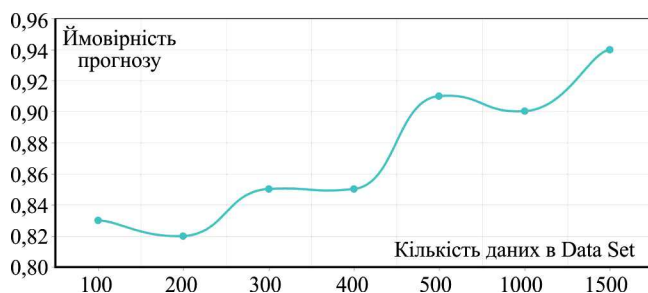


Рис. 5. Графічне подання залежності ймовірності прогнозу від обсягу даних

**Обговорення результатів дослідження.** Під час аналізу результатів дослідження було прийнято рішення перевірити ефективність розробленого засобу з використанням алгоритму random forest, і для цього було здійснено порівняння його з алгоритмом лінійної регресії (табл. 2).

Табл. 2. Результати верифікації розробленого засобу (linear regression)

Кількість даних в Data Set	Ймовірність прогнозу	Час роботи, с
100	0,75	0,09
200	0,78	0,11
300	0,84	0,15
400	0,82	0,22
500	0,85	0,27
1000	0,87	1,42
1500	0,88	1,57

Для порівняння алгоритмів було взято ті самі, дані що використовувалися у роботі засобу з використанням алгоритму RandomForest. Також було записано час швидкодії та ймовірність підбору адекватної методології розроблення програмного забезпечення під певний проєкт. Графічні залежності ймовірності прогнозу та залежності його від обсягу даних при використанні RandomForest та Linear Regression наведено на рис. 6.

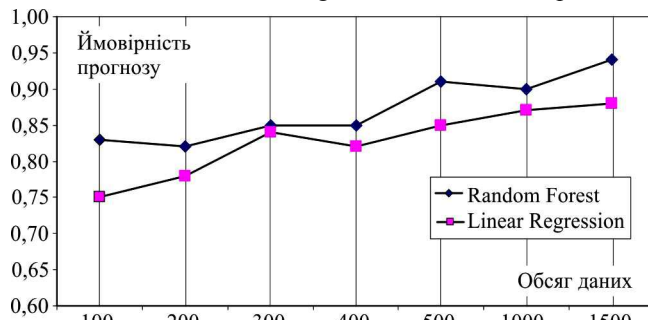


Рис. 6. Графічне подання залежності ймовірності прогнозу від обсягу даних для алгоритмів Random Forest та Linear Regression

Отже, ми можемо побачити на графіку, що показники ймовірності прогнозу з використанням методу Linear Regression є нижчими на 5-7 %, ніж за використання алгоритму Random Forest.

## Висновки

Проаналізовано сучасні підходи до створення засобів автоматизації процесу підбору оптимальної методології розроблення ПЗ. На підставі виконаного аналізу визначено можливість використання методів машинного навчання для вибору методології розроблення. Виділено найбільш значущі метрики проєкту, такі як: тип проєкту, величина команди, тип бюджетування, кваліфікація команди, тривалість виконання проєкту, ризик зміни вимог, локація команди та інтенсивність комунікації зі замовником. Розроблено ефективний та гнучкий засіб для вибору методології розроблення програмного забезпечення з використанням методу машинного навчання random forest та з урахуванням зазначених вище метрик проєкту. Засіб розроблено як клієнт-серверний додаток з логічною частиною на стороні клієнта та з базою даних на базі Google Cloud Firestore Data Base. Проведено верифікацію розробленого засобу та порівняння ефективності використання алгоритмів Random Forest та Linear Regression для задачі вибору методології розроблення ПЗ, яка показала, що показники ймовірності прогнозу з використанням методу Linear Regression є нижчими на 5-7 %, ніж з використанням алгоритму Random Forest.

## References

1. Forest Algorithm. (2020). Random Forest Algorithm Title of resources. Retrieved from: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>.
2. Hernández Ledesma, Gil, Ramos, Erik, Fernández, Carlos, Aguilar Cisneros, Jorge, Rosas-Sumano, Juan, & Morales-Ignacio, Luis. (2017). Selection of Best Software Engineering Practices: A Multi-Criteria Decision Making Approach. *Research in Computing Science*, 136(136), 47-60. <https://doi.org/10.13053/racs-136-1-4>
3. Kononenko, I. V., & Lutsenko, S. Yu. (2017). Method for selection of project management approach based on fuzzy concepts. *Bulletin of NTU "KhPI". Series: Strategic management, portfolio, program and project management*, 2(1224), 8-17. <https://doi.org/10.20998/2413-3000.2017.1224.2>
4. Kumar, K., & Kumar, S. (2013). A Rule-based Recommendation System for Selection of Software Development Life Cycle Models. *ACM SIGSOFT Software Engineering Notes*, 38, 1-6. <https://doi.org/10.1145/2492248.2492269>
5. Silva, V. B. S., Schramm, F., & Damasceno, A. C. (2016). A multicriteria approach for selection of agile methodologies in software development projects. 2016 *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, 2056-2060. <https://doi.org/10.1109/SMC.2016.7844542>
6. Sommerville, Ian. (2011). *Software engineering*. Boston: Pearson, 260 p.
7. Victor, M., & Upadhyay, N. (2011). Selection of Software Testing Technique: A Multi Criteria Decision Making Approach. In: Nagamalai, D., Renault, E., & Dhanuskodi, M. (Eds.). Trends in Computer Science, Engineering and Information Technology. *CCSEIT 2011. Communications in Computer and Information Science*, 204, Springer, Berlin, Heidelberg, 453-462. [https://doi.org/10.1007/978-3-642-24043-0\\_46](https://doi.org/10.1007/978-3-642-24043-0_46)
8. Yakovyna, V., Seniv, M., & Symets, I. (2020). The Relation between Software Development Methodologies and Factors Affecting Software Reliability, 2020 *IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT)*, 377-381. <https://doi.org/10.1109/CSIT49958.2020.9321937>

## **SUITE FOR SELECTION OF SOFTWARE DEVELOPMENT METHODOLOGY WITH REFERENCE TO PROJECT METRICS**

This paper describes the development of a tool for selecting a software development methodology using machine learning with reference to project metrics. To achieve this goal, we decided to apply the method of random forest machine learning for the task of choosing the software methodology. We identified the most important project metrics for the choice of software development methodology, designed the software architecture, developed a database structure, implemented and tested the software methodology. We performed the analysis of the most common approaches to creating decision support tools for the process of selecting the optimal software development methodology, which has revealed that currently various methods are used to automate the choice of software development methodology, mainly these are multi-criteria decision-making methods, but machine learning methods are not applied, and project metrics are not fully taken into account in the decision-making process. Therefore, we decided to use the Random Forest algorithm to support decision-making in the process of choosing the software methodology. The most significant project metrics and their range values were also identified, including as follows: project type, team size, budget type, team qualification, project duration, risk of requirement changes, team location and intensity of communication with the customer. In order to provide project managers with practical decisions on the choice of software development methodology, a software tool was developed that allows taking into account the above project metrics and uses the Random Forest algorithm. For software implementation of the tool, the Swift 5 programming language was chosen, XCode 10.1 development environment, Google Firebase Realtime Data Base was used in order to guarantee data saving. The tool is designed as a client-server application with a logical part on the client side and a database based on Google Cloud Firestore Data Base. A non-relational Google Firebase Database has been selected to store the results entered by the user of the developed tool. In order to verify the operation of the developed tool and the random forest algorithm, we conducted the study of the dependence of the probability of forecasting for the compliance of the project methodology and performance on the amount of data in the Data Set, as well as comparing the efficiency of Random Forest and Linear Regression algorithms for software development methodology. As a result of the research, an effective and flexible tool was developed to select the software development methodology taking into account the most important project metrics and using the Random Forest machine learning method, the probability of which is 5-7 % higher than using Linear Regression.

**Keywords:** Software engineering; software development life cycle; machine learning methods; random forest algorithm; decision support system.