

## АНОТАЦІЯ

*Курінько Д.Д.* Моделі та методи машинного навчання для виявлення та усунення антипатернів в програмних компонентах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки. – Національний університет «Одеська політехніка», МОН України, Одеса, 2026.

У **вступі** показана актуальність вирішення задач автоматизованого виявлення потреби в рефакторингу об'єктно-орієнтованого програмного коду в сучасних програмних системах. Підкреслено недоліки наявних моделей і методів виявлення рефакторинг-можливостей, зокрема підходів на основі «запахів» коду, програмних метрик і методів машинного навчання, які не враховують контекст проєкту, взаємозалежність ознак та зашумленість даних. Визначено об'єкт, предмет, мету, задачі та методи дослідження; наведено наукову новизну та практичне значення отриманих результатів; висвітлено особистий внесок здобувача.

У **першому розділі** дисертаційної роботи виконано системний огляд проблематики виявлення та усунення антипатернів у програмних компонентах як ключового чинника зниження внутрішньої якості та ускладнення еволюції промислових програмних систем. Показано, як зростання масштабів і складності програмного забезпечення призводить до накопичення дефектів проєктування та зв'язок антипатернів з деградацією підтримуваності, ускладненням модифікацій та підвищенням вартості розвитку. На цій основі сформульовано постановку задачі автоматичного виявлення та усунення антипатернів і окреслено місце таких рішень у сучасному інженерному процесі.

Далі наведено класифікацію антипатернів і показано, що їх прояви не зводяться до одного типу сигналів: узагальнено структурні, семантичні та еволюційні (історичні) ознаки, а також підкреслено мультимодальність проявів і пов'язану з цим проблему невизначеності. Після цього проаналізовано традиційні

підходи (метрики, правила, детектори), а також окремо розглянуто структурні і текстові/LLM-орієнтовані підходи, із фокусом на їх сильні сторони та обмеження з погляду практичного застосування.

Окремо були проаналізовані сучасні підходи на базі машинного та глибинного навчання: описано класичні моделі на метриках і мітках «запахів» коду, графові та репрезентаційні моделі подання коду, гібридні рішення та uncertainty-aware підходи, а також напрями, де ці підходи використовується для підтримки рефакторингу та роботи з дублікатами коду.

Аналіз робіт в області автоматичного усунення антипатернів показав, що існуючі моделі, як правило, спираються на один тип ознак та не відображають багатовимірну природу антипатернів, що спричиняє хибні спрацювання або пропуски. Для подолання цього недоліку була запропонована багаторівнева модель на гібридному графовому поданні коду з об'єднанням структурних, семантичних і еволюційних ознак в єдиному просторі.

Подальший аналіз існуючих систем виявлення антипатернів показав, що детекція антипатернів на рівні методу/класу не виявляє компонентні/системні антипатерни та не враховує динаміку якості між релізами. Для усунення цього недоліку було запропонованого багаторівневе подання та аналіз із урахуванням зв'язків між рівнями, а також метод композиції «мінімально інвазивних» послідовностей рефакторингів, що враховує причинно-наслідкові залежності змін.

Аналіз сучасних підходів виявлення антипатернів показав, що вони розвиваються розрізнено, тому можуть давати суперечливі висновки та рекомендації. Було показано, що перспективним є використання гібридних подань, які інтегрують ознаки в єдиний вектор. У якості такого рішення, в роботі було запропоноване уніфіковане гібридне подання як спільну основу і для детекції антипатернів, і для оптимізації планів рефакторингу.

Подальший аналіз методів виявлення антипатернів показав, що їх робота базується на припущенні, що всі типи антипатернів відомі заздалегідь, що не відповідає сучасній практиці розробки програмного забезпечення. Крім того, більшість методів не надає каліброваної оцінки невизначеності прогнозів, що

підвищує ризик агресивних або, навпаки, надто обережних рішень. Для подолання даних недоліків, у роботі пропонується введення open-set/low-confidence режиму в моделі виявлення антипатернів та явного моделювання невизначеності при рекомендації рефакторингів.

Аналіз практики впровадження інструментів виявлення антипатернів показав, що їх результати погано інтегровані в роботу команд та сучасні практики безперервної інтеграції та доставлення. Для подолання цього недоліку, в роботі був запропонований модуль рекомендації рефакторингів, метод композиції послідовностей змін та модель статистичного контролю процесів, що дозволяє формалізовано оцінювати результативність втручань і знижувати ризик хибних рішень.

За результатами проведеного аналізу сформульовано мету та завдання досліджень.

У **другому розділі** дисертаційної роботи уточнено постановку задачі автоматичного виявлення антипатернів з урахуванням їх мультимодальної природи (структура, семантика, метрики, еволюція), багаторівневого контексту («метод – компонент – проєкт») та потреби open-set/low-confidence режиму з явним урахуванням невизначеності. Обґрунтовано обмеження правил та метрик і одно-модальних підходів на базі машинного навчання та сформульовано вимоги до нової моделі (переносимість, інкрементальність для безперервної інтеграції та доставлення).

Запропоновано гібридне графове подання коду у вигляді Code Property Graph з додатковими семантичними зв'язками та 4-канальними ознаками вузлів (структурні, семантичні ембеддинги, метричні, еволюційні), включно з нормалізацією на рівні проєкту та інкрементальним оновленням.

Розроблено багаторівневу модель – на локальному рівні поєднано послідовнісний енкодер і гетерогенну GNN над підграфом CPG зі зв'язком span-to-node та fusion; на рівні компонента виконано агрегацію локальних дескрипторів; на проєктному рівні побудовано граф взаємодії компонентів і застосовано «message passing», а фінальне подання отримано через ієрархічну увагу.

Для прийняття рішень використано мультиміткову класифікацію з оцінюванням невизначеності (ентропія/енергія, епістемічна невизначеність) та механізмом селективного передбачення або відмови; формалізовано відповідні функції втрат. Також описано конвеєр застосування моделі в індустріальних умовах (у тому числі, для pull request), засоби інтерпретованості та експериментальну перевірку на мультимовному корпусі з порівнянням з базовими та абляційними варіантами.

Таким чином, сформульовано **перший пункт наукової новизни**: *вперше* запропоновано багаторівневу модель виявлення антипатернів на основі гібридного графа, що уніфікує синтаксичні дерева, графи потоку керування і графи залежностей у граф властивостей коду, з додатковими семантичними та еволюційними ознаками. Запропонована модель дозволяє виявляти типові і нетипові антипатерни, працювати з багатомовними програмними проектами і забезпечувати сумісність із процесами безперервної інтеграції та доставлення.

В **третьому розділі** дисертаційної роботи запропоновані моделі та методи рекомендації, планування та процесного оцінювання рефакторингів для усунення антипатернів.

**Для безпечного, прозорого автоматизованого усунення дублікатів коду із прогнозованим впливом на процес** був запропонований метод рекомендації рефакторингів через багатоцільову оптимізацію з оцінкою невизначеності, спеціалізований на усуненні дублікатів коду

Ідея методу полягає в тому, що для кожної групи дублікатів розглядається кілька можливих рефакторингів. Кожен варіант оцінюється одночасно за трьома цілями – вигода, трудомісткість, СІ-ризик, причому разом із прогнозом обчислюється невизначеність. Далі через багатоцільову (Парето) оптимізацію відбираються найкращі компромісні рішення, а за високої невизначеності система переходить у low-confidence/open-set режим і може утриматися від рекомендації.

Робота запропонованого методу складається з наступних кроків:

- 1) виявлення клонів – формуються групи дублікатів коду (клон-класи) у проекті;

- 2) генерація дій – для кожної групи задається набір кандидатних рефакторингів + варіант «No refactoring»;
- 3) гібридне подання – для фрагментів клонів будується CFG (AST/CFG/PDG) та фіксується контекст компонентів;
- 4) обчислення ознак – обчислюються структурні, семантичні, метричні та еволюційні ознаки, які агрегуються на рівні групи клонів;
- 5) прогноз ефектів – для кожної пари «група–дія» оцінюються вигода, трудомісткість і СІ-ризик;
- 6) невизначеність – через ймовірнісну модель отримуються дисперсія (ентропія) та калібрована впевненість прогнозів;
- 7) low-confidence фільтр – кандидати з високою невизначеністю відсікаються або переводяться в ручний перегляд (open-set);
- 8) парето-відбір – серед решти будуються Парето-оптимальні рішення за (вигода, трудомісткість, ризик);
- 9) ранжування зі штрафом – парето-множина впорядковується з урахуванням політики проєкту та штрафу за невизначеність;
- 10) вихід/утримання – повертається пріоритизований список рекомендацій або рішення «утриматися», якщо безпечних варіантів немає.

Експериментальне дослідження показало, що запропонований метод показав високу якість вибору типу рефакторингу (Accuracy = 0,80 та F1macro = 0,76), а також високу якість пріоритизації (NDCG@5 = 0,76 і U@5 = 0,62) тобто метод формує більш корисні top-k рекомендації та підтримує «обережну» поведінку через відсікання невпевнених рішень.

Таким чином, сформульовано **другий пункт наукової новизни: удосконалено** метод рекомендації рефакторингів через багатоцільову оптимізацію з оцінкою невизначеності, спеціалізований на усуненні дублікатів коду. Використання методу забезпечує безпечне, прозоре автоматизоване усунення дублікатів коду із прогнозованим впливом на процес.

**Для максимізації очікуваної користі від послідовності рефакторингів за умов обмежень інтерфейсних змін і бюджету безперервної інтеграції та**

**доставлення**, був запропонований метод композиції «мінімально інвазивних» послідовностей рефакторингів на основі причинно-наслідкового виведення з історії репозиторію.

Ідея роботи методу полягає в тому, що він буде узгоджений план рефакторингів із набору кандидатів так, щоб максимізувати очікуваний (консервативний) приріст якості та одночасно обмежити інвазивність і ризик: враховуються бюджети змін, сумісність кроків, вплив на API, а також CI/CD-ризик. Невизначеність прогнозів використовується для «обережного» вибору – перевага надається діям із надійним ефектом, а небезпечні/невпевнені варіанти відсікаються або приводять до утримання від рекомендації.

Робота запропонованого методу складається з наступних кроків:

- 1) збір кандидатів у рефакторинги з їхніми оцінками і метаданими впливу;
- 2) опис поточного стану компонентів через метрики якості, антипатерни, еволюційні та CI/CD показники;
- 3) оцінка очікуваного внеска у покращення з урахуванням контексту компонента;
- 4) корекція ефекту за невизначеністю (перехід від точкової оцінки до нижньої довірчої межі);
- 5) формулювання багатоцільової задачі планування (максимізація корисності при мінімізації інвазивності/ризиків та дотриманні бюджетів);
- 6) накладання обмеження сумісності та порядку і інваріанти коректності;
- 7) пошук найкращої допустимої послідовності через евристичний пошук із відсіканням гілок за бюджетами та ризиком;
- 8) формування плану або утримання від рекомендації, якщо безпечного рішення в межах обмежень не існує.

Результати експериментального дослідження показали, що запропонований метод забезпечив середній приріст якості 0,084, при середній інвазивності 0,036 та середній зміні частки невдалих збірок  $-0,004$ .

Таким чином, сформульовано **третій пункт наукової новизни: вперше** запропоновано метод композиції «мінімально інвазивних» послідовностей

рефакторингів на основі причинно-наслідкового виведення з історії репозиторію. Використання методу дозволяє максимізувати очікувану користь за умов обмежень інтерфейсних змін і бюджету безперервної інтеграції та доставлення.

**Для зменшення хибних тривог та забезпечення керованого зворотного зв'язку для безперервної інтеграції та доставлення в процесі проведення рефакторингів,** була запропонована модель оцінювання ефективності рефакторингів на основі статистичного контролю процесів із атрибуцією впливу.

Запропонована модель розглядає рефакторинг не як разову «подію до/після», а як інтервенцію у часовому процесі розробки, тому ефективність змін оцінюється на основі статистичного контролю процесів (SPC) на часових рядах метрик.

Її ключова ідея полягає в тому, щоб у CI/DevOps-умовах відрізнити реальний, стійкий ефект рефакторингу від природних коливань, шуму та супутніх змін (фіч, тестів, залежностей), і формалізовано робити один із трьох висновків: «покращення», «погіршення» або «недостатньо доказів» із контрольованим рівнем хибних сигналів.

У роботі це реалізується як двоконтурний моніторинг: одночасно відстежуються метрики якості коду та процесні CI-метрики, показники попередньо уніфікуються за напрямом «більше = краще», нормуються та агрегуються робастно, а для рядів із залежностями застосовується усунення автокореляції («prewhitening»), щоб зменшити хибні тривоги. Далі момент інтервенції фіксується, і на постінтервенційному відрізку SPC-методами (EWMA/CUSUM та їх узагальненнями) детектуються малі, але стійкі зсуви відносно базового режиму з межами, налаштованими під заданий рівень ризику; за потреби ефект додатково атрибується з урахуванням коваріат (наприклад, churn/зміни тестів), після чого формується підсумковий вердикт і зворотний зв'язок для налаштування політик відбору «безпечних» рефакторингів у наступних ітераціях.

Експериментальне дослідження моделі показало, що найбільш результативним для виявлення зсувів у метриках CI-стабільності є підхід на базі CUSUM: порівняно з EWMA частка хибних тривог зменшується на 16,7%, чутливість зростає на 9,7%, імовірність детекції збільшується на 3,2%, а середній

час до сигналу скорочується на 32,1%, тобто небажані зсуви фіксуються суттєво швидше. Водночас варіант EWMA з бутстреп-межами демонструє консервативнішу поведінку – хибні тривоги також зменшуються на 16,7%, але ціною невеликого падіння чутливості на 2,8% та імовірності детекції на 2,1% і незначного збільшення часу до сигналу (+3,6%).

Таким чином, сформульовано **четвертий пункт наукової новизни**: *удосконалено* модель оцінювання ефективності рефакторингів на основі статистичного контролю процесів із атрибуцією впливу. Використання моделі забезпечує доказові рішення «покращення, погіршення, змін не виявлено» з заданим рівнем довіри, зменшує хибні тривоги та надає керований зворотний зв'язок для безперервної інтеграції та доставлення.

В **четвертому розділі** дисертаційної роботи проведено дослідження інтегрованого інструментального засобу.

Реалізовано та описано інтегрований інструментальний засіб для наскрізного керування антипатернами (end-to-end pipeline):

- побудова гібридного подання коду (CPG + семантичні ембеддинги + VCS-ознаки);
- виявлення антипатернів;
- генерація/багатокритеріальне ранжування рефакторингів (Парето + нижні довірчі межі з урахуванням невизначеності);
- планування мінімально інвазивної послідовності з бюджетами/ризик-обмеженнями;
- SPC-оцінювання ефекту (EWMA/CUSUM);
- формування інженерних артефактів для відтворюваності і трасованості, включно з механізмом утримання за низької впевненості.

Експериментальне дослідження проведено на 7 Java-проектах (JUnit5, Commons Lang, Guava, Spring Boot, Elasticsearch, Hadoop, Jenkins) у фіксованих релізних зрізах із контрольованим середовищем і порівнянням з baseline-підходами. Отримано узагальнений end-to-end ефект: середнє зменшення Technical Debt Index (TDI) на 16,4%, приріст Maintainability Index (MI) у середньому на +5,66,

зниження Cognitive Complexity (CC) на 12,1% та зменшення Clone density на 32,0%. Узгодженість прогнозованого та фактичного ефектів є високою ( $R^2 \approx 0,91$ ; розбіжності «Expected vs Actual» у межах приблизно від  $-1,6$  до  $+0,3\%$ ).

У порівнянні з базовими засобами інтегрована система дає більший сукупний ефект (для класичних аналізаторів покращення близькі до нуля, для часткових рекомендацій – помітно нижчі за інтегровані), а також забезпечує повний цикл «виявлення – план рефакторингу – підтвердження ефекту».

Таким чином, отримані результати підтверджують коректність і ефективність технічних рішень, запропонованих у дисертаційній роботі.

Розроблені в роботі моделі та методи, а також інструментальний засіб отримав впровадження у діяльності НВП «Каре» та знайшли відображення у навчальному процесі та науково-дослідницькій діяльності Національного університету «Одеська політехніка».

*Ключові слова:* комп'ютерні науки, машинне навчання, інтелектуальний аналіз даних, моделі та методи аналізу даних, аналіз складних структур даних, графові та структурні моделі, оцінювання невизначеності, прийняття рішень в умовах невизначеності.

### **Список публікацій здобувача за темою дисертації**

1) Годовиченко М.А.; Курінько Д.Д. «Аналіз існуючих підходів до автоматизації рефакторингу об'єктно-орієнтованих програмних систем» Publ. Nauka i Tekhnika. Odesa: Ukraine. Herald of Advanced Information Technology 8 (2), 179-196. <https://doi.org/10.15276/hait.08.2025.11>. (Index Scopus).

<https://hait.op.edu.ua/index.php/journal/article/view/187>

2) Курінько Д.Д. Модель оцінювання ефективності рефакторингу на основі статистичного контролю // Наука і техніка сьогодні. Серія «Техніка». – 2025. – Т. 50, № 9. – С. 1265–1280. – DOI: [https://doi.org/10.52058/2786-6025-2025-9\(50\)-1265-1280](https://doi.org/10.52058/2786-6025-2025-9(50)-1265-1280). (Index Copernicus)

<https://perspectives.pp.ua/index.php/nts/article/view/29453>

3) Курінько Д.Д.; Кривда В.І. «Рекомендація рефакторингів із багатоцільовою оптимізацією та урахуванням невизначеності для дублювання коду» Publ. Nauka i Tekhnika. Odesa: Ukraine. Herald of Advanced Information Technology 8 (3), 301–315. <https://doi.org/10.15276/hait.08.2025.19> (Index Scopus).

<https://hait.op.edu.ua/index.php/journal/article/view/209>

4) Курінько Д.Д. «Гібридні графи для запахів коду: багаторівнева модель виявлення антипатернів у програмних компонентах» Publ. Nauka i Tekhnika. Odesa: Ukraine. ААІТ 8 (3), 274–285. <https://doi.org/10.15276/aait.08.2025.18> (Index Copernicus).

<https://aait.op.edu.ua/index.php/journal/article/view/201>

5) Курінько Д.Д.; Кривда В.І. «Від комітів до причинності: побудова маловпливових послідовностей рефакторингів черезпричинно-наслідковий аналіз репозиторіїв»// Наука і техніка сьогодні. Серія «Техніка». – 2025. – Т. 52, № 11. – С. 1752–1773. – DOI: [https://doi.org/10.52058/2786-6025-2025-11\(52\)-1752-1773](https://doi.org/10.52058/2786-6025-2025-11(52)-1752-1773) . (Index Copernicus).

<https://perspectives.pp.ua/index.php/nts/article/view/32491>

6) Курінько Д.Д. Інтегрований підхід до виявлення рефакторингу в ООП-системах // Modern The Integration of Research, Innovation and Economy : Proceedings of the 1st International Scientific and Practical Conference. – Seville, Spain : International Scientific Unity, October 8–10, 2025. – P. 26–30.

<https://isu-conference.com/en/archive/the-integration-of-research-innovation-and-economy-08-10-25/>

7) Курінько Д.Д., Кривда В.І. Мультимодальні графові подання для надійного виявлення антипатернів в еволюційних кодових базах // Інформатика. Культура. Техніка. – 2025. – Т. 2, № 2. – С. 294–299. – DOI: <https://doi.org/10.15276/ict.02.2025.45> .

<https://ict.op.edu.ua/index.php/journal/article/view/56>

8) Курінько Д.Д. Проблеми виявлення потреби у рефакторингу в об'єктно-орієнтованому коді // Innovative Research in Science and Economy :

Proceedings of the 2nd International Scientific and Practical Conference. – Brussels, Belgium : International Scientific Unity, December 3–5, 2025. – P. 764–767.

*<https://isu-conference.com/en/archive/innovative-research-in-science-and-economy-03-12-25/>*

9) Курінько Д.Д. Маловпливові послідовності рефакторингів: причинно-орієнтований підхід до аналізу історій репозиторіїв // Abstracts of XV International Scientific and Practical Conference. – Sofia, Bulgaria. – P. 264–267.

*<https://eu-conf.com/en/events/the-impact-of-modern-digital-technologies-on-the-future-of-professions/>*

10) Курінько Д.Д. Селективне виявлення антипатернів з open-set-головками та оцінкою невизначеності // Proceedings of the 5th International Scientific and Practical Conference. – Liverpool, United Kingdom : Cognum Publishing House, 2025. – P. 249–252.

*<https://sci-conf.com.ua/v-mizhnarodna-naukovo-praktichna-konferentsiya-modern-science-trends-challenges-solutions-11-13-12-2025-liverpul-velikobritaniya-arhiv/>*

11) Курінько Д.Д. Інкрементальне оновлення Code Property Graph для прискорення виявлення антипатернів у CI/CD-конвеєрах // Abstracts of XVI International Scientific and Practical Conference. – Munich, Germany. – P. 283–287.

*<https://eu-conf.com/en/events/conceptual-framework-and-dynamics-of-the-development-of-science/>*

## ABSTRACT

*Kurinko D.D.* Machine Learning Models and Methods for Detecting and Eliminating Anti-patterns in Software Components. – Qualification scientific work in the form of manuscript.

Thesis for the PhD degree in specialty 122 Computer science. – Odessa Polytechnic National University, Ministry of Education and Science of Ukraine, Odesa, 2026.

The **introduction** highlights the relevance of solving problems related to the automated detection of the need for refactoring object-oriented software code in modern software systems. It highlights the shortcomings of existing models and methods for identifying refactoring opportunities, in particular approaches based on code "smells," software metrics, and machine learning methods that do not take into account the context of the project, the interdependence of features, and data noise. The object, subject, purpose, tasks, and methods of the study are defined; the scientific novelty and practical significance of the results obtained are presented; the personal contribution of the applicant is highlighted.

The **first chapter** of the dissertation provides a systematic review of the issues of identifying and eliminating anti-patterns in software components as a key factor in reducing internal quality and complicating the evolution of industrial software systems. It shows how the growth in the scale and complexity of software leads to the accumulation of design defects, and anti-patterns are associated with degradation of maintainability, complication of modifications, and increased development costs. On this basis, the task of automatic detection and elimination of anti-patterns is formulated, and the place of such solutions in the modern engineering process is outlined.

Next, a classification of anti-patterns is presented and it is shown that their manifestations are not limited to one type of signal: structural, semantic, and evolutionary (historical) features are generalized, and the multimodality of manifestations and the associated problem of uncertainty are emphasized. After that, traditional approaches (metrics, rules, detectors) are analyzed, and structural and text/LLM-oriented approaches

are considered separately, with a focus on their strengths and limitations from the point of view of practical application.

Modern approaches based on machine and deep learning were analyzed separately: classic models based on metrics and smell tags, graph and representation models of code representation, hybrid solutions and uncertainty-aware approaches, as well as areas where these approaches are used to support refactoring and working with code duplicates were described.

An analysis of works in the field of automatic anti-pattern removal showed that existing models tend to rely on a single type of feature and do not reflect the multidimensional nature of anti-patterns, which causes false positives or false negatives.

To overcome this shortcoming, a multi-level model based on a hybrid graph representation of code was proposed, combining structural, semantic, and evolutionary features in a single space.

Further analysis of existing anti-pattern detection systems showed that detection at the method/class level does not detect component/system anti-patterns and does not take into account the dynamics of quality between releases.

To address this shortcoming, a multi-level representation and analysis was proposed, taking into account the relationships between levels, as well as a method of composing "minimally invasive" refactoring sequences that considers the cause-and-effect relationships of changes.

An analysis of current approaches to identifying anti-patterns has shown that they are developing in isolation, which can lead to conflicting conclusions and recommendations. It has been shown that the use of hybrid representations that integrate features into a single vector is promising.

As such a solution, the paper proposed a unified hybrid representation as a common basis for both anti-pattern detection and refactoring plan optimization.

Further analysis of anti-pattern detection methods showed that their work is based on the assumption that all types of anti-patterns are known in advance, which does not correspond to modern software development practices. In addition, most methods do not

provide a calibrated assessment of prediction uncertainty, which increases the risk of aggressive or, conversely, overly cautious decisions.

To overcome these shortcomings, the paper proposes the introduction of an open-set/low-confidence mode in anti-pattern detection models and explicit modeling of uncertainty when recommending refactorings.

An analysis of the practice of implementing anti-pattern detection tools has shown that their results are poorly integrated into the work of teams and modern practices of continuous integration and delivery.

To overcome this shortcoming, the paper proposes a refactoring recommendation module, a method for composing sequences of changes, and a statistical process control model that allows for formalized evaluation of the effectiveness of interventions and reduces the risk of wrong decisions.

Based on the results of the analysis, the research goals and objectives were formulated.

The **second chapter** of the dissertation clarifies the task of automatic detection of anti-patterns, taking into account their multimodal nature (structure, semantics, metrics, evolution), multi-level context (method – component – project), and the need for an open-set/low-confidence mode with explicit consideration of uncertainty. The limitations of rules/metrics and single-modal approaches based on machine learning are justified, and requirements for a new model (portability, incrementality for continuous integration and delivery) are formulated.

A hybrid graph representation of code in the form of a Code Property Graph with additional semantic links and 4-channel node features (structural, semantic embeddings, metric, evolutionary) is proposed, including normalization at the project level and incremental updating.

A multi-level model has been developed: at the local level, a sequential encoder and a heterogeneous GNN over a CPG subgraph with span-to-node and fusion connections are combined; at the component level, local descriptors are aggregated; at the project level, a component interaction graph is constructed and message passing is applied, and the final representation is obtained through hierarchical attention.

For decision-making, multi-label classification with uncertainty estimation (entropy/energy, epistemic uncertainty) and a selective prediction/rejection mechanism are used; the corresponding loss functions are formalized. The application pipeline in industrial conditions (including for pull requests), interpretability tools, and experimental verification on a multilingual corpus with comparison to baseline and ablation variants are also described.

Thus, **the first point of scientific novelty** is formulated: for the first time, a multilevel model for detecting anti-patterns based on a hybrid graph is proposed, which unifies syntactic trees, control flow graphs, and dependency graphs into a code property graph, with additional semantic and evolutionary features. The proposed model allows detecting typical and atypical anti-patterns, working with multilingual software projects, and ensuring compatibility with continuous integration and delivery processes.

The **third section** of the dissertation proposes models and methods for recommending, planning, and process-based evaluation of refactorings to eliminate anti-patterns.

**For safe, transparent, automated elimination of code duplicates with a predictable impact on the process**, a method for recommending refactorings through multi-objective optimization with uncertainty assessment, specialized in eliminating code duplicates, was proposed.

The idea behind the method is that several possible refactorings are considered for each group of duplicates. Each option is evaluated simultaneously according to three objectives—benefit, labor intensity, and CI risk – and uncertainty is calculated along with the forecast. Next, the best compromise solutions are selected through multi-objective (Pareto) optimization, and in case of high uncertainty, the system switches to low-confidence/open-set mode and may refrain from making a recommendation.

The proposed method consists of the following steps:

- 1) clone detection – groups of code duplicates (clone classes) are formed in the project;
- 2) action generation – for each group, a set of candidate refactorings + the "No refactoring" option is specified.

- 3) hybrid representation – a CPG (AST/CFG/PDG) is constructed for clone fragments and the context of components is fixed;
- 4) feature calculation – structural, semantic, metric, and evolutionary features are calculated and aggregated at the clone group level;
- 5) effect prediction – for each "group–action" pair, the benefit, labor intensity, and CI risk are evaluated;
- 6) uncertainty – dispersion (entropy) and calibrated confidence of predictions are obtained through a probabilistic model;
- 7) low-confidence filter – candidates with high uncertainty are cut off or transferred to manual review (open-set);
- 8) Pareto selection – Pareto-optimal solutions are constructed among the rest based on (benefit, labor intensity, risk);
- 9) Ranking with penalty – the Pareto set is sorted taking into account the project policy and the penalty for uncertainty;
- 10) exit/hold – a prioritized list of recommendations or a "hold" decision is returned if there are no safe options.

Experimental research has shown that the proposed method demonstrated high quality of refactoring type selection (Accuracy=0.80 and F1macro=0.76), as well as high quality of prioritization (NDCG@5=0.76 and U@5=0.62), i.e., the method generates more useful top-k recommendations and supports "cautious" behavior by cutting off uncertain decisions.

Thus, the **second point of scientific novelty** is formulated: the refactoring recommendation method has been improved through multi-objective optimization with uncertainty estimation, specialized in eliminating code duplicates. The use of the method ensures safe, transparent, automated elimination of code duplicates with a predictable impact on the process.

**To maximize the expected benefits of a sequence of refactorings under the constraints of interface changes and the budget for continuous integration and delivery**, a method was proposed for composing "minimally invasive" sequences of refactorings based on causal inference from repository history.

The idea behind the method is that it builds a coordinated refactoring plan from a set of candidates in order to maximize the expected (conservative) quality gain while limiting invasiveness and risk: change budgets, compatibility of steps, impact on the API, and CI/CD risk are taken into account. The uncertainty of forecasts is used for "cautious" selection—preference is given to actions with a reliable effect, while dangerous/uncertain options are cut off or lead to refraining from a recommendation.

The proposed method consists of the following steps:

- 1) collecting refactoring candidates with their assessments and impact metadata;
- 2) describing the current state of components through quality metrics, anti-patterns, evolutionary and CI/CD indicators;
- 3) assessing the expected contribution to improvement, taking into account the context of the component;
- 4) correcting the effect for uncertainty (moving from a point estimate to a lower confidence limit);
- 5) formulation of a multi-objective planning task (maximizing utility while minimizing invasiveness/risk and adhering to budgets);
- 6) imposing compatibility and order constraints and correctness invariants;
- 7) searching for the best acceptable sequence through heuristic search with pruning branches by budget and risk.
- 8) forming a plan or refraining from making a recommendation if there is no safe solution within the constraints.

The results of the experimental study showed that the proposed method provided an average quality increase of 0.084, with an average invasiveness of 0.036 and an average change in the proportion of failed builds of  $-0.004$ .

Thus, **the third point of scientific novelty** is formulated: for the first time, a method for composing "minimally invasive" refactoring sequences based on causal inference from the repository history is proposed. Using this method allows maximizing the expected benefit under the constraints of interface changes and the budget for continuous integration and delivery.

**To reduce false alarms and ensure controlled feedback for continuous integration and delivery during refactoring**, a model for evaluating the effectiveness of refactoring based on statistical process control with attribution of impact has been proposed.

The proposed model considers refactoring not as a one-time "before/after" event, but as an intervention in the development process over time, so the effectiveness of changes is assessed based on statistical process control (SPC) on time series metrics.

Its key idea is to distinguish, in CI/DevOps conditions, the real, sustainable effect of refactoring from natural fluctuations, noise, and accompanying changes (features, tests, dependencies), and formally draw one of three conclusions: improvement, deterioration, or insufficient evidence with a controlled level of false signals.

In practice, this is implemented as dual-loop monitoring: code quality metrics and CI process metrics are tracked simultaneously, indicators are pre-unified according to the principle "more = better," normalized and aggregated robustly, and for series with dependencies, autocorrelation removal (prewhitening) is applied to reduce false alarms. Next, the moment of intervention is recorded, and in the post-intervention segment, SPC methods (EWMA/CUSUM and their generalizations) detect small but persistent shifts relative to the baseline mode with limits set for a given risk level; if necessary, the effect is additionally attributed taking into account covariates (e.g., churn/test changes), after which a final verdict and feedback are formed to adjust the policies for selecting "safe" refactorings in subsequent iterations.

An experimental study of the model showed that the most effective approach for detecting shifts in CI stability metrics is the CUSUM-based approach: compared to EWMA, the false alarm rate decreases by 16.7%, sensitivity increases by 9.7%, the probability of detection increases by 3.2%, and the average time to signal is reduced by 32.1%, meaning that unwanted shifts are detected significantly faster. At the same time, the EWMA variant with bootstrap limits demonstrates more conservative behavior—false alarms also decrease by 16.7%, but at the cost of a slight drop in sensitivity by 2.8% and detection probability by 2.1% and a slight increase in time to signal (+3.6%).

Thus, **the fourth point of scientific novelty** is formulated: the model for evaluating the effectiveness of refactorings based on statistical process control with attribution of influence has been improved. The use of the model provides evidence-based decisions of "improvement, deterioration, no changes detected" with a given level of confidence, reduces false alarms, and provides manageable feedback for continuous integration and delivery.

The **fourth section** of the dissertation examines an integrated tool.

An integrated tool for end-to-end pipeline management of anti-patterns has been implemented and described:

- construction of a hybrid code representation (CPG + semantic embeddings + VCS features);
- detection of anti-patterns;
- generation/multi-criteria ranking of refactorings (Pareto + lower confidence limits taking into account uncertainty);
- planning of a minimally invasive sequence with budgets/risk constraints;
- SPC effect evaluation (EWMA/CUSUM);
- formation of engineering artifacts for reproducibility and traceability, including a mechanism for retention at low confidence.

The experimental study was conducted on 7 Java projects (JUnit5, Commons Lang, Guava, Spring Boot, Elasticsearch, Hadoop, Jenkins) in fixed release slices with a controlled environment and comparison with baseline approaches. A generalized end-to-end effect was obtained: an average decrease in the Technical Debt Index (TDI) by 16.4%, an increase in the Maintainability Index (MI) by an average of +5.66, a decrease in Cognitive Complexity (CC) by 12.1%, and a decrease in Clone density by 32.0%. The consistency between the predicted and actual effects is high ( $R^2 \approx 0.91$ ; "Expected vs Actual" discrepancies range from approximately  $-1.6$  to  $+0.3\%$ ).

Compared to basic tools, the integrated system provides a greater cumulative effect (for classic analyzers, the improvements are close to zero, for partial recommenders, they are significantly lower than for integrated ones) and also provides a complete cycle of "detection – refactoring plan – confirmation of effect."

Thus, the obtained results confirm the correctness and effectiveness of the technical solutions proposed in the dissertation.

The models and methods developed in the work, as well as the tool, have been implemented in the activities of NVP "Kare" and are reflected in the educational process and research activities of the National University "Odessa Polytechnic".

Keywords: computer science, machine learning, intelligent data analysis, models and methods of data analysis, analysis of complex data structures, graph-based and structural models, uncertainty estimation, decision-making under uncertainty.

### **List of publications of the applicant on the topic of the dissertation**

1) Hodovychenko M. A., Kurinko D. D. Analysis of Existing Approaches to Automated Refactoring of Object-Oriented Software Systems // Herald of Advanced Information Technology. – Odesa : Nauka i Tekhnika, 2025. – Vol. 8, № 2. – P. 179–196. – DOI: <https://doi.org/10.15276/hait.08.2025.11>. (Index Scopus).

<https://hait.op.edu.ua/index.php/journal/article/view/187>

2) Kurinko D. D. A Model for Evaluating Refactoring Effectiveness Based on Statistical Process Control // Science and Technology Today. Series «Engineering». – 2025. – T. 50, № 9. – C. 1265–1280. – DOI: [https://doi.org/10.52058/2786-6025-2025-9\(50\)-1265-1280](https://doi.org/10.52058/2786-6025-2025-9(50)-1265-1280). (Index Copernicus).

<https://perspectives.pp.ua/index.php/nts/article/view/29453>

3) Kurinko D. D., Kryvda V. I. Uncertainty-Aware Multi-Objective Refactoring for Code Duplication // Herald of Advanced Information Technology. – Odesa : Nauka i Tekhnika, 2025. – Vol. 8, № 3. – P. 301–315. – DOI: <https://doi.org/10.15276/hait.08.2025.19>. (Index Scopus).

<https://hait.op.edu.ua/index.php/journal/article/view/209>

4) Kurinko D. D. Hybrid Graphs for Code Smells: A Multi-Level Model for Anti-Pattern Detection in Software Components // Applied Aspects of Information Technology. – 2025. – Vol. 8, № 3. – P. 274–285. – DOI: <https://doi.org/10.15276/aait.08.2025.18>. (Index Copernicus).

<https://aait.op.edu.ua/index.php/journal/article/view/201>

5) Kurinko D. D., Kryvda V. I. From Commits to Causality: Constructing Low-Impact Refactoring Sequences via Repository-Based Causal Inference // Science and Technology Today. Series «Engineering». – 2025. – T. 52, № 11. – C. 1752–1773. – DOI: [https://doi.org/10.52058/2786-6025-2025-11\(52\)-1752-1773](https://doi.org/10.52058/2786-6025-2025-11(52)-1752-1773). (Index Copernicus).

<https://perspectives.pp.ua/index.php/nts/article/view/32491>

6) Kurinko D. D. An Integrated Approach to Detecting Refactoring Needs in OOP Systems // Modern The Integration of Research, Innovation and Economy : Proceedings of the 1st International Scientific and Practical Conference. – Seville, Spain : International Scientific Unity, October 8–10, 2025. – P. 26–30.

<https://isu-conference.com/en/archive/the-integration-of-research-innovation-and-economy-08-10-25/>

7) Kurinko D. D., Kryvda V. I. Multimodal Graph Representations for Reliable Anti-Pattern Detection in Evolving Codebases // Informatics. Culture. Technology. – 2025. – T. 2, № 2. – C. 294–299. – DOI: <https://doi.org/10.15276/ict.02.2025.45>.

<https://ict.op.edu.ua/index.php/journal/article/view/56>

8) Kurinko D. D. Challenges of Detecting the Need for Refactoring in Object-Oriented Code // Innovative Research in Science and Economy : Proceedings of the 2nd International Scientific and Practical Conference. – Brussels, Belgium : International Scientific Unity, December 3–5, 2025. – P. 764–767.

<https://isu-conference.com/en/archive/innovative-research-in-science-and-economy-03-12-25/>

9) Kurinko D. D. Low-Impact Refactoring Sequences: A Causality-Oriented Approach to Repository History Analysis // Abstracts of XV International Scientific and Practical Conference. – Sofia, Bulgaria. – P. 264–267.

<https://eu-conf.com/en/events/the-impact-of-modern-digital-technologies-on-the-future-of-professions/>

10) Kurinko D. D. Selective Anti-Pattern Detection with Open-Set Heads and Uncertainty Estimation // Proceedings of the 5th International Scientific and Practical

Conference. – Liverpool, United Kingdom : Cognum Publishing House, 2025. – P. 249–252.

*<https://sci-conf.com.ua/v-mizhnarodna-naukovo-praktichna-konferentsiya-modern-science-trends-challenges-solutions-11-13-12-2025-liverpul-velikobritaniya-arhiv/>*

11) Kurinko D. D. Incremental Updating of the Code Property Graph to Accelerate Anti-Pattern Detection in CI/CD Pipelines // Abstracts of XVI International Scientific and Practical Conference. – Munich, Germany. – P. 283–287.

*<https://eu-conf.com/en/events/conceptual-framework-and-dynamics-of-the-development-of-science/>*