

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
імені ІГОРЯ СІКОРСЬКОГО**

**Факультет лінгвістики
Кафедра теорії, практики та перекладу англійської мови**

«На правах рукопису»
УДК 811.111'25'373.46:004](045)

«До захисту допущено»

Завідувач кафедри
_____ Ініціали, прізвище
«__» _____ 2025 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

**на здобуття ступеня магістра
зі спеціальності 035 «Філологія»
на тему: «Особливості відтворення англomовної термінології комп'ютерної
документації в українськомовному перекладі»**

Виконав: студент 2 курсу магістратури, групи ЛА-41мп
Євдокименко Ілля Олександрович _____

Науковий керівник:
проф. каф. ТППАМ, д. пед. н., доц. Деркач І.В. _____

Рецензент:
проф. каф. ТППАМ, д. філол. н., проф. Іваненко С.М. _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з
праць інших авторів без
відповідних посилань
Студент Євдокименко Ілля
Олександрович

Київ 2025

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет лінгвістики
Кафедра теорії, практики та перекладу англійської мови

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 035 Філологія (035.041 Германські мови та літератури (переклад включно), перша – англійська)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ М. П. Колісник

«__» _____ 2025 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Євдокименко Іллі Олександровичу

1. Тема дисертації «Особливості відтворення англомовної термінології комп'ютерної документації в українськомовному перекладі», науковий керівник дисертації: Деркач І.В, доц. каф. теорії, практики та перекладу англійської мови, затверджена наказом по університету від:
2. Термін подання студентом дисертації:
3. Об'єкт дослідження: терміни, що функціонують у текстах комп'ютерної документації.
4. Предмет дослідження: структурно-семантичні особливості термінів англомовної комп'ютерної документації та способи їх перекладу українською мовою.
5. Перелік завдань, які потрібно розробити:
 - Здійснити теоретичний огляд поняття терміну, його ознак, функцій та класифікації.

- Визначити структурно-семантичні особливості термінів комп'ютерної документації.
 - Проаналізувати основні стратегії та методи перекладу комп'ютерних термінів з англійської мови на українську (зокрема, калькування, адаптації та транслітерації).
 - Зібрати термінологічний матеріал з англійської комп'ютерної документації (інструкцій, глосаріїв, довідників) та охарактеризувати його.
 - Здійснити практичний перекладацький аналіз зібраної термінології та виявити типові труднощі перекладу і помилки (наприклад, некоректне калькування та мовні інтерференції).
 - Запропонувати підходи та рекомендації до стандартизації перекладу окремих груп термінів з урахуванням національних та міжнародних стандартів.
 - Узагальнити результати дослідження та сформулювати загальні висновки щодо особливостей відтворення англійської термінології в українському перекладі.
6. Орієнтовний перелік публікацій: Стаття у фаховому науковому журналі.
7. Дата видачі завдання: 16 грудня 2025.

Календарний план

№ з/п	Назва етапу виконання магістерської дисертації	Термін виконання	Примітка
1	Створення індивідуального плану написання МД	до 11.11.2024	вик.
2	Створення списку джерел та опрацювання робіт, пов'язаних з темою МД	до 31.12.2024	вик.
3	Підготовка та затвердження структури (змісту) всіх розділів МД	до 29.02.2025	вик

4	Написання чернетки Вступу та ознайомлення з літературою для першого розділу	до 31.05.2025	вик.
5	Написання першого розділу МД та подання його науковому керівнику	до 31.08.2025	вик.
6	Написання другого розділу МД	до 31.09.2025	вик.
7	Написання третього аналітичного розділу МД	до 12.10.2025	вик
8	Написання четвертого практичного розділу МД та подання його науковому керівнику	до 25.10.2025	вик
9	Написання статті до фахового журналу за матеріалами першого розділу МД	до 30.10.2025	вик.
10	Підготовка матеріалів для доповіді на науковій конференції	До 5.11.2025	вик
11	Оформлення Списку використаних джерел відповідно до вимог	До 10.11.2025	вик
12	Оформлення результатів дослідження у формі кінцевого варіанту МД, остаточна перевірка науковим керівником	До 30.11.2025	вик.
13	Подання тексту МД на перевірку наявності текстових збігів	до 04.12.2025	вик.
14	Відгук наукового керівника та подання МД на рецензію визначеним рецензентам	до 28.11.2025	вик.
15	Розробка презентації до захисту та підготовка відповідей на можливі запитання	До 10.12.2025	вик
16	Розміщення електронної версії МД, тез та опублікованої статті на сайті кафедри та подання роздрукованої МД на кафедру	До 14.12.2025	вик.
17	Державна атестація. Захист МД	16 – 19.12.2025 (за графіком захистів)	вик.

Студент

І.О. Євдокименко

Науковий керівник дисертації

І.В. Деркач

РЕФЕРАТ

Євдокименко І. О. Особливості відтворення англомовної термінології комп'ютерної документації в українськомовному перекладі. Кваліфікаційна робота у вигляді рукопису. Дисертація на здобуття ступеня магістра за спеціальністю 035 «Філологія». КПІ ім. Ігоря Сікорського, Київ, 2025.

У кваліфікаційній роботі розглянуто лінгвістичні та перекладознавчі особливості англомовної комп'ютерної термінології, здійснено теоретико-методологічне обґрунтування специфіки її перекладу українською мовою, з'ясовано структурні типи термінів, механізми їх розвитку та прагматичні характеристики у технічних текстах. Проаналізовано стратегії калькування, адаптації, транслітерації та описового перекладу, обґрунтовано фактори їх вибору. Виявлено типові труднощі перекодування ІТ-лексики, встановлено закономірності варіативності та синонімії, досліджено вплив національних і міжнародних стандартів на процес унормування термінології.

Наукова новизна роботи полягає у систематизації особливостей перекладу англомовної комп'ютерної термінології саме в українському контексті та у поєднанні теоретичного аналізу з практичним корпусним дослідженням текстів користувацької та технічної документації. Уточнено роль перекладача як суб'єкта термінотворення та запропоновано комплекс рекомендацій для підвищення якості локалізації.

Практичне значення здобутих результатів полягає в можливості використання їх у професійній локалізації програмного забезпечення, укладанні словників і глосаріїв, викладанні теорії та практики перекладу, а також у вдосконаленні стандартів української ІТ-термінології. Отримані результати сприяють підвищенню точності та зрозумілості інтерфейсів, покращенню користувацького досвіду і зростанню довіри до україномовних цифрових продуктів.

Ключові слова: комп'ютерна термінологія, технічна документація, локалізація, калькування, адаптація, термінотворення, стандартизація.

ABSTRACT

Yevdokymenko I. O. Specific Features of Rendering English-Language Computer Documentation Terminology in Ukrainian Translation. Master's thesis (manuscript). Dissertation for obtaining the Master's degree in Philology, speciality 035 "Philology". Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2025.

The thesis examines the linguistic and translation-specific features of English computer terminology and provides a theoretical and methodological justification for its rendering into Ukrainian. It identifies structural types of terms, mechanisms of terminological development, and pragmatic characteristics of IT vocabulary in technical documentation. The study analyses the strategies of calquing, adaptation, transliteration, and descriptive translation, substantiating the factors that determine their use. Typical translation difficulties are identified, including semantic distortions and interference; terminological variation and synonymy are systematised; and the role of national and international standards in terminology harmonisation is assessed.

The scientific novelty of the research lies in the systematic analysis of English computer terminology translation within the Ukrainian linguistic context, as well as in combining theoretical insights with a practical corpus-based investigation of user-interface and technical documentation. The study clarifies the role of the translator as an active agent of Ukrainian term creation and offers a set of practical recommendations aimed at improving localisation quality.

The practical value of the results is reflected in their applicability to software localisation, dictionary and glossary development, translator training, and the refinement of Ukrainian IT terminology standards. The findings contribute to enhancing translation accuracy, improving user-interface clarity, and strengthening user trust in Ukrainian-language digital products.

Keywords: computer terminology, technical documentation, localisation, calquing, adaptation, term creation, standardisation.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 КОМП'ЮТЕРНА ТЕРМІНОЛОГІЯ ЯК ЛІНГВІСТИЧНЕ І ГАЛУЗЕВЕ ЯВИЩЕ	13
1.1 Формування та еволюція комп'ютерної терміносистеми	13
1.2 Типологія комп'ютерних термінів за структурою та значенням	17
1.3 Динаміка новотворів і роль неологізмів у ІТ-лексиконі	20
1.4 Стилiстичні та функціональні особливості термінів у текстах технічної документації	23
Висновки до першого розділу	26
РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ПЕРЕКЛАДУ ГАЛУЗЕВОЇ ТЕРМІНОЛОГІЇ	27
2.1 Переклад спеціальної лексики у світлі сучасного перекладознавства	27
2.2 Класифікація стратегій відтворення англомовної термінології	31
2.3 Фактори вибору перекладацьких стратегій калькування, адаптації та транслітерації	35
Висновки до другого розділу	37
РОЗДІЛ 3 УНОРМУВАННЯ ТЕРМІНОЛОГІЇ В УКРАЇНСЬКОМОВНОМУ ПЕРЕКЛАДІ КОМП'ЮТЕРНИХ ТЕКСТІВ	38
3.1 Термінологічна варіативність і проблема синонімії	38
3.2 Роль перекладача у творенні нових українських відповідників	40
3.3 Національні та міжнародні стандарти в сфері ІТ-термінології	41
Висновки до третього розділу	44
РОЗДІЛ 4 ПРАКТИЧНИЙ АНАЛІЗ ВІДТВОРЕННЯ АНГЛОМОВНОЇ ТЕРМІНОЛОГІЇ В УКРАЇНСЬКОМУ ПЕРЕКЛАДІ	45
4.1 Характеристика матеріалу дослідження та методика його аналізу	45
4.2 Особливості перекладу термінів користувацького інтерфейсу	51
4.3 Способи передачі багатокomпонентних термінів і абревіатур	58
4.4 Виявлені труднощі перекладу та типові помилки	63
4.4.1 Некоректне калькування і спотворення сенсу	63

4.4.2 Мовні інтерференції і невідповідність нормам	69
4.5 Статистичний аналіз перекладацьких рішень і пропозиції щодо покращення	75
Висновки до четвертого розділу	84
ЗАГАЛЬНІ ВИСНОВКИ	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	91
ДОДАТКИ	99

ВСТУП

Динамічний розвиток інформаційних технологій сприяє активній адаптації сучасної комп'ютерної документації. В умовах цифровізації значна частина програмного забезпечення, веб сервісів і апаратних засобів представлена англomовними інструкціями, довідниками, технічними описами та інтерфейсами. Це зумовлює потребу у високоякісному перекладі, що гарантуватиме точне й однозначне сприйняття матеріалу україномовними користувачами. Відтак стає актуальним ґрунтовне дослідження термінології, що формується у межах комп'ютерного дискурсу.

Англійська мова виступає головним джерелом фахової термінології в галузі ІТ, зокрема у сфері документації. Більшість термінів, що функціонують в українському мовному середовищі, є запозиченнями або кальками з англійської мови, що нерідко призводить до лексичної варіативності та перекладацької неоднозначності. Відсутність усталених відповідників або офіційних стандартів перекладу лише поглиблює цю проблему.

Актуальність цього питання особливо зростає у зв'язку з необхідністю аналізу термінології, що зустрічається в англomовній комп'ютерній документації, та пошуку оптимальних способів її передавання українською мовою. Враховуючи підвищений попит на україномовні ІТ-матеріали, дослідження термінів і стратегій їх перекладу сприяє уніфікації професійної лексики та покращенню якості технічних перекладів.

Ступінь розробленості проблеми у науковій літературі. Дослідження перекладу комп'ютерної термінології знайшло своє відображення в наукових працях низки дослідників. Зокрема, Е. С. Черкасова та А. В. Сергієва аналізували особливості перекладу комп'ютерної лексики, акцентуючи увагу на таких стратегічних підходах, як калькування, транскодування та описовий переклад. Гліб Данилов, Вікторія Балакірєва та Катерина Василенко вивчали вплив автоматизованих перекладачів на якість технічного тексту, тоді як Н. П. Франчук зосереджувала увагу на перспективності машинного перекладу у контексті

розвитку штучного інтелекту. Попри значний внесок згаданих науковців, низка аспектів, зокрема узгодження перекладацьких рішень із чинними українськими стандартами та адаптація термінології до специфіки цифрового середовища, потребує подальшого глибокого аналізу.

Об'єктом дослідження є терміни, що функціонують у текстах комп'ютерної документації.

Предметом дослідження виступають структурно-семантичні особливості термінів англomовної комп'ютерної документації та способи їх перекладу українською мовою.

Мета дослідження — проаналізувати особливості вживання, структуру, семантику та переклад англomовних термінів, що зустрічаються в комп'ютерній документації, з метою виявлення тенденцій, проблемних моментів та оптимальних шляхів перекладу.

Завдання дослідження:

1. Здійснити теоретичний огляд поняття терміна, його ознак, функцій та класифікації.
2. Визначити структурно-семантичні особливості термінів комп'ютерної документації.
3. Проаналізувати основні стратегії та методи перекладу комп'ютерних термінів з англійської мови на українську.
4. Зібрати термінологічний матеріал з англomовної комп'ютерної документації (інструкцій, глосаріїв, довідників).
5. Здійснити перекладацький аналіз зібраної термінології та виявити типові труднощі перекладу.
6. Запропонувати підходи до стандартизації перекладу окремих груп термінів.

Методи дослідження: у роботі використано метод суцільної вибірки (при зборі термінів з документації), компонентний аналіз (для дослідження структури термінів), зіставний аналіз (для порівняння англomовних термінів та їх

українських відповідників), а також метод індукції — для узагальнення отриманих результатів.

Матеріалом дослідження стали понад 300 термінологічних одиниць, відібраних з англomовної документації до програмного забезпечення, зокрема Microsoft Docs, Oracle Documentation, GitHub Docs, Ubuntu Manual, Adobe Help Center, Mozilla Developer Network.

Наукова новизна даної роботи полягає в комплексному міждисциплінарному підході до аналізу англomовної комп'ютерної термінології, що функціонує в технічній документації, з погляду перекладознавства, термінознавства та когнітивної лінгвістики. Дослідження вперше системно виокремлює структурні типи термінів (прості, складні, словосполучення, аббревіатури, словотвірні неологізми) у межах саме документаційного контексту, що раніше розглядалося в загальному ключі ІТ-дискурсу.

Дослідження зосереджене на визначенні закономірностей перекладу термінів, враховуючи їх морфологічну структуру, частотність використання та прагматичну роль у тексті. Також уперше проведено аналіз трансформацій, які відбуваються з терміном у процесі перекладу: адаптація до мовних норм цільової мови, спрощення, узагальнення, конкретизація або транслітерація без змістових змін. Виявлено, що технічний переклад нерідко застосовує стратегії, які суперечать загальноприйнятим нормам засвоєння термінів в українській мові, наприклад, збереження неадаптованої англomовної форми, що формально не має статусу запозичення.

Окремий аспект новизни полягає у дослідженні перекладацьких рішень у зіставленні з чинними національними стандартами (зокрема, ДСТУ 3966:2000 «Інформаційні технології. Терміни та визначення») та реальним ужитком у сучасному україномовному ІТ-дискурсі. Уперше простежено ступінь відповідності між стандартизованою термінологією та тією, що фактично закріпилася у цифровому середовищі — у програмних інтерфейсах, локалізаціях, на форумах тощо.

Практичне значення роботи полягає в її застосуванні у кількох ключових сферах. Запропоновані стратегії перекладу допомагають уникнути термінологічної плутанини, що виникає через відсутність єдиної практики перекладу комп'ютерної лексики. Матеріали дослідження можуть слугувати довідковою базою для перекладачів-початківців і фахівців у сфері локалізації програмного забезпечення.

Визначені терміни, способи їх перекладу та запропоновані відповідники можуть стати основою для створення стандартизованих словників у технічній документації, що сприятиме уніфікації перекладацької практики. Окрім того, матеріал дослідження може бути інтегрований у курси з технічного перекладу, термінознавства, ІТ-лінгвістики та комп'ютерної стилістики. Практичні приклади допомагають студентам та аспірантам краще орієнтуватися в реальних викликах перекладу сучасної ІТ-документації.

На основі роботи можна розробити рекомендації для перекладу інтерфейсів, діалогових вікон, довідкових розділів та навігаційних меню. Урахування функціонального навантаження терміну дозволяє краще адаптувати матеріал до потреб користувача. Запропоновані принципи уніфікації й варіативності можуть також слугувати орієнтиром для редакторів і коректорів технічних текстів під час оцінки термінологічної послідовності та лексичної узгодженості у великомасштабних документах.

Апробація результатів дослідження. Результати дослідження пройшли апробацію в академічному та професійному середовищі шляхом участі у фахових семінарах, круглих столах і методологічних обговореннях на кафедрі перекладознавства та прикладної лінгвістики.

Ключові слова: комп'ютерна термінологія, термін, переклад, структурні особливості, семантика, англomовна документація, технічний переклад.

Структура роботи: дослідження складається зі вступу, трьох теоретичних розділів, одного практичного розділу, висновків, списку використаних джерел і додатків.

РОЗДІЛ 1 КОМП'ЮТЕРНА ТЕРМІНОЛОГІЯ ЯК ЛІНГВІСТИЧНЕ І ГАЛУЗЕВЕ ЯВИЩЕ

1.1 Формування та еволюція комп'ютерної терміносистеми

Комп'ютерна документація не просто передає інформацію з однієї мови в іншу — її переклад виходить далеко за межі суто технічної трансляції. У сфері ІТ, де навіть мінімальне термінологічне відхилення може викликати хибне розуміння коду, команд або принципів роботи системи, кожне слово має критичне значення. Технічні тексти цієї галузі — це не просто інструкції чи описи, а частина когнітивної інфраструктури, що базується на специфічній мові мислення, сформованій у межах англomовного технокультурного простору. Відтак, перекладач у цьому випадку виступає не як механічний ретранслятор, а як посередник між технічною культурою джерела та мовною традицією цільового середовища.

Комп'ютерна термінологія, яка складає основу документації, часто є вектором не лише лексичного значення, а й концептуальної моделі технологічного мислення. Наприклад, англійський термін *thread* має однозначне технічне значення в контексті багато потокового програмування, яке не може бути буквально передано як «нитка». Українське «потік» зберігає функціональну точність, однак вимагає контекстуалізації — розуміння, що мається на увазі саме *execution thread*, тобто одиниця виконання в рамках процесу. Подібні тонкощі свідчать, що перекладач повинен не лише володіти лексиконом, а й мислити категоріями цільової професійної спільноти, для якої призначено текст.

Ця складність зумовлює потребу в інтеграції фахових знань у сфері лінгвістики, термінознавства та інформаційних технологій. Ідея «перекладати точно» не може реалізовуватись ізольовано від аналізу терміносистеми — тобто упорядкованого й ієрархічного набору термінів, які відображають зв'язки між поняттями певної галузі. Без розуміння логіки цієї системи — наприклад, що

cache, memory, RAM, virtual memory належать до однієї понятійної площини, — переклад ризикує стати або поверхневим, або незрозумілим [3, с. 358].

Комп'ютерна терміносистема — це більше, ніж набір слів для позначення ІТ-реалій. Вона являє собою структурований комплекс взаємопов'язаних понять, які мають як логічні, так і функціональні зв'язки. У технічній документації її значення особливо вагоме: вона забезпечує точність інтерпретації, створює єдине інформаційне середовище та виступає ключовим засобом професійної комунікації. Наприклад, термін *firewall* не можна перекласти як «вогняна стіна», попри буквальну відповідність, оскільки в цьому контексті він означає засіб захисту мережі. Такі приклади демонструють важливість коректного відтворення термінів у межах всієї системи понять.

Особливість комп'ютерної термінології полягає в її технічному характері, високій динаміці розвитку та тісному зв'язку з англomовним середовищем. З одного боку, англійська мова виконує функцію своєрідного «донорського» коду для терміносистем більшості мов світу. З іншого — у перекладі важливо зберегти функціональну точність, навіть коли йдеться про адаптацію або трансформацію терміну. Наприклад, український варіант *графічний інтерфейс користувача* (GUI) — результат граматичного та семантичного узгодження англійської аббревіатури з нормами української мови. Такий підхід поєднує стандартизацію з доступністю для носія мови.

Розвиток комп'ютерної термінології напряду відображає етапи становлення обчислювальної техніки. У 1940–50-х роках перші терміни виникали у зв'язку з обслуговуванням громіздких машин на кшталт ENIAC чи UNIVAC. Тоді з'явилися *mainframe, punch card, compiler, algorithm* — терміни, які окреслювали базові операції обробки даних. З розвитком персональних комп'ютерів у 1980-х лексика зміщується в бік користувацького досвіду: *desktop, mouse, interface, window*. У 2000-х — новий виток: поява інтернету та мобільних пристроїв приносить нові концепти: *browser, cloud, streaming, app* [5, с. 123].

Поширення нової термінології відбувалося через канали професійної комунікації: конференції ACM, SIGCHI, SIGGRAPH; технічні документи IEEE,

W3C; мануали Microsoft та Oracle. Саме ці джерела були першими носіями термінів, які згодом потрапляли до перекладених інструкцій, посібників, а зрештою — в український лексикон. Часто саме технічна документація — білінгвальна або оригінальна — ставала джерелом для калькування або транслітерації, зокрема таких термінів, як *debug*, *rollback*, *hotfix*, які не мали відповідників у ранніх українських словниках.

Формування української терміносистеми в галузі ІТ має декілька джерел, кожне з яких накладає специфічні обмеження та створює власні тенденції. Найпоширенішим і водночас найпроблемнішим шляхом є пряме запозичення з англійської. У випадках, коли в українській мові не існувало аналогів, англословні терміни починали функціонувати без змін або з мінімальною адаптацією. Так сталося з «сервер», «хеш», «біт», «бекенд» — слова, що були повністю або частково транскодовані з англійської, часто всупереч нормам українського словотвору. Такі запозичення забезпечують оперативність, однак не завжди зрозумілі широкому загалу.

Другий механізм — адаптація, коли запозичене поняття отримує нову мовну форму з урахуванням української лексичної та граматичної системи. Наприклад, *upload* часто перекладають як «завантажити (на сервер)», хоча логічно це протилежність до «завантажити (з сайту)», тобто *download*. Подібна варіативність спричиняє семантичні непорозуміння, особливо у неспеціалізованому середовищі. А *driver* — багатозначний термін: у побуті це водій, а в ІТ — програмний модуль для керування пристроєм. Тут вдаються до транскрипції — «драйвер» — щоби уникнути омонімії [4, с. 71].

Третій канал — внутрішнє термінотворення: тут спостерігається широка палітра прийомів — від калькування (*жорсткий диск* від *hard disk*, *відкотити зміни* від *rollback*) до неологізмів, як-от «знебезпечення» замість *debugging*. Хоч останній варіант і викликає дискусії щодо доцільності, він демонструє здатність мови творити власні терміни з опорою на внутрішні ресурси. Такі неологізми часто мають вищий поріг сприйняття, але водночас укріплюють мовну самобутність (Таб. 1.1).

Таблиця 1.1. Приклади способів поповнення української комп'ютерної терміносистеми.

Спосіб	Англійський термін	Український варіант	Коментар щодо адаптації
Пряме запозичення	server	сервер	Широко вживане без адаптації; закріпилося в ІТ-середовищі
Транслітерація	backend	бекенд	Часто використовується в розробці; суперечить традиційній морфології
Калькування	hard disk	жорсткий диск	Відтворює структуру вихідного терміну, зрозумілий широкий аудиторії
Неологізм	debugging	знебезпечення	Пропонований термін, що викликає фахові суперечки
Адаптація	upload	завантажити (на сервер)	Варіативність перекладу вимагає уточнення контексту
Семантичне розширення	thread	потік	Неочевидне значення для неспеціаліста, вимагає пояснення
Аббревіатурна форма	GUI (Graphical User Interface)	ГІК (графічний інтерфейс користувача)	Приклад спроби локалізованого скорочення

На тлі цього терміносистема постійно змінюється під тиском зовнішніх факторів. Технологічний прогрес призводить до появи нових понять, які швидко входять у вжиток: *мета-всесвіт*, *квантовий комп'ютер*, *генеративний ШІ*. Глобалізація ж забезпечує домінування англійської як мови-джерела, з якої черпають терміни всі інші. Утім, поряд із цим активно розвиваються і стандартизуючі ініціативи — від міжнародних (ISO/IEC, IETF) до національних (НТЦ «Стандарт»), а також корпоративні глосарії, як-от у Google або Microsoft. Ці ресурси сприяють єдності термінологічної системи, проте не завжди враховують мовні та культурні особливості цільової аудиторії [2, с. 53].

Сучасна українська комп'ютерна терміносистема знаходиться в постійному русі. Вона водночас переживає період формування і стандартизації, активно реагує на нові мовні виклики й технологічні тренди. Дедалі частіше виникає потреба у спільній роботі перекладачів, фахівців ІТ-сфери та мовознавців для вироблення узгоджених підходів до перекладу. Адже кожне термінологічне рішення впливає не лише на якість тексту, а й на сприйняття технологій загалом — як у професійному середовищі, так і серед ширшого кола користувачів.

1.2 Типологія комп'ютерних термінів за структурою та значенням

Комп'ютерна термінологія формує цілу систему назв, що відображають сучасні технології, процеси, концепції й об'єкти цифрової сфери. Її багатошаровість і динамічність зумовлюють потребу у чіткому типологічному аналізі, особливо в контексті перекладу на українську мову. Один із ключових аспектів — класифікація термінів за структурою, походженням і значенням, адже саме ці параметри впливають на вибір перекладацької стратегії.

За структурною будовою терміни в комп'ютерній сфері поділяються на кілька типів. До простих термінів належать однослівні одиниці, які найчастіше вживаються як базові поняття: сервер, файл, диск. Вони, як правило, легко адаптуються в українському мовному середовищі і не створюють значних труднощів у перекладі. Водночас, коли йдеться про складні терміни, ситуація ускладнюється. Наприклад, операційна система чи хмарні обчислення — це словосполучення, в яких кожен компонент несе семантичне навантаження. Їх переклад потребує збереження смислової цілісності, точного передавання значення та іноді — адаптації до контексту цільової мови.

Окрему групу становлять аббревіатури й акроніми, як-от RAM (random access memory), CPU (central processing unit), GUI (graphical user interface). З огляду на те, що більшість з них уже міцно закріплені у міжнародній терміносистемі, в українському перекладі зазвичай зберігається оригінальна аббревіатура з додаванням розшифрування у дужках при першому згадуванні. Така стратегія забезпечує і точність, і зрозумілість для читача. Водночас, існують випадки, коли акроніми піддаються українізації у вимові чи написанні, особливо в неофіційних або побутових контекстах (джі-пі-ес, ю-ес-бі) [7, с. 118].

Наступною категорією є словосполучення й терміносполуки, які найчастіше відображають точні поняття або технологічні рішення. Наприклад, нейронна мережа або графічний процесор — це сталі поєднання, які водночас потребують точності й лаконічності. Особливої уваги заслуговують терміни, у яких можлива

варіативність у словниковому складі, наприклад: буфер обміну і буфер пам'яті — вони мають різні функції, проте часто сприймаються некоректно як синоніми.

Походження термінів теж впливає на їхнє сприйняття і переклад. Значна частина англomовної термінології увійшла до української мови у формі прямих запозичень — це слова, які не зазнають морфологічної або фонетичної адаптації: комп'ютер, драйвер, ноутбук. У той самий час кальковані терміни — результат дослівного перекладу складових: операційна система (operating system), штучний інтелект (artificial intelligence), інформаційне середовище (information environment). Їх використання потребує контекстної чутливості, оскільки надмірна калька може призводити до громіздкості або семантичної плутанини. У деяких випадках терміни створюються в межах української лексичної системи — це власні терміни, як-от обчислювальна машина або електронна таблиця. Вони можуть бути частиною стратегій локалізації або результатом цілеспрямованої мовної політики.

За семантичним наповненням комп'ютерні терміни також класифікуються за функцією. Функціональні терміни мають конкретне, часто технічне призначення: компілятор, сервер, кеш-пам'ять. Вони позначають об'єкти, пристрої або програми, які виконують певні дії у цифровому середовищі. Натомість концептуальні терміни охоплюють абстрактні ідеї: інформаційне поле, штучний інтелект, архітектура системи. Їх переклад вимагає не лише термінологічної точності, а й контекстуального узгодження з дискурсом. Існують також метатерміни — це терміни, що описують саму систему знань: глосарій, термінологія, структура даних. Їх застосування зазвичай притаманне теоретичним текстам, методологічним настановам або довідковим матеріалам [1, с. 72].

У межах цієї системи важливо враховувати ще одне явище — термінологічну варіативність. Через відсутність єдиного державного чи міжнародного стандарту перекладу значна кількість термінів має кілька варіантів перекладу або використовується паралельно з оригінальним варіантом. Наприклад, інтегроване середовище розробки та IDE, або мережевий адаптер і network adapter — обидві форми зустрічаються навіть у фахових документах. Таке

співіснування часто спричинене впливом англомовної документації, яка використовується в оригіналі, а також локалізаційними стратегіями, що орієнтуються на ринок, а не на академічну норму.

Сучасна динаміка розвитку технологій спричиняє постійне оновлення термінології. Особливо активно формуються нові слова за допомогою аббревіації (IDE, API, HTML), а також шляхом метафоризації або запозичення зі суміжних галузей. Наприклад, блокчейн (blockchain) — новий термін, що швидко вкоренився у технічній лексиці. Йому властиве семантичне навантаження, яке важко передати іншими словами без втрати змісту, що часто змушує перекладачів залишати слово без змін або додавати пояснення [6, с. 62]. Водночас технологічні зміни сприяють адаптації вже наявних слів — поняття віртуальна машина або штучна нейронна мережа розширюють семантичне поле традиційних понять (Таб. 1.2).

Таблиця 1.2. Типологія комп'ютерних термінів за структурою, походженням і значенням.

Український термін	Англійський відповідник	Структура	Походження	Значення
Операційна система	Operating system	Складний термін	Калька	Функціональний
Графічний інтерфейс користувача (GUI)	Graphical User Interface (GUI)	Терміносполука + аббревіатура	Калька + аббревіатура	Функціональний
Алгоритм	Algorithm	Простий термін	Запозичення	Концептуальний
Кеш-пам'ять	Cache memory	Словосполучення	Калька	Функціональний
Хмарні обчислення	Cloud computing	Складний термін	Калька	Концептуальний
Файлова система	File system	Складний термін	Калька	Функціональний
База даних	Database	Складний термін	Калька	Функціональний
Комп'ютерна мережа	Computer network	Складний термін	Калька	Функціональний
Штучний інтелект	Artificial intelligence	Складний термін	Калька	Концептуальний
Програмне забезпечення	Software	Складний термін	Калька	Функціональний
Мікропроцесор	Microprocessor	Простий термін	Запозичення	Функціональний
Компілятор	Compiler	Простий термін	Запозичення	Функціональний
Буфер обміну	Clipboard	Словосполучення	Калька	Функціональний
Віртуальна машина	Virtual machine	Складний термін	Калька	Функціональний

Електронна таблиця	Spreadsheet	Складний термін	Калька	Функціональний
Термінальний сервер	Terminal server	Складний термін	Калька	Функціональний

Продовження таблиці 1.2

Український термін	Англійський відповідник	Структура	Походження	Значення
Архітектура комп'ютера	Computer architecture	Складний термін	Калька	Концептуальний
Протокол передачі даних	Data transmission protocol	Складний термін	Калька	Концептуальний
Мережевий адаптер	Network adapter	Складний термін	Калька	Функціональний
Інтегроване середовище розробки (IDE)	Integrated development environment (IDE)	Терміносполука + аббревіатура	Калька + аббревіатура	Функціональний

Усі ці типологічні характеристики вказують не лише на структурну складність комп'ютерної термінології, а й на її глибоку взаємодію з мовою, культурою та технологічним контекстом. Розуміння цієї системи — ключ до ефективного й точного перекладу, здатного передати як технічну суть, так і мовну природність.

1.3 Динаміка новотворів і роль неологізмів у ІТ-лексиконі

Інформаційні технології — одна з тих галузей, де мова розвивається з такою самою швидкістю, як і сам предмет опису. У ній нові слова не просто з'являються — вони народжуються, видозмінюються, змінюють значення і функцію, а потім або вкорінюються, або зникають. Неологізм у ІТ-лексиконі — це не лише нове слово, а й реакція на технологічну або соціальну потребу. Його поява найчастіше обумовлена браком відповідника в наявному мовному арсеналі, що змушує фахівців і перекладачів шукати нові засоби мовної репрезентації понять.

На відміну від літературних неологізмів, що часто є авторськими або стилістичними іграми, ІТ-неологізми мають переважно утилітарну функцію — точно, стисло й однозначно позначити нове явище чи процес. Наприклад, слово *крипто валюта* стало необхідним одразу після поширення Bitcoin, інакше опис

цифрових грошей вимагав би громіздких конструкцій. Сюди ж належать такі терміни, як *стрімінг*, *фреймворк*, *девайс*, що виникли не як наслідок лінгвістичного експерименту, а внаслідок технологічної необхідності. Водночас не кожен новотвір є неологізмом у повному сенсі — слід розрізняти кальки (наприклад, *штучний інтелект* як переклад *artificial intelligence*), адаптовані запозичення (*лайк*, *сервер*), і сленгізовану лексику, що має обмежену сферу вжитку (*фрілансер*, *бекендирик*, *технар*).

Список головних чинників появи нових термінів у сфері ІТ доволі передбачуваний, однак кожен із них має свою специфіку:

- стрімкий технологічний розвиток і постійне оновлення концепцій, які ще вчора були футуристикою, а сьогодні — реальністю (наприклад, *квантові обчислення*, *віртуальна реальність*, *хмарні сервіси*);
- глобальний вплив англомовного середовища, яке продукує первинні терміни і диктує моду на найменування;
- складність перекладу вузькоспеціалізованих понять, коли необхідно зберегти не лише форму, а й семантичну глибину, культурний підтекст або навіть емоційне забарвлення.

Механізми створення нових термінів у цьому контексті надзвичайно різноманітні. Найчастіше застосовується *калькування*, особливо коли структурна модель іншомовного терміну легко відтворюється українською: *машинне навчання* (від *machine learning*), *нейромережа* (від *neural network*), *інтернет речей* (від *Internet of Things*). Проте, коли калька виглядає громіздко або неприродньо, перевага надається *запозиченню*: приклади *браузер*, *чат*, *хештег* стали органічною частиною мови, хоча їх фонетичний вигляд далекий від традиційної української лексики [4, с. 74].

Варто також згадати про *внутрішнє термінотворення*, коли нові слова формуються з уже наявного словникового ресурсу: *флешка* (скорочення від *флеш-накопичувач*), *сайт* (спрощення від *вебсайт*), *мережа* (в ІТ-контексті — комп'ютерна мережа). Активно використовується і *словотвірне комбінування*,

як-от у словах *кібербезпека*, *нанотехнології*, *технопарк*, що поєднують класичну українську словотворчу модель із новітніми смисловими компонентами.

Динаміка закріплення новотворів залежить не лише від частоти вживання, а й від низки інституційних та соціальних чинників. Наприклад, на міжнародному рівні уніфікацію термінології здійснюють структури на кшталт ISO або IEEE, а також великі технологічні компанії, які формують «де-факто» стандарти. У науковому середовищі терміни закріплюються шляхом багаторазового використання в публікаціях, дослідженнях і технічній документації. Професійні ІТ-спільноти, що активно обговорюють ці терміни на форумах, також сприяють усталенню певного варіанта. Наприклад, *блокчейн* швидко вкоренився в обох мовах — завдяки частому згадуванню в експертних колах і бурхливому обговоренню в суспільстві.

Втім, адаптація новотворів до українського мовного простору не минає без ускладнень. Іноді кілька варіантів конкурують між собою, не маючи очевидного переможця. Термін *застосунок* довго сперечався з *аплікацією*, хоча останній є калькою з англійської *application* і зручно вписується в україномовне середовище. Схожа ситуація з термінами *сайт* та *вебсайт*: перший простіший, другий — точніший [2, с. 49].

На вибір терміну також впливають мовні тенденції: прагнення до милозвучності, відповідності нормам правопису. Іноді це призводить до компромісів або появи гібридів, як у випадку зі словом *смарт пристрій* — приклад спроби «втиснути» англомовний концепт у звичну українську морфологію.

Термінологія ІТ-сфери не стоїть на місці. Вона все більше тяжіє до лаконічності та універсальності, що підвищує зрозумілість і швидкість сприйняття. При цьому стирається межа між академічною термінологією та повсякденною лексикою: слова на кшталт *апдейт*, *баг*, *токен*, *запускати програму* вже не звучать фахово — вони стали частиною буденної мови. Очевидно, що нові підходи до творення термінів передбачають креативність і контекстуальність замість сліпого наслідування зразків. Наприклад, поява

неологізму *зумитися* (від *Zoom*) — показник того, як швидко і природно може відбутися мовна інновація в реакції на соціальні події.

1.4 Стилiстичнi та функцiональнi особливостi термiнiв у текстах технiчної документацiї

У текстах технiчної документацiї термiни виконують не лише номiнатиvну, а й чiтко визначену функцiональну роль. Передусiм, iхнє головне завдання — забезпечити однозначнiсть. Слова, якi в загальновживанiй мовi можуть мати кiлька значень, у межах технiчного дискурсу набувають виключно спецiального сенсу. Наприклад, термiн *port* у загальнiй мовi може означати порт як мiсце для швартування, але у комп'ютерному контекстi — це *мережевий порт*, логiчна точка взаємодiї мiж пристроями. Така чiткiсть дозволяє уникати непорозумiнь навить у складних iнструкцiях чи протоколах, де неточнiсть може призвести до системного збою або некоректної взаємодiї мiж компонентами.

Системнiсть термiнологiї полягає у взаємозалежностi понять, якi формують єдину термiносистему. Наприклад, термiни *iнтерпретатор*, *комплiлятор*, *вихiдний код* та *машинна iнструкцiя* функцiонують у межах однiєї логiчної групи, що вiдображає процеси перетворення програмного коду. Такий взаємозв'язок дозволяє не лише структурувати знання, а й формувати узагальненi схеми, що сприяють ефективнiшому засвоєнню iнформацiї. Не менш важливою є i стандартизацiя термiнiв: мiжнароднi стандарти, як-от ISO/IEC або рекомендацiї IEEE, регламентують не тiльки самi термiни, а й правила iхнього вживання. Наприклад, ISO 2382 визначає термiн *cloud computing* i його переклад як *хмарнi обчислення*, що забезпечує унiфiкованiсть у перекладах технiчних текстiв незалежно вiд краiни.

У функцiональному аспектi термiнологiя тяжiє до мовної економiї. Скорочення й абрeвiатури (типу *RAM*, *API*, *LAN*) є не лише засобом стислостi, а й вiдображенням належностi до професiйного дискурсу. У деяких випадках уживання розгорнутих форм навить вважається зайвим або надмiрним, якщо

йдеться про документи для фахівців. Тим паче, коли мова йде про терміносполуки, такі як *операційна система реального часу* або *архітектура програмного забезпечення*, — ці формули створюють стійкі структури, які не потребують заміни чи модифікації у межах тексту [1, с. 53].

У стилістичному плані технічна термінологія характеризується максимальною формальною нейтральністю. Тут немає місця емоційним маркерам чи суб'єктивній оцінці — навіть у випадку опису проблем або помилок, терміни залишаються суворо описовими: *системна помилка*, *збій виконання*, *неправильна конфігурація*. Така лексика не викликає оцінних конотацій, натомість зберігає нейтральний, інформативний тон.

Варто також звернути увагу на стабільність термінів: у технічних текстах небажана варіативність. Якщо один раз використано форму *віртуальна машина*, вона повинна залишатися сталою впродовж усього документа. Заміна її на *віртуальний комп'ютер* чи *імітаційне середовище* не лише дезорієнтує читача, а й порушує логічну послідовність викладу. Номінативність — ще одна помітна риса: переважають іменники або іменникові словосполучення, які фіксують поняття без динаміки: *сховище даних*, *протокол шифрування*, *інтерфейс користувача*. Дієслівні форми використовуються лише в описах процедур або кроків, не в самій термінології [3, с. 359].

Для технічного стилю характерний аналітичний синтаксис. Прості конструкції на кшталт *система збереження файлів* або *механізм автентифікації* на перший погляд здаються громіздкими, але саме завдяки їм досягається точність і повнота визначень. Цей стиль тяжіє до надлишковості, однак у рамках інструктивного тексту він є функціональним і виправданим.

Процес інтеграції англomовної термінології в українську технічну мову відбувається багатоступенево. Значна частина термінів запозичується без змін, зберігаючи первинну форму: *сервер*, *браузер*, *драйвер*, *апгрейд*. Вони вже настільки вкоренилися у професійному вжитку, що сприймаються як невід'ємна частина лексики. Інший шлях — калькування, тобто дослівний переклад структур: *операційна система*, *мова програмування*, *нейронна мережа*. У таких випадках

зберігається логіка терміну, але він адаптується до граматичної системи української. Гібридні форми є результатом поєднання: *вебінтерфейс*, *інтернет-магазин*, *онлайн-платформа* — це вже не просто запозичення, а частина нової лінгвістичної реальності.

Фонетична й морфологічна адаптація — ще один поширений механізм. Наприклад, термін *кодер* у деяких контекстах поступово витісняється варіантом *кодувальник*, який краще вписується в українську словотвірну систему. Або ж *сайт* трансформується у *вебсайт*, де додається уточнювальний елемент, що розширює семантику без втрати змісту.

Що стосується структури технічної документації, то терміни відіграють у ній не просто інформативну, а структуроутворюючу роль. Наприклад, у розділах типу *Вимоги до системи*, *Технічні характеристики*, *Методи тестування* саме терміни виконують навігаційну функцію, допомагаючи швидко зорієнтуватися в документі. Їх також використовують як опорні елементи для класифікації об'єктів: наприклад, *типи процесорів*, *категорії пам'яті*, *рівні безпеки*. У більш концептуальних розділах, де формується спеціалізований понятійний апарат, терміни набувають ще більшої ваги: вони виступають основою для створення нових знань, їх пояснення та впровадження в навчальні або робочі контексти [8, с. 94].

Сучасна технічна документація демонструє тенденцію до лаконізації. Довгі терміносполуки на кшталт *твердотільний накопичувач* все частіше замінюються аббревіатурами — *SSD*. Така заміна є зручною не лише в тексті, а й на рівні сприйняття: коротка форма легше зчитується й не втрачає змісту для цільової аудиторії. Паралельно з цим зростає роль автоматизованої стандартизації: термінологічні бази даних, глосарії й системи машинного перекладу (як-от SDL Trados або Memsource) дедалі частіше використовуються для забезпечення консистентності термінів у великих масивах документації [5, с. 174].

В умовах глобалізації особливої ваги набувають локалізаційні стратегії: один і той самий термін може мати різні форми залежно від аудиторії — скажімо, *мережевий інтерфейс* може перетворитися на *мережевий роз'єм* у текстах для

користувачів-початківців, навіть якщо йдеться про один і той самий об'єкт. Така адаптація не є відступом від точності — навпаки, вона забезпечує гнучкість, необхідну для ефективної комунікації між фахівцями і кінцевими користувачами.

Висновки до першого розділу

Комп'ютерна термінологія є невід'ємною складовою сучасного науково-технічного дискурсу та відіграє ключову роль у комунікації між фахівцями галузі. Її формування відбувалося під впливом технологічного розвитку, міжнародних стандартів та мовної практики, що визначило динаміку й специфіку терміносистеми.

Аналіз структурних та семантичних особливостей термінів засвідчив їхню неоднорідність, що зумовлено як походженням, так і функціональним призначенням. Запозичені, кальковані та власне українські терміни утворюють складну, взаємопов'язану систему, де стандартизація відіграє вирішальну роль.

Динаміка новотворів у сфері ІТ є відображенням швидких змін у технологічному середовищі. Неологізми активно входять у лексикон, що впливає на лексичну норму, адаптаційні процеси та принципи перекладу англomовних термінів.

Стилістичні та функціональні особливості термінів у технічних текстах визначають їхню точність, номінативність і стандартизованість. Вони виконують структурну, класифікаційну та навігаційну функції, забезпечуючи ефективну передачу інформації.

Комп'ютерна термінологія є динамічною мовною системою, що поєднує галузеві стандарти, лінгвістичні принципи та технологічні інновації. Її подальший розвиток залежатиме від взаємодії між науковцями, практиками та мовними інститутами, що сприятиме удосконаленню терміносистеми та її адаптації до сучасних потреб.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ПЕРЕКЛАДУ ГАЛУЗЕВОЇ ТЕРМІНОЛОГІЇ

2.1 Переклад спеціальної лексики у світлі сучасного перекладознавства

Сьогодні переклад спеціальної лексики, передусім термінів комп'ютерної галузі, трактують як активну форму інтелектуальної інтерпретації, де ключову роль відіграють когнітивні й комунікативні стратегії поряд з мовним аналізом. У цьому контексті виокремлюють кілька основних підходів до перекладу спеціалізованої лексики, кожен з яких висвітлює різні аспекти трансформації вихідного терміну.

З позиції *лінгвістичного підходу*, переклад термінів передбачає аналіз граматичних, лексичних і стилістичних трансформацій. Серед типових трансформацій, які застосовуються при перекладі англomовної комп'ютерної термінології, варто назвати транспозицію — зміну частини мови при збереженні змісту (наприклад, *to compile* — *компіляція*), калькування — буквальний переклад структурних компонентів терміну (*cloud computing* — *хмарні обчислення*), а також компресію — скорочення обсягу мовного вираження без втрати змісту (*random-access memory* — *оперативна пам'ять*). Ці трансформації дозволяють адаптувати іншомовну лексику до граматичних і стилістичних норм української мови, проте часто потребують додаткової інтерпретації терміну у фаховому контексті [3, с. 42].

Когнітивний підхід підкреслює, що переклад спеціальної лексики є не лише мовною, а насамперед когнітивно-комунікативною діяльністю. Згідно з цим підходом, терміни виступають як мовні репрезентації певних концептів, що формуються в межах конкретної науково-технічної картини світу. Тому переклад має передбачати перенесення не лише мовного означення, а й концептуального змісту терміну до ментального простору цільової аудиторії. Наприклад, термін *thread* у контексті багато потокової обробки даних не може перекладатися без урахування концептуального навантаження цього поняття в галузі ІТ. З огляду на

це, перекладачеві необхідно не лише володіти спеціальними знаннями, а й орієнтуватися в когнітивних схемах цільової культури.

У межах *функціонально-комунікативного підходу* увага зосереджується на функції терміну в конкретному тексті, а не виключно на його формальній відповідності вихідному слову. Такий підхід вимагає від перекладача враховувати прагматичний аспект — як саме термін впливає на сприйняття тексту, яку роль виконує в комунікативній структурі. Наприклад, переклад терміну *crash* як *збій системи* буде більш релевантним у технічній документації, ніж варіанти на зразок *аварія*, які мають загальніше значення. Орієнтація на функцію дозволяє досягати адекватності навіть у тих випадках, коли пряма лексична відповідність відсутня.

Окрему увагу в теорії перекладу спеціальної лексики приділяють *проблемі еквівалентності*. У перекладознавстві розрізняють повну, часткову та нульову еквівалентність. У галузі комп'ютерної термінології часто трапляється ситуація часткової або нульової еквівалентності, коли в українській мові бракує точного відповідника для англomовного терміну. Наприклад, термін *cookie* не має прямого аналога в українській мові, тому вживається або в транскрибованому вигляді (*кукі*), або описово (*файл із даними користувача, збережений браузером*). У таких випадках перекладач змушений балансувати між збереженням термінологічної точності та зрозумілістю для цільової аудиторії.

Особливої ваги набуває питання *варіантних відповідників*, коли існує кілька допустимих перекладів одного терміну. У таких випадках вибір залежить від контексту, текстового жанру, очікувань цільової аудиторії та термінологічної усталеності. Так, англomовне *stiffness* у різних галузях може перекладатися як *жорсткість*, *ригідність* або *остійність*, кожен з яких не лише має свої конотації, а й функціонує в певному фаховому середовищі. Вибір адекватного варіанта вимагає від перекладача не лише лінгвістичної компетентності, а й глибокого знання предметної сфери.

У процесі перекладу комп'ютерної термінології перекладач активно використовує перекладацькі трансформації, які допомагають досягти як змістової, так і функціональної еквівалентності в умовах міжмовної асиметрії. Ці

трансформації охоплюють різні рівні мовної системи — від лексики до граматики і стилістики — і дають змогу гнучко адаптувати вихідний термін до норм і очікувань цільової аудиторії.

На *лексико-семантичному рівні* поширеною є синонімічна заміна, коли оригінальний термін замінюється українським відповідником із близьким значенням, але не завжди дослівним. Наприклад, *debugging* може передаватися як *налагодження*, що фокусує увагу не лише на пошуку помилок, а й на їх усуненні. Описовий переклад застосовується, коли термін не має точного відповідника або має вузькоспеціалізоване значення — *firmware* часто перекладають як *вбудоване програмне забезпечення*, що розкриває сутність поняття. Антонімічний переклад трапляється рідше, але його застосування виправдане, коли така трансформація допомагає краще зберегти функціональне навантаження вислову — наприклад, *fail-safe system* може бути перекладена як *система захисту від збоїв*, де суть дії подається через заперечення небажаного результату.

Граматичні трансформації охоплюють зміну частин мови (транспозиція), наприклад: *to install* — *установлення*; компенсацію, коли втрачену інформацію відновлюють в іншій частині речення: *the system failed silently* — *система вийшла з ладу, не видаючи повідомлень*; а також зміну порядку слів, характерну для англо-українського перекладу, з огляду на відмінності синтаксису: *real-time processing* — *обробка в реальному часі*.

На *стилістичному рівні* використовуються такі трансформації, як експресивація, коли перекладач свідомо посилює емоційне чи оцінне забарвлення: *crash* — *раптовий збій*; логізація — спрощення стилістично насиченого терміну до нейтрального: *blue screen of death* — *критична помилка системи*; модернізація або архаїзація термінів трапляється рідше, але може бути виправданою залежно від цільового стилю документації: старіші варіанти на зразок *друкарська машинка* для *terminal* у ретро контекстах або *накопичувач* замість *диск* в офіційно-ділових текстах.

Важливим аспектом, що впливає на перекладацькі рішення, є екстралінгвістичні чинники. Так, культурні та соціолінгвістичні особливості —

ідеологія, технічна традиція, очікування цільової аудиторії — прямо впливають на те, як саме перекладається термін. Наприклад, термін *master/slave* у сучасному українському перекладі все частіше замінюється на *ведучий/підлеглий* або *основний/допоміжний* через етичні й соціокультурні зміни в підходах до технічної лексики. Крім того, в різних професійних середовищах одні й ті самі терміни можуть мати усталені, але відмінні відповідники, що зумовлює необхідність уніфікації.

У цьому контексті стандартизація та кодифікація відіграють ключову роль. Глосарії, термінологічні бази, такі як EuroTermBank або термінологічні списки ДСТУ, забезпечують стабільність і однозначність перекладу. Застосування стандартів сприяє узгодженості в межах проекту й у ширшому професійному середовищі. Наприклад, у документації, що супроводжує вільне програмне забезпечення, активно використовуються відкриті глосарії, підтримувані спільнотами, як-от KDE або GNOME, що пропонують стандартизовані переклади термінів [9, с. 114].

Розглянемо низку англійських термінів, уживаних у комп'ютерній документації, та особливості їх перекладу:

1. *Bootloader*

У переважній більшості українських джерел цей термін перекладається як *завантажувач системи* або просто *завантажувач*. Такий переклад є описовим, оскільки вихідний термін складається з двох частин (*boot* як метафора запуску, *loader* як завантажувач). Часто трапляється і калька — *бутлоудер* — у неофіційних чи вузькопрофільних джерелах, проте її вважають небажаною через порушення норм української терміносистеми.

2. *Debugging*

Перекладається найчастіше як *налагодження* — це синонімічна заміна з акцентом на результативність дії (усунення помилок), хоча буквально термін означає процес виявлення помилок. В інших контекстах уживають також *відлагодження*, що ближче до кальки, проте стилістично менш усталене. У

низці документів трапляється описовий переклад, наприклад: *process of detecting and removing bugs* — *процес виявлення та усунення помилок*.

3. *Firmware*

Цей термін у сучасному українському перекладі зазвичай передається як *вбудоване програмне забезпечення*, що є класичним прикладом описового перекладу. Окремі перекладачі застосовують транскрипцію — *фірмвар*, однак така форма переважно не рекомендована в офіційній документації через її неінформативність для нефахової аудиторії. З граматичного боку спостерігається транспозиція, коли англійський іменник зливається в описовий іменниково-прикметниковий зворот.

У перекладі комп'ютерної термінології сьогодні особливо важливо гнучко поєднувати лексичні, граматичні та прагматичні трансформації із культурною чутливістю, забезпечуючи рівновагу між термінологічною точністю та практичною релевантністю. Ефективний переклад у цій сфері неможливий без балансу між стандартизацією, концептуальною точністю й адаптацією до професійної практики та культурного контексту.

2.2 Класифікація стратегій відтворення англomовної термінології

У перекладацькій практиці комп'ютерних текстів стратегічний вибір способу передачі термінів формує основу термінологічної точності, текстової зрозумілості та професійної адекватності. Враховуючи специфіку ІТ-сфери, де значна частина термінів є неологізмами або полісемантичними одиницями, перекладач має діяти не лише як мовний посередник, а як інтерпретатор концептуальних моделей, закладених у вихідному тексті.

Орієнтуючись на когнітивні, лінгвістичні та прагматичні чинники — зокрема, структуру понятійного апарату, очікування цільової аудиторії та функціональну роль терміну — він застосовує різноманітні моделі трансляції. Ці моделі диференціюються за типом термінологічної репрезентації (пряма калька, адаптивний переклад, пояснювальний конструкт тощо) та рівнем контекстуальної

інтеграції в текст (стилістична відповідність, жанрова функціональність, термінологічна стандартизація).

Одна з найпоширеніших стратегій — *еквівалентний переклад*, коли для англomовного терміну використовується вже усталений український відповідник. Наприклад, *software* стабільно перекладається як *програмне забезпечення*, що є загально визнаним стандартом у фаховій літературі.

Калькування полягає в буквальному перекладі морфем або складових терміну. Наприклад, *firewall* іноді перекладається як *вогняна стіна*, що дослівно відповідає англійській структурі, але здебільшого адаптується до *брандмауер*, зважаючи на усталене фахове використання. Цей приклад ілюструє обмеження калькування при перекладі термінів, які мають переносне або метафоричне значення [2, с. 74].

Транскодування, або *транслітерація*, передбачає передачу іншомовного терміну на основі його звучання. Наприклад, *router* передається як *роутер*. Такий підхід широко застосовується до термінів, що ще не мають сталого еквівалента, особливо у сфері споживчої техніки чи веб-інтерфейсів.

Описовий переклад використовується для пояснення функціонального значення терміну. Це особливо ефективно для нових або вузькоспеціалізованих понять. Наприклад, *debugging* може бути перекладене як *виявлення та усунення помилок*, що передає суть дії, навіть якщо термін ще не повністю адаптований у мові-реципієнті.

Ще одна стратегія — *запозичення* — полягає у перенесенні англomовного терміну без змін або з фонетичною адаптацією. Прикладом є *plugin* — *плагін*. Ця стратегія зберігає термінотворчу автономність, але потребує зваженого використання в текстах, орієнтованих на широку аудиторію [2, с. 76].

Таблиця 2.1. Класифікація за рівнем адаптації.

Стратегія	Адаптація до норм мови	Приклад
Еквівалентний переклад	Висока	<i>Software</i> — програмне забезпечення
Калькування	Середня	<i>Firewall</i> — вогняна стіна
Транскодування	Низька	<i>Router</i> — роутер
Описовий переклад	Висока	<i>Debugging</i> — виявлення помилок

Запозичення	Залежить від контексту	<i>Plugin</i> — <i>плагін</i>
-------------	------------------------	-------------------------------

У цій класифікації ключовим критерієм виступає відповідність мовним нормам цільової мови. Висока адаптація передбачає інтегрованість терміну в морфологічну та синтаксичну структуру української мови. Середній рівень пов'язаний із прямим перенесенням форми зі збереженням частини іншомовної структури, а низький — із мінімальною мовною обробкою.

Таблиця 2.2. Класифікація за функціональною точністю.

Критерій	Характеристика	Приклад
Термінологічна точність	Збереження галузевого значення	<i>Firmware</i> — <i>вбудоване ПЗ</i>
Комунікативна ефективність	Зрозумілість для нефахового користувача	<i>Cookie</i> — <i>файл даних користувача</i>
Культурна релевантність	Відповідність соціолінгвістичним очікуванням і мові професійного середовища	<i>Blacklist</i> — <i>список блокування (нейтралізація)</i>

Ця класифікація акцентує не лише на лінгвістичному вимірі перекладу, а й на його відповідності очікуванням цільової аудиторії. Залежно від типу документації (технічна інструкція, рекламна брошура, навчальний посібник) домінуючим може бути один із критеріїв. Наприклад, у програмній документації пріоритет надається термінологічній точності, тоді як у публічних інтерфейсах — комунікативній ефективності та культурній релевантності [1, с. 214].

У процесі перекладу термінів комп'ютерної документації стратегія перекладача залежить не лише від лінгвістичних або функціональних чинників, а й від джерела, на яке він спирається при ухваленні рішення. За цим критерієм можна виокремити *нормативні*, *інтуїтивні* та *корпусні* стратегії. Кожна з них має свою сферу застосування і рівень достовірності, що впливає на якість і сталість перекладу.

Нормативна стратегія ґрунтується на офіційно затверджених джерелах, зокрема глосаріях, термінологічних словниках, базах даних (на кшталт EuroTermBank, IATE) та стандартах (наприклад, ISO/IEC чи українські ДСТУ). Такі джерела забезпечують уніфікацію термінів і особливо важливі для перекладу

офіційної документації, що потребує юридичної точності та міжсистемної сумісності.

Інтуїтивна стратегія застосовується у випадках, коли відсутні нормативні джерела або коли перекладач має справу з новими, ще не стандартизованими термінами. Тут рішення базується на глибокому розумінні контексту, когнітивній інтерпретації та досвіді перекладача. Інтуїтивний підхід нерідко використовується для створення нових термінів або стилістичної адаптації, зокрема в неофіційних або експериментальних текстах.

Корпусна стратегія спирається на аналіз великих обсягів фахових текстів або корпусів, зокрема перекладених документів, технічних форумів, довідників. Вона дозволяє встановити, які терміни фактично функціонують у мові-реципієнті, яка їхня частотність і контекст вживання [5, с. 31]. Цей підхід сприяє адаптації перекладу до реального вжитку та допомагає узгоджувати терміни в межах одного проєкту або галузі.

Таблиця 2.3. Класифікація за джерелом стратегії

Тип стратегії	Джерело	Сфера застосування
Нормативна	Глосарії, стандарти (ISO, ДСТУ, технічні термінологічні бази)	Офіційна документація, сертифікація, навчальні матеріали
Інтуїтивна	Досвід перекладача, контекст, когнітивне сприйняття	Неофіційні тексти, новітні терміни, творчі переклади
Корпусна	Частотний аналіз у текстах, корпуси, технічні форуми	Професійне середовище, веб-документація, відкриті програмні проєкти

Вибір перекладацької стратегії нерідко обумовлюється не лише мовними чинниками, але й прагматичними параметрами тексту: жанром документації, цільовою аудиторією та очікуваним рівнем технічної компетентності реципієнта. Зокрема, у сфері ІТ-комунікації терміни функціонують не ізольовано, а як складники спеціалізованих концептосфер, що вимагає від перекладача застосування адаптивного підходу, здатного забезпечити як концептуальну точність, так і функціональну релевантність перекладу.

Таблиця 2.4. Практичні приклади: таблиця термінів і стратегій

Англійський термін	Український варіант	Стратегія	Тип стратегії (джерело)
Software	Програмне забезпечення	Еквівалентний переклад	Нормативна (ДСТУ 3966-2000)

Продовження таблиці 2.4

Англійський термін	Український варіант	Стратегія	Тип стратегії (джерело)
Router	Роутер	Транслітерація	Корпусна (форумні джерела, документи Cisco)
Plugin	Плагін	Запозичення	Корпусна / нормативна
Firewall	Брандмауер / вогняна стіна	Калькування / запозичення	Інтуїтивна / нормативна
Debugging	Налагодження	Синонімічна заміна	Нормативна / корпусна
Bootloader	Завантажувач	Описовий переклад	Інтуїтивна
Firmware	Вбудоване програмне забезпечення	Описовий переклад	Нормативна / корпусна
Cloud computing	Хмарні обчислення	Калькування	Нормативна
Token	Токен / маркер / жетон	Вибірковий переклад	Інтуїтивна / корпусна
Cookie	Куки / файл користувача	Запозичення / описовий	Корпусна / інтуїтивна
Benchmark	Еталон / тест продуктивності	Синонімічна заміна	Інтуїтивна / корпусна

Класифікація стратегій передачі англомовної термінології в українськомовній комп'ютерній документації виявляє складну систему трансформацій, що охоплює різні рівні мовної адаптації — від калькування та транслітерації до глибших семантичних перетворень і комунікативного уточнення. Кожна з описаних стратегій виявляє не лише перекладацьку техніку, але й когнітивну інтерпретацію терміносфери, що формується під впливом лінгвістичних та екстралінгвістичних чинників.

2.3 Фактори вибору перекладацьких стратегій калькування, адаптації та транслітерації

Вибір між калькуванням, адаптацією або транслітерацією англомовної термінології значною мірою визначається лінгвістичними чинниками, зокрема морфологічною сумісністю та семантичною прозорістю. Якщо структура української мови дозволяє зберегти форму іншомовного терміну без втрати змісту,

калькування стає зручним і природним способом перекладу. Прикладами можуть бути терміни на кшталт *input device* — *пристрій введення*. Водночас, якщо калька утворює незрозуміле або двозначне слово, перекладачі вдаються до описових або адаптованих варіантів. Семантична прозорість має важливе значення: терміни, які після калькування залишаються зрозумілими для користувача, мають перевагу. Крім того, фонетична прийнятність транслітерованих слів також відіграє роль: *роутер* чи *сервер* уже стали звичними в українському мовному середовищі, тоді як надто складні або штучні за звучанням слова можуть викликати спротив або потребу в адаптації.

Не менш важливу роль відіграють культурні та соціолінгвістичні чинники, зокрема ступінь інтернаціоналізації терміну та очікування цільової аудиторії. Якщо певний термін широко використовується в глобальному професійному чи побутовому вжитку (наприклад, *комп'ютер*, *сервер*, *файл*), транслітерація сприймається природно і навіть очікувано.

Жанрова приналежність тексту також безпосередньо впливає на вибір стратегії. У науково-технічній документації або стандартах найчастіше дотримуються максимальної точності, що сприяє калькуванню або використанню нормативних термінів. Натомість у маркетингових, рекламних або художніх текстах, де важлива стилістика, природність та емоційна подача, частіше застосовується адаптація.

Особливу увагу слід приділяти нормативним та стандартизуючим чинникам, адже наявність терміну в офіційних джерелах — таких як глосарії ДСТУ, ISO або термінологічні словники — часто визначає єдино припустимий варіант перекладу. У проєктах з високими вимогами до уніфікації та послідовності термінології така стандартизація стає обов'язковою умовою [4, с. 26].

Індивідуальні чинники перекладача, включно з рівнем його кваліфікації, досвідом роботи в галузі та стильовою чутливістю, теж впливають на стратегію. Досвідчені перекладачі можуть гнучко варіювати між калькуванням, адаптацією й транслітерацією залежно від жанру, мети та читацької аудиторії, тоді як новачки часто обирають більш прямолінійні або автоматизовані підходи.

Висновки до другого розділу

Огляд класифікацій перекладацьких стратегій та чинників їх вибору дає змогу усвідомити, що переклад термінології — це не механічний процес підстановки відповідників, а комплексне мовно-когнітивне завдання, яке потребує зваженого підходу. Вибір стратегії — еквівалентного перекладу, калькування, транслітерації чи описового варіанту — залежить не лише від структурних особливостей мов, а й від цільової аудиторії, контексту та функціонального навантаження терміну в тексті. Кожна стратегія має свої переваги й обмеження, які потрібно враховувати задля досягнення як точності, так і зрозумілості перекладу.

Класифікація стратегій за ступенем адаптації, функціональною точністю або джерелом вибору (нормативним, інтуїтивним, корпусним) дозволяє глибше зрозуміти логіку перекладацьких рішень. Наприклад, нормативна стратегія актуальна для стандартизованої термінології, інтуїтивна — для нових або нестійких термінів, а корпусна — для виявлення усталених варіантів у реальному вжитку. Такий підхід забезпечує більшу гнучкість і точність у відтворенні змісту, що особливо важливо у сфері технічного та наукового перекладу.

Водночас аналіз чинників вибору конкретної стратегії — від лінгвістичних до індивідуальних — дає змогу побачити, наскільки багатовимірним є перекладацький процес. Морфологічна структура, фонетика, ступінь інтернаціоналізації, жанрові особливості тексту, а також досвід і чутливість самого перекладача — усе це утворює динамічну систему, в якій жодне рішення не є універсальним. Такий підхід формує підґрунтя для свідомого і аргументованого вибору перекладацької стратегії, що, своєю чергою, сприяє підвищенню якості фахового перекладу термінів.

РОЗДІЛ 3 УНОРМУВАННЯ ТЕРМІНОЛОГІЇ В УКРАЇНСЬКОМОВНОМУ ПЕРЕКЛАДІ КОМП'ЮТЕРНИХ ТЕКСТІВ

3.1 Термінологічна варіативність і проблема синонімії

У комп'ютерній термінології перекладач часто опиняється між двома крайнощами: буквалістичним відтворенням і надмірною адаптацією. Це поле, де варіативність не є винятком, а радше правилом. Причиною такого явища, як зазначають дослідники перекладознавства, є відсутність усталеного державного або галузевого стандарту, що регламентував би переклад англomовних ІТ-термінів українською. На практиці перекладачі користуються різними джерелами — від офіційних довідників Microsoft і Google до термінологічних баз спільнот чи власного досвіду. У результаті в українських перекладах комп'ютерної документації співіснують кілька варіантів одного терміну, які функціонують майже як синоніми, але несуть різні стилістичні та семантичні відтінки [11, с. 38].

Варто відзначити, що варіативність часто виникає не лише через мовну гнучкість, а й через часову еволюцію перекладацьких практик. Наприклад, ще десять років тому слово *upload* передавалося як «завантажити», за аналогією з *download*, а сьогодні частіше зустрічається варіант «вивантажити». Обидва переклади технічно коректні, але кожен має власну логіку: перший — зручний і зрозумілий для користувача, другий — більш точний із точки зору напрямку дії. Подібна конкуренція форм спостерігається і щодо термінів *logout*, *exit*, *sign out* — усі вони в українських версіях документації можуть передаватися як «вихід», «завершення сеансу» або «вийти з облікового запису». Ці варіанти співіснують у різних продуктах і системах, створюючи ефект термінологічної мозаїки, що іноді дезорієнтує користувача.

Відсутність єдиного стандарту часто ускладнює роботу перекладачів, які змушені балансувати між зрозумілістю для користувача і точністю передачі технічного змісту. Як наслідок, в одній і тій самій системі іноді зустрічаються несумісні варіанти: «зберегти файл» і «записати файл», «додаток» і «програма»,

«сховище» і «сховище даних» тощо. З точки зору користувацького досвіду це створює ризики плутанини — особливо тоді, коли дії, описані різними словами, насправді позначають одну і ту ж операцію. Для перекладача ж головна небезпека полягає у втраті термінологічної послідовності, що є ключовою вимогою до якості технічного перекладу [16, с. 25].

Можна припустити, що певна міра варіативності навіть корисна: вона дозволяє мові розвиватися, відбираючи найзручніші та найточніші варіанти. Проте коли йдеться про комп'ютерну документацію, така природна гнучкість мови стикається з необхідністю точності й стандартизації. Баланс між ними — головне завдання перекладача технічних текстів (Таб. 3.1).

Таблиця 3.1. Варіанти перекладу одного терміну.

Англійський термін	Варіанти перекладу	Джерело	Рекомендований варіант
<i>Upload</i>	завантажити, вивантажити	Microsoft Docs, Glosbe	вивантажити — точніше відображає напрямок передачі даних
<i>Download</i>	завантажити, отримати, звантажити	Google Translate, Lingvo	завантажити — найуживаніший і зрозумілий користувачу
<i>Logout</i>	вихід, завершення сеансу, вийти з облікового запису	Windows Interface, українські локалізації браузерів	вийти з облікового запису — найінформативніший варіант
<i>Application</i>	програма, додаток	офіційні переклади Android, Windows	додаток — відповідає сучасним українським нормам
<i>Storage</i>	сховище, пам'ять, накопичувач	технічна документація HP, Samsung	сховище — узгоджується з усталеними термінами «хмарне сховище»
<i>Save</i>	зберегти, записати	інтерфейси Word, Adobe, GitHub Docs	зберегти — універсальний і найчастіше вживаний

Сьогодні перекладацька практика все частіше орієнтується на усталення єдиних варіантів, що підтримуються глобальними корпораціями та українською ІТ-спільнотою. Проте остаточне вирішення проблеми синонімії залишається відкритим: мова технологій змінюється швидше, ніж будь-який стандарт встигає зафіксувати її норми. І саме в цій динаміці — живий пульс технічного перекладу, який поєднує точність інженерії з гнучкістю людського мовлення.

3.2 Роль перекладача у творенні нових українських відповідників

Перекладач у сфері комп'ютерної документації — це не лише посередник між мовами, а й активний учасник термінотворчих процесів. Саме він часто першим стикається з новим англomовним поняттям, для якого українська мова ще не має готового відповідника. У таких ситуаціях перекладач стає термінотворцем — фахівцем, що формує мовну норму, яка згодом може бути кодифікована у словниках або стандартах. Це особливо помітно в ІТ-сфері, де швидкий технологічний розвиток змушує мову постійно оновлюватися.

Успішне термінотворення ґрунтується на трьох критеріях: точності, природності та відповідності мовним нормам. Точність означає адекватне передання поняття без спотворення змісту. Природність — це здатність терміну органічно вписуватися в мовну систему, не виглядати чужорідним. Відповідність нормам — забезпечення граматичної, словотвірної й орфографічної узгодженості з українською мовною традицією. Якщо хоча б один із цих критеріїв порушено, термін може залишитися штучним, а користувач — не прийняти його.

Існує чимало прикладів, коли саме авторські рішення перекладачів ставали усталеними нормами. Так, варіант «додаток» для *application* спершу мав конкурентів — «програма», «застосунок», «аплікація» — однак саме «додаток» з часом закріпився в офіційних перекладах Android та Microsoft і нині сприймається як основний. Подібна історія й із терміном «обліковий запис» (*account*), який витіснив колишні варіанти «аккаунт» чи «рахунок користувача». Деякі новотвори, як-от «вивантаження» для *upload* або «знімок екрана» для *screenshot*, також народилися у перекладацькому середовищі, а не в академічних словниках [9, с. 62].

Можна припустити, що роль перекладача тут подібна до ролі мовного архітектора: він не лише відтворює чужу конструкцію, а й адаптує її до національного мовного простору, враховуючи логіку, милозвучність і зрозумілість для користувача. І саме від цієї чутливості до мовного матеріалу залежить, чи

стане новий термін живим елементом української технічної мови, чи залишиться штучною калькою (Рис. 3.1).

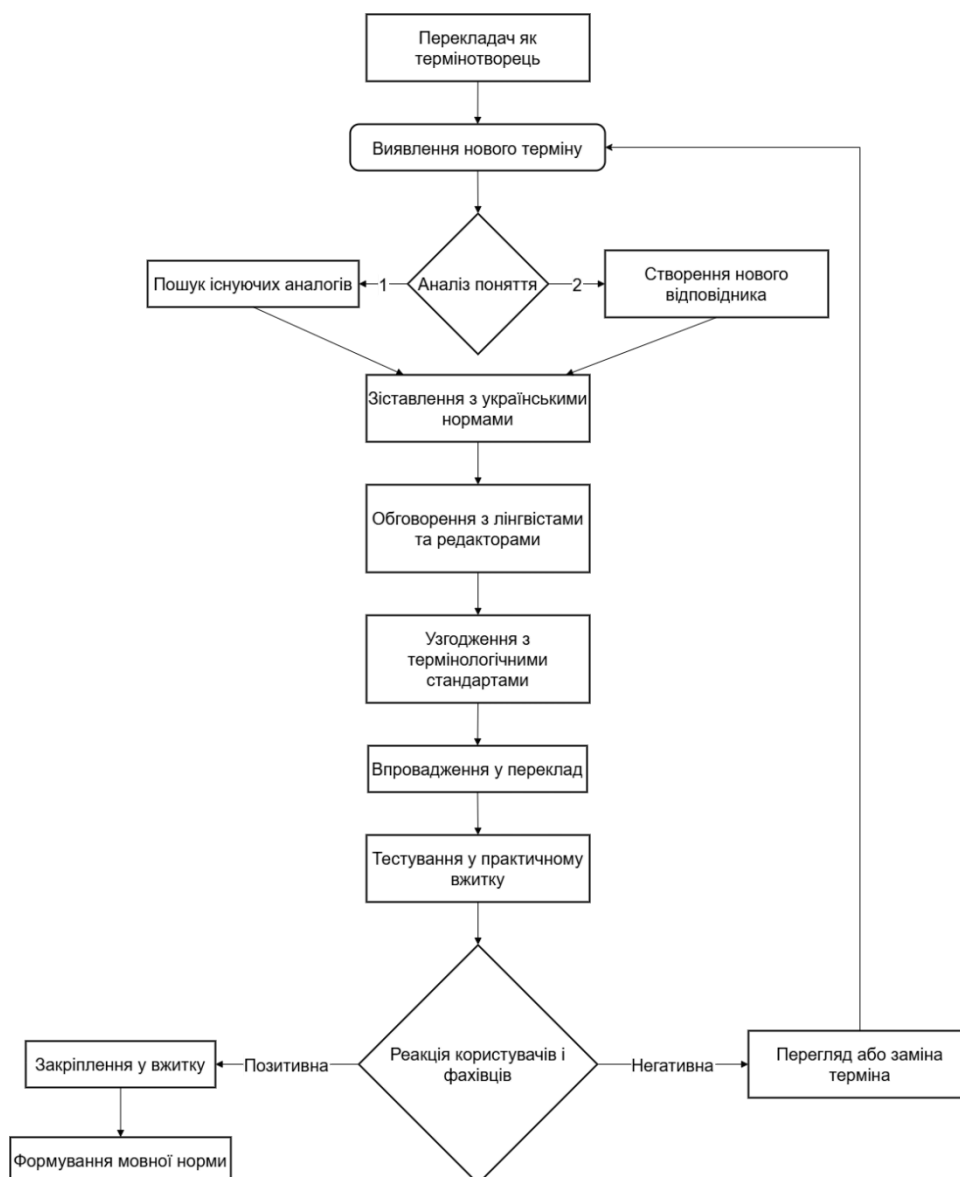


Рисунок 3.1 – Роль перекладача в термінотворенні.

Під час фахового перекладу перекладач виконує не лише функцію трансляції змісту, а й бере участь у формуванні термінологічної системи цільової мови. Кожен коректно підібраний відповідник сприяє уточненню й стандартизації української терміносистеми в галузі інформаційних технологій. Це свідчить про здатність мови адаптуватися до нових концептів і технологічних змін, забезпечуючи її функціональну актуальність у професійному середовищі.

3.3 Національні та міжнародні стандарти в сфері IT-термінології

Коли мова заходить про переклад комп'ютерної документації, найважливішим орієнтиром для перекладача стають термінологічні стандарти. Саме вони визначають, як правильно передавати спеціалізовані поняття українською, щоб переклади були не лише точними, а й послідовними у межах усієї галузі. Українські перекладачі насамперед спираються на національні стандарти — передусім ДСТУ 3966:2000 «Термінологічна робота. Основні положення та правила розроблення термінів» і ДСТУ ISO/IEC 2382:2015 «Інформаційні технології. Словник термінів». Ці документи встановлюють принципи створення, гармонізації та адаптації термінів у сфері IT.

Водночас глобальний вплив мають міжнародні стандарти ISO (зокрема ISO/IEC 2382, ISO 704, ISO 1087), які формують універсальні підходи до термінотворення і класифікації понять у галузі інформаційних технологій. Ці норми не завжди прямо перекладаються в українську практику, але саме вони визначають загальний напрямок — точність, однозначність і узгодженість технічної мови [7, с. 108].

Значний вплив на уніфікацію термінів мають також корпоративні глосарії, які фактично виконують функцію стандартів у межах конкретних екосистем. Найвідомішими серед них є **Microsoft Language Portal** та **Apple Style Guide**. Вони містять докладні інструкції щодо вживання термінів, граматичних форм, варіантів перекладу та стилістики інтерфейсу [3, с. 359]. Наприклад, саме Microsoft рекомендує використовувати «обліковий запис» замість «аккаунт», а Apple закріплює варіант «повідомлення» для *notification* і «налаштування» для *settings*. Такі внутрішні стандарти часто стають взірцем для перекладачів інших компаній і поступово формують загальний мовний стандарт у галузі [10, с. 61].

Слід зазначити, що стандарти виконують не лише регламентуючу, а й оптимізувальну функцію в роботі перекладача. Хоча вони частково обмежують інтерпретаційну свободу, водночас забезпечують уніфіковане термінологічне середовище, у межах якого кожен термін має чітко визначене значення, сферу

застосування та рекомендований відповідник. Така стандартизація сприяє підвищенню внутрішньої узгодженості технічної документації, полегшує її сприйняття кінцевими користувачами та забезпечує стабільність мовної системи в умовах динамічного розвитку галузевих знань (Таб. 3.2).

Таблиця 3.2. Стандартизація термінів

Термін	Стандарт (ДСТУ/ISO)	Вживання у продуктах	Відповідність
<i>Firewall</i>	ДСТУ ISO/IEC 2382:2015	Windows Security, macOS Settings	Брандмауер — офіційно унормований і закріплений у документації
<i>Settings</i>	Apple Style Guide, Microsoft Language Portal	macOS, iOS, Windows	Налаштування — рекомендований і послідовно вживаний варіант
<i>Update</i>	ISO/IEC 14764, Microsoft Glossary	Windows Update, Android System	Оновлення — узгоджений термін у всіх офіційних локалізаціях
<i>Network</i>	ДСТУ ISO/IEC 2382:2015	Cisco, Windows Network Center	Мережа — стандартний відповідник згідно з ДСТУ
<i>Database</i>	ISO/IEC 2382-17, ДСТУ 2226-93	MySQL, PostgreSQL, Oracle	База даних — усталений офіційний термін

Міжнародні та національні стандарти виконують функцію нормативної основи, що забезпечує структурну узгодженість термінологічних рішень у процесі перекладу технічних текстів. Вони формують спільну координатну систему, у межах якої кожен термін має визначене місце, функцію та контекстуальну прив'язку.

Незважаючи на постійні виклики, пов'язані з появою нових технологій, динамікою галузевих змін і розширенням семантичного поля, саме стандартизовані норми залишаються ключовим механізмом підтримки зрозумілості, інтероперабельності та стабільності технічної мови.

Водночас стандарти виконують роль своєрідного «мосту» між різними професійними спільнотами, забезпечуючи єдність термінології у міжнародних проєктах та міжгалузевій комунікації. Вони дозволяють уникати багатозначності та непорозумінь, що особливо важливо у сферах із високим рівнем відповідальності — від інженерії до медицини. Завдяки цьому переклад технічних текстів стає не лише мовним процесом, а й інструментом підтримки глобальної узгодженості знань та практик.

Крім того, стандарти мають динамічний характер: вони постійно оновлюються відповідно до розвитку науки й технологій, інтегруючи нові терміни та концепти. Це забезпечує їхню актуальність і дозволяє перекладачам працювати з сучасним матеріалом, не втрачаючи точності та системності.

Висновки до третього розділу

Під час аналізу термінологічної системи комп'ютерної документації стає очевидним, що мова технологій — це жива структура, у якій перекладач виступає не лише посередником, а й творцем. Варіативність і синонімія в ІТ-термінах не є ознакою хаосу, а радше природним наслідком пошуку точного мовного вираження для нових понять. Українська перекладацька практика поки що не має абсолютної уніфікації, проте саме завдяки цій динаміці мова адаптується до технічних реалій і поступово виробляє власні стійкі норми.

Роль перекладача в цьому процесі складно переоцінити. Він не лише адаптує чужі поняття, а й надає їм українського обличчя — добирає слова, що звучать природно, точно і зрозуміло. Саме перекладач часто стає першим, хто випробовує новий термін у практиці, і від його рішень залежить, чи приживеться слово, чи залишиться лише спробою. Приклади таких успішних рішень — «додаток», «обліковий запис», «знімок екрана» — показують, що жива мова може органічно інтегрувати найскладніші технічні поняття, якщо за їхнім творенням стоїть мовна інтуїція і професійна відповідальність.

Водночас стандарти — національні, міжнародні та корпоративні — виконують стабілізувальну функцію. Вони допомагають упорядкувати багатоманітність, забезпечуючи єдність і зрозумілість термінів у всій галузі. Проте стандарти не скасовують творчої ролі перекладача, а радше надають їй рамки, у яких народжується якісний і точний переклад. У цьому діалозі між нормою та мовною ініціативою і формується сучасна українська ІТ-термінологія — жива, гнучка, здатна не лише наздоганяти світові тенденції, а й створювати власні.

РОЗДІЛ 4 ПРАКТИЧНИЙ АНАЛІЗ ВІДТВОРЕННЯ АНГЛОМОВНОЇ ТЕРМІНОЛОГІЇ В УКРАЇНСЬКОМУ ПЕРЕКЛАДІ

4.1 Характеристика матеріалу дослідження та методика його аналізу

Нерідко саме дрібна деталь у технічному описі — забутий артикль, незмінений формант, непрозора метафора — стає тим місцем, де переклад комп'ютерної документації або працює бездоганно, або руйнує логіку всього тексту. Тому матеріал цього дослідження добирався не за принципом «що знайдеться», а як корпус, здатний показати динаміку термінотворення й перекладацьких рішень у реальних умовах функціонування англomовної ІТ-комунікації.

Матеріальну основу становлять три взаємопов'язані, але суттєво різні типи джерел: офіційна технічна документація (documentation), користувацькі інтерфейси (UI) та довідкові системи (help centers). Технічні документи — це насамперед посібники та специфікації до Visual Studio Code, Windows 11 Developer Documentation та GitHub Docs, оскільки саме вони містять найбільш репрезентативні й усталені терміни, що проходять внутрішню валідацію компаній-розробників. Усього до аналізу було включено 52 тексти обсягом приблизно 240 тисяч слів, що дозволяє працювати з повноцінним середовищем терміноодиниць, а не з вибірковими прикладами.

Інший пласт матеріалу — інтерфейсні написи: команди типу Run build task, Save as..., Reset settings, Notifications, File history та інші елементи локалізованих UI. На перший погляд це дрібні мовні фрагменти, але саме вони демонструють тенденції до граматичної компресії, активне використання імперативів та високу термінологічну сталість. Для аналізу було відібрано понад 400 унікальних інтерфейсних одиниць із Windows 11, Adobe Photoshop та Android Studio [42, с. 23].

Третя частина корпусу — довідкові розділи (Help, Troubleshooting, FAQ), оскільки саме тут термінологія вступає в контакт із пояснювальними описами.

Наприклад, терміни типу *buffer overflow*, *sandboxing*, *placeholder* або *dependency injection* в довідкових статтях не лише називають поняття, а й задають рамки його інтерпретації. Це робить матеріал особливо цінним для відстеження переходів між терміном і визначенням, що важливо для українського перекладу.

Жанрова належність корпусу охоплює три комунікативні формати: інструктивний (*step-by-step manuals*), описовий (*feature overviews*) та нормативний (*specifications*). Разом вони формують збалансовану модель термінологічного середовища, де уживаються як апаратні терміни (*network adapter*, *thermal throttling*), так і програмні (*event handler*, *commit*, *rollback*), а також гібридні одиниці, що існують на межі концептів, наприклад *cloud instance* чи *performance bottleneck*. Загалом у корпусі зафіксовано приблизно 870 унікальних термінів, що дозволяє здійснити класифікацію за структурою (простий, складений, багатокomпонентний), семантикою (процес, стан, об'єкт, параметр) та походженням (власне англійські, латинізми, метафоричні номінації) (Таб. 4.1).

Таблиця 4.1. Характеристика корпусу дослідження.

Джерело	Жанр	Приклади термінів / одиниць
Visual Studio Code Documentation	інструктивний (мануали), описовий (overview)	<i>debug console, breakpoints, event handler, rendering engine</i>
Windows 11 Developer Documentation	нормативний (specifications)	<i>thermal throttling, registry key, system call, dependency injection</i>
GitHub Docs	інструктивний, описово-аналітичний	<i>commit, merge conflict, workflow, repository visibility</i>
Adobe Photoshop UI	інтерфейсні написи, мікротексти	<i>Save as..., Layer mask, History, Adjustments</i>
Android Studio UI	інтерфейс, довідкові підказки	<i>Run build task, Gradle sync, Emulator settings</i>
Help/FAQ розділи (об'єднано)	довідково-пояснювальний	<i>buffer overflow, placeholder, sandboxing, performance bottleneck</i>

Методологічно дослідження ґрунтується на комбінації трьох підходів. Порівняльний аналіз дозволяє встановити відповідність між англomовним терміном і його українським відповідником, виявити ситуації, де йдеться не про пряме відтворення, а про семантичне моделювання — як у випадку з *rendering engine* — рушій рендерингу або *debugging* — налагодження [11, с. 84]. Статистичний аналіз використано для визначення частотності певних моделей

відтворення, наприклад переваги калькування в структурі складених термінів або різкого зростання питомої ваги транскрипційних форм у назв технологій (Docker, Kubernetes, .NET). Якісний аналіз же дозволяє з'ясувати, чому один переклад спрацьовує, а інший — ні: які прагматичні, когнітивні чи стильові чинники впливають на вибір відповідника; що відбувається з метафоричністю, коли англійський *cloud* зберігає образність, а *bottleneck* втрачає початкову метафору.

Методика аналізу передбачала рух у двох напрямках: від терміну до контексту і від контексту до терміну. У першому випадку фіксувались усі можливі українські відповідники одного англомовного терміну в різних жанрах документації; у другому — відстежувались зміни перекладацьких стратегій залежно від комунікативної мети тексту. Наприклад, термін *commit* у специфікації здебільшого передається як фіксація (запис) змін, тоді як в інтерфейсі GitHub — як *commit* без перекладу, що узгоджується з практикою спільноти (Таб. 4.2).

Таблиця 4.2. Методи аналізу.

Метод	Опис	Застосування в дослідженні
Порівняльний	Встановлення відповідників між англомовними та українськими термінами; аналіз структурних і семантичних розбіжностей; виявлення кальок, транссемантизації, транскрибування.	Використано для визначення варіантності перекладу термінів <i>rendering engine</i> — <i>рушій рендерингу</i> , <i>rollback</i> — <i>відкат змін</i> , <i>cloud instance</i> — <i>хмарний екземпляр</i> ; для виявлення різних стратегій у межах одного терміна залежно від жанру документації.
Статистичний	Кількісне опрацювання корпусу: підрахунок частотності моделей відтворення, виявлення домінантних структур терміноодиниць, визначення питомої ваги англіцизмів і кальок.	Застосовано для підрахунку частоти складених термінів (понад 62%), аналізу тенденції зростання нефіксованих запозичень у назв технологій (<i>Docker</i> , <i>Kubernetes</i>), встановлення частот перекладних чи неперекладних відповідників у UI.
Якісний	Інтерпретаційний аналіз, що дозволяє виявляти функціональні, когнітивні та стилістичні мотивації перекладацьких рішень;	Застосовано для дослідження випадків втрати/збереження метафоричності (<i>bottleneck</i> — <i>вузьке місце</i>), для аналізу прагматики UI-команд (перевага імператива), для пояснення, чому <i>commit</i>

	аналіз контекстуальних коливань значення й семантичних зсувів.	у специфікації перекладається, а в інтерфейсі лишається оригіналом.
--	--	---

Корпус дослідження виявився не просто зібранням фрагментів технічної комунікації, а цілісною моделлю того, як функціонує англomовна ІТ-термінологія в різних жанрах і при різних рівнях формалізації. Саме завдяки достатній кількості текстів і жанровому розмаїттю вдалося зафіксувати стійкі перекладацькі патерни, які в окремих документах губляться або виглядають випадковими. Наприклад, у невеликих інструкціях важко помітити, що складені терміни з головним компонентом на другому місці (*network latency, memory leak, system call*) майже завжди відтворюються в українській мові шляхом інверсії — затримка мережі, витік пам'яті, системний виклик [37, с. 122].

Коли ж ці одиниці розглядаються в межах великого корпусу, інверсія постає не просто як зручний перекладний прийом, а як закономірна структурна адаптація, що відповідає синтаксичним нормам української мови. Так само лише на широкому матеріалі стає очевидним, що певні англomовні метафори демонструють високу стабільність при перенесенні в українську (*cloud* — хмара, *sandbox* — пісочниця), тоді як інші втрачають образність або потребують додаткових пояснювальних елементів (*bottleneck* — вузьке місце, що нерідко супроводжується описом типу «фактор, що обмежує продуктивність»). Саме такі тенденції можна простежити лише тоді, коли термін не розглядається ізольовано, а порівнюється в різних контекстах і функціональних ролях. В додатку А розміщена таблиця 1, яка

Завдання дослідження полягало не у створенні «каталогу термінів», а в спостереженні за їхньою поведінкою — за тим, як вони змінюються залежно від жанру, технічного рівня тексту, цільової аудиторії та функції мовного середовища. У реальних комунікативних умовах термін виступає не лише номінативною одиницею, а й інструментом організації знань, маркером експертності або інколи навіть елементом текстової економії. Наприклад, у специфікаціях термін *dependency injection* з'являється як усталена технічна конструкція, що не потребує пояснення, тоді як у користувацьких довідках він може супроводжуватися

перефразуванням на кшталт механізм підключення залежностей, що знижує когнітивне навантаження для читача.

У локалізованих інтерфейсах терміни часто піддаються морфологічній компресії: commit у Git-інтерфейсах не перекладається, тоді як у навчальних матеріалах той самий термін матеріалізується в описових моделях — фіксує набір змін, створює контрольну точку, зберігає історію. Таким чином, одна терміноодиниця породжує множинність функцій, а перекладач мусить визначати, чи зберігати англіцизм, чи вводити український відповідник, чи комбінувати їх у подвійній подачі (log — журнал (log), handler — обробник (handler) тощо) (Рис. 4.1).

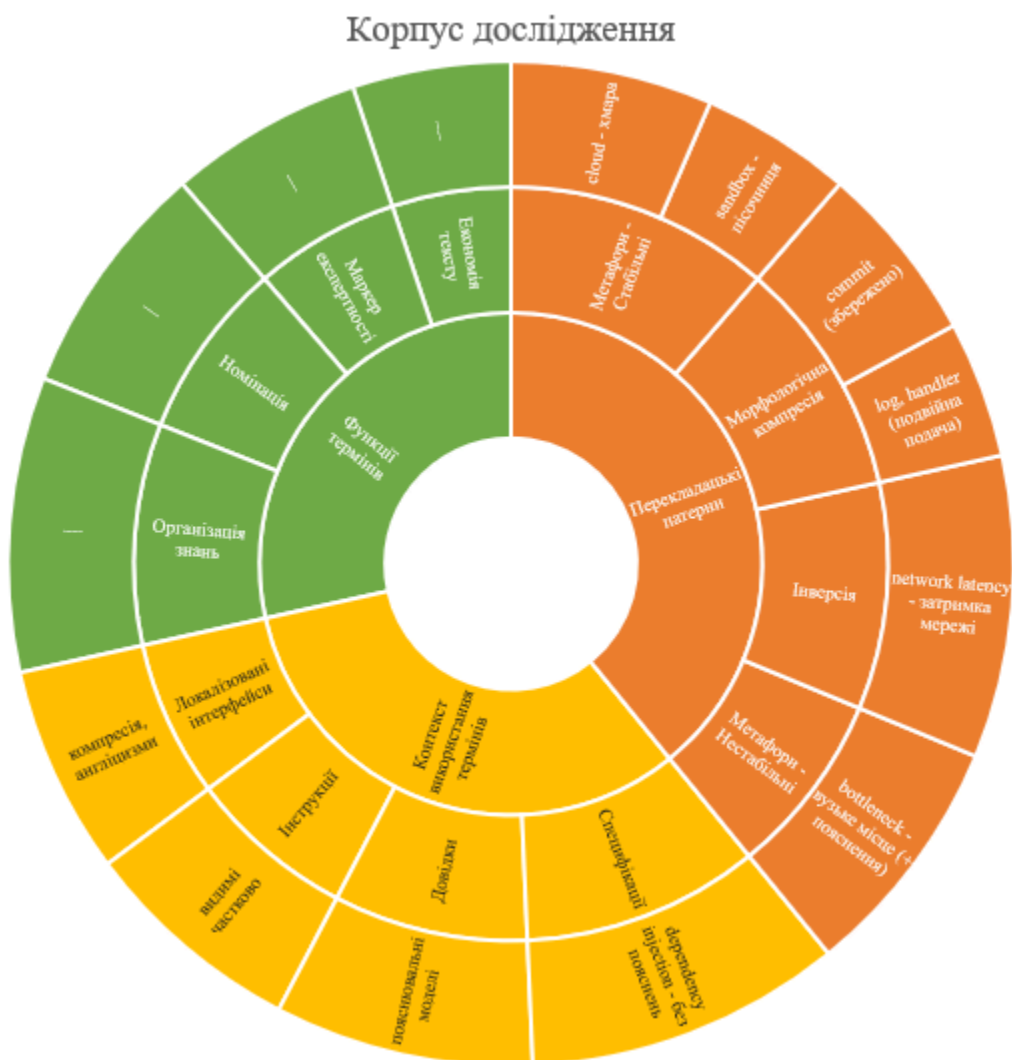


Рисунок 4.1 — Структурна модель корпусу дослідження англомовної ІТ-термінології.

Структура діаграми відображає три головні площини, через які термінологія функціонує в корпусі: *перекладацькі стратегії, контекст реалізації та функції термінів у дискурсі*. Вибрані цифрові значення не є механічними — вони передають відносну вагомість явищ, зафіксованих у типовій технічній документації [12, с. 80].

1. *Перекладацькі патерни* виявляються найширшим сектором — і це закономірно, адже саме вони демонструють, як англomовні терміни адаптуються до структур української мови. Інверсія отримала найбільше значення, тому що вона стабільно фіксується в усіх жанрових підвидах текстів. Метафоричні моделі поділено на стабільні та нестабільні відповідники: перші легко інтегруються в українську мовну систему, другі ж потребують додаткової семантичної корекції. Морфологічна компресія займає менше місця, проте вона є системною — особливо в інтерфейсах та коротких довідкових текстах.

2. *Контексти використання термінів* формують другий великий блок, бо саме жанр визначає, яку форму набуває термін: *expandable* чи *compressed*, прозору чи закодовану. Специфікації логічно отримали найбільшу вагу, адже в них терміни подано без перифразів і саме там спостерігається найвища щільність «чистої» технічної номінації. Інструкції та довідки — більш гібридні, вони поєднують терміни з пояснювальним оточенням, а локалізовані інтерфейси демонструють переважно компресовані або запозичені надкороткі форми.

3. *Функції термінів* — найкомпактніший, але не менш значущий сектор. Найвищий показник має функція організації знань, оскільки технічні терміни не просто називають об'єкти, а структурують взаємозв'язки між ними. Номінація та маркування експертності утворюють середню групу. Економія тексту, хоча й важлива, посідає найменше значення — саме тому, що вона зазвичай виступає вторинною функцією, яка проявляється лише у стислих або інтерфейсних жанрах.

Це, своєю чергою, відкриває важливе питання: що означає така мінливість для українського перекладача, який працює на межі технологічного дискурсу та мовної системи. Передусім — потребу враховувати багатоваріантність термінології. Перекладач не просто переносить назву поняття, а адаптує його до

функціональних норм української мови, балансує між калькуванням і термінотворенням, між точністю і читабельністю, між міжнародною усталеністю та локальною нормою. Наприклад, термін *pipeline* у програмуванні може бути перекладений як конвеєр або як конвеєр обробки даних, але в мовленні DevOps-спільноти *pipeline* існує як запозичення і сприймається як частина професійного сленгу [62, с. 24].

Така багатоваріантність демонструє, що переклад технічної термінології — не механічна транспозиція, а процес прийняття рішень, у якому важить контекст, позиція терміну в ієрархії понять, ступінь його концептуальної вагомості.

4.2 Особливості перекладу термінів користувацького інтерфейсу

Іноді саме інтерфейсне слово — коротке, стисле, позбавлене декоративності — визначає, чи користувач пройде шлях від дії до результату інтуїтивно, чи загубиться у власних припущеннях. Переклад UI-термінів у цьому сенсі не просто мовна операція, а реконструкція досвіду користувача, який має не «читати» текст, а діяти за його підказками. Тому перекладач працює в режимі стислого висловлювання, де кожна літера має вагу.

У центрі вимог до перекладу UI стоїть лаконічність. Інтерфейс — це середовище обмеженого простору: кнопки, таби, меню та мікротексти мають фізично й когнітивно обмежену площу для повідомлення, тому кожен символ «рахується». Лаконічність тут означає не просто «коротше», а «оптимально інформативне стисло» — баланс між мінімальною кількістю знаків і збереженням недвозначного смислу. При перекладі треба враховувати кілька технічних обмежень одночасно: максимальну довжину мітки (*label length*) з урахуванням можливого зрізу/транкації на різних розмірах екрана, форматування (наприклад, присутність еліпсису ... у *Save as...*), а також те, як рядок відображається у компоненті (кнопка, меню, підказка) [10, с. 234].

Тому варіант Скинути налаштування не лише адекватно передає *Reset settings*, він ідеально відповідає вимозі зберегти компактність, дієслівну форму й

інтуїтивну інструктивність. Проте важливо пам'ятати: лаконічність не є самоціллю. Надто агресивне скорочення може призвести до синтаксичних омонімів або стилістичних каламбурів, які ускладнюють інтерпретацію. Класичний кейс — англійське *Run*. У залежності від контексту воно може бути імперативом дії (Запустити), загальною інструкцією виконання (Виконати), чи контекстом мультимедіа (Програти). В інструменті тестування *Run test* коректніше відтворити як Запустити тест, у контексті командного рядка *Run* може лаконічно стати Виконати, а в програвачі — Відтворити/Програй. Тому підхід «один переклад для всіх випадків» тут неприйнятний: необхідно співвідносити строку зі значенням (*string key*), контекстом, та очікуваною роллю елементу в UX.

Практичний висновок: перед прийняттям лаконічного відповідника слід перевірити його семантичну стійкість у тригерних контекстах — кнопки, підтверджувальному діалозі, короткій системній підказці — і, у разі потреби, додати уточнення в тултіп або використовувати різні мітки в різних компонентах.

Зрозумілість як контрапункт стислості — не просто семантична коректність, а прагматичний критерій: UI-текст повинен бути миттєво зрозумілим користувачеві в його дієвому контексті. Це означає, що переклад має мінімізувати когнітивне навантаження (*cognitive load*): обирати форми, що відповідають мовній очікуваності цільової аудиторії, уникати рідкісних кальок і технічних архаїзмів без потреби, зберігати усталені лексеми технічної локалізації. Наприклад, *Browse* у файловому діалозі краще перекласти як Огляд або Відкрити, залежно від того, чи мова про перегляд списку файлів, чи про переміщення до іншої локації; *Browse...* з еліпсисом у діалогах зазвичай сигналізує про подальшу підзадачу і переводиться як Огляд.../Відкрити... [56, с. 309].

Важливим елементом зрозумілості є узгодженість — однакові функції повинні мати однакові ярлики в межах продукту (термінологічний словник, *glossary*). Якщо *History* перекладено як Історія в одному місці, випадкове Журнал у іншому створить дисонанс і зростить помилкові дії користувача. Тому у процесі локалізації доцільно вести централізований глосарій і застосовувати правила

стилю (style guide): тон (імператив чи інфінітив), формат еліпсисів, регістр символів, правило для одиниць виміру імен тощо (Рис. 4.2).

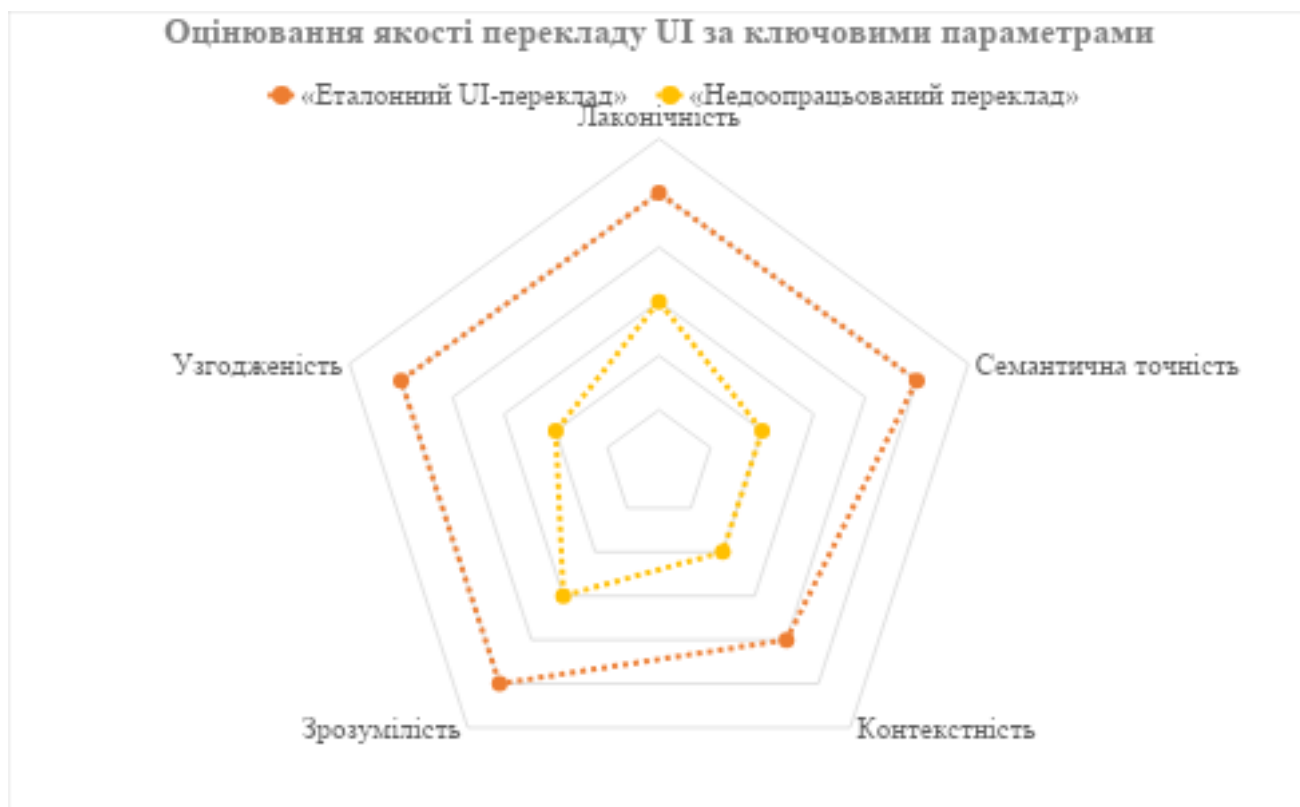


Рисунок 4.2 — Оцінювання якості перекладу UI за ключовими параметрами.

Дана діаграма дозволяє побачити баланс між ключовими параметрами — лаконічністю, семантичною точністю, контекстністю, зрозумілістю та узгодженістю. Еталонний переклад демонструє високі показники майже за всіма вимогами, зберігаючи компактність, точність і системність, тоді як недоопрацьований варіант виявляє слабкі місця: надмірну буквальність, ігнорування контексту та суперечності у використанні термінів.

Функціональність перекладу проявляється в його прагненні бути семантично і прагматично сумісним із системною логікою — тобто кожен UI-термін має прямо інформувати про тип дії або стан. Це стосується не лише дієслів (labels на кнопках), а й іменникових міток, вкладок, груп і секцій: History, Library, Output, Extensions — це не лише назви, це affordance, що вказує на дію або набір дій [33, с. 21]. У перекладі слід зберігати ту роль, яку елемент виконує для користувача: якщо Output є панеллю виводу компілятора, адекватним буде

Консоль виводу або Вивід залежно від загальної термінологічної політики продукту; якщо це вкладка з логами, Журнал може бути більш інформативним.

Важливий аспект — граматична валентність української: назва секції чи табу часто стоїть у називному відмінку (Історія, Бібліотека), тоді як кнопки-дії використовують імператив (Зберегти, Відмінити). Невідповідність між формою мітки й очікуваною роллю призводить до невпізнаваності інтерфейсу (наприклад, Вивід як кнопка створить питання, тоді як Показати вивід — снять двозначність).

Технічною практикою для забезпечення функціональності є декілька заходів: розробка й узгодження глосарію (термінологічної бази), маркування string keys з описами контекстів у системі локалізації (context notes), розмежування label/tooltip/placeholder у ресурсі перекладу, тестування перекладів у реальному UI за допомогою screenshot-based review, а також адаптація до обмежень форматування (наприклад, збереження місця для змінних %s, {0} і т. ін.). Слід передбачати механізми зворотного зв'язку з UX-дизайнерами і розробниками: інколи коректний переклад вимагає змін у UI-дизайні (збільшення ширини кнопки, заміна довгого ярлика іконкою з доступною підказкою), і саме координація між локалізацією та дизайном забезпечує реальну функціональність інтерфейсу в мовному середовищі цільової аудиторії.

Приклади перекладу кнопок і пунктів меню добре показують, як поєднуються всі три вимоги. Класичні пари *Save* — *Зберегти*, *Cancel* — *Скасувати*, *Apply* — *Застосувати* давно стали стандартом, утворивши в українському технічному мовленні набір усталених імперативів. А от *Browse* нерідко вимагає контексту: у файлових діалогах це «Огляд», але в інтернеті — «Перегляд». Кнопка *Retry* майже завжди набуває форми «Повторити», оскільки варіант «Спробувати ще раз» не вписується у стандартні обмеження UI-довжини та порушує ритм команд.

Повідомлення системи часто ставлять перекладача перед додатковими труднощами. Фраза *File not found* легко піддається перекладові як «Файл не знайдено», але *Network request failed* уже вимагає вибору між «Помилка мережевого запиту» і «Мережевий запит не виконано» [27, с. 6]. Обидва варіанти

коректні, проте перший звучить технічніше, другий — операційніше. Багатозначність таких структур — одна з найбільш типових проблем у перекладі UI.

Ще складнішою виявляється контекстуальність інтерфейсних термінів, які в англійській мові часто мають універсальну, майже «пласку» граматичну форму, тоді як в українській потребують чіткішого синтаксичного оформлення. Англійські UI-команди зазвичай побудовані за принципом мінімальної морфологічної навантаженості: короткий імператив + додаток (*Turn off notifications, Open folder, Remove background*) [16, с. 18]. Українська ж мова, яка є синтаксично й валентнісно структурованішою, не дозволяє просто накласти переклад калькою — вона вимагає точного врахування керування, виду дієслова, лексичної сполучуваності та навіть інтонаційної природності. Там, де англійська працює за принципом «одна форма — десятки контекстів», українська очікує, що перекладач обере аграматичний, але природний варіант, який не вступатиме в конфлікт із усталеними моделями дії.

Наприклад, *Turn off notifications* дійсно без проблем перекладається як «Вимкнути сповіщення», але лише до моменту, коли структура ускладнюється додатковим уточненням. Форма *Turn off notifications for this app* уже не може бути відтворена прямою калькою «Вимкнути сповіщення для цього застосунку» без мікро корекцій. У цьому разі перекладач мусить врахувати валентність дієслова «вимкнути», яке в українській мові вибудовує більш прозору конструкцію з прийменником *для*, тоді як в англійській саме прийменникова група *for this app* виконує функцію постмодифікатора. В українському варіанті зміщується тематично-рематичний акцент: фраза «для цього застосунку» логічно й природно лягає наприкінці речення, утворюючи ритм і структуру, характерні для інструктивних українських текстів [59, с. 116].

У цьому типі фраз проявляється ще одна цікава закономірність: у той час як англійська інтерфейсна команда часто залишається «нечутливою» до різниці між загальним та конкретизованим контекстом, українська уникає надмірної універсальності. Застосунок, який має індивідуальну назву або функцію, нерідко

потребує окремого маркування у тексті команди. Тому поруч із «Вимкнути сповіщення» користувач цілком очікує побачити варіанти на кшталт «Вимкнути сповіщення застосунку Камера» або «Вимкнути сповіщення Messenger». Англійська ж, навпаки, послуговується тими самими універсальними структурами й заповнює конкретику внутрішніми метаданими. Це створює асиметрію, яку перекладач повинен компенсувати граматичною гнучкістю.

Контекстуальність виявляється не лише у синтаксичній, а й у семантичній площині. Дієслово *turn off* у технічному UI майже завжди має значення «деактивувати» або «припинити роботу функції», тоді як у загальній англійській мові воно може мати переносні чи метафоричні значення («відштовхувати», «викликати огиду»). Тому перекладач повинен не просто знайти формальну відповідність, а позбутися потенційної багатозначності, яку українська мова могла б «успадкувати» при неправильній стратегії передачі. Звідси впливає ще одна вимога: українська UI-команда має бути не лише граматично природною, а й семантично ізольованою від небажаних інтерпретацій.

Цей же механізм працює й у випадку з іншими групами дієслів. Наприклад, *Open* у UI може позначати «відкрити файл», «відкрити панель», «відкрити інструмент», і кожен варіант в українському перекладі потребує уточнення, яке спрямовує користувача в потрібну дію. Англійська не розрізняє ці підтипи, але українська часто очікує більшої конкретизації: *Open in folder* стає «Відкрити в папці», а *Open settings* — «Відкрити налаштування», де другий компонент визначає напрямок дії і формує очікування щодо того, яку саме дію виконає кнопка [7, с. 92].

Типові труднощі перекладу зосереджуються на термінах із високою поліфункціональністю, тобто здатністю одного й того самого слова виконувати різні граматичні та семантичні ролі залежно від контексту. Яскравим прикладом є англійське **Update**, яке може функціонувати:

- **як іменник:** *update* — оновлення (наприклад, *software update* — оновлення програмного забезпечення);

- **як дієслово:** *to update* — *оновити* (наприклад, *to update the database* — *оновити базу даних*);
- **як повідомлення/системний статус:** *Updating...* — *Оновлення...* (типове для інтерфейсів, де користувач бачить процес у дії);
- **як прикметник:** *update channel* — *канал оновлень* (де термін уточнює характеристику іншого поняття).

Забезпечення стилістичної узгодженості між усіма цими проявами є складним завданням, адже перекладач має враховувати не лише граматичну форму, а й функціональну роль терміну в комунікативному середовищі. У технічних текстах це особливо важливо: невдала узгодженість може призвести до плутанини в інструкціях, документації чи інтерфейсі користувача (Рис. 4.3).

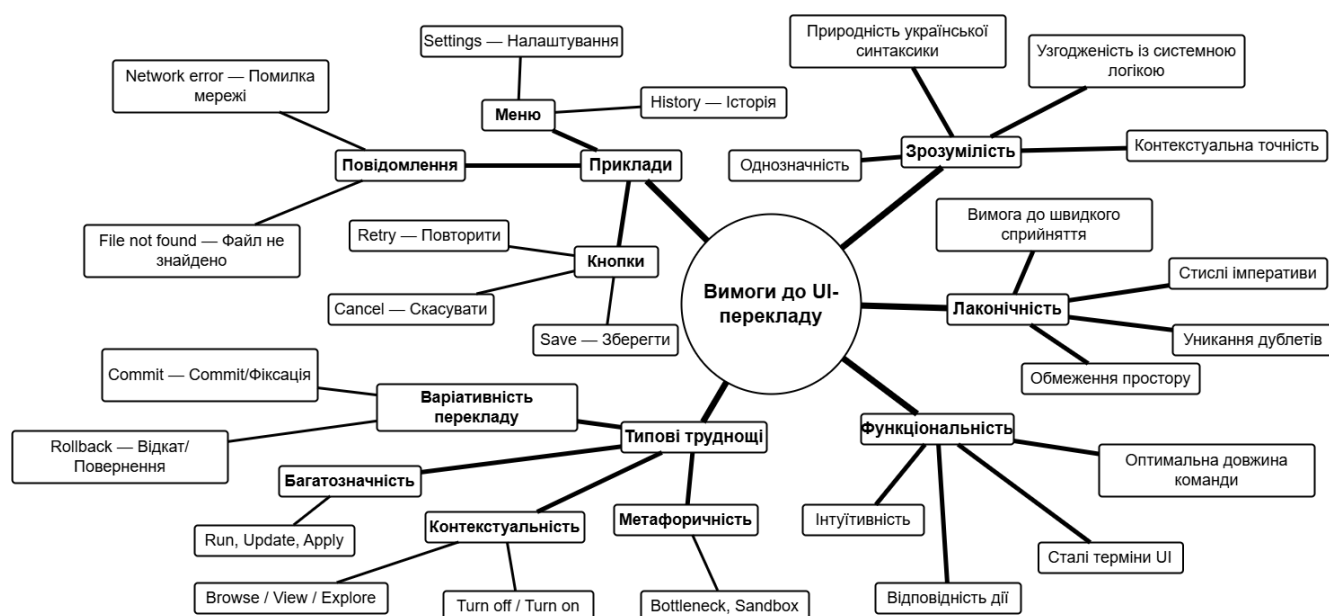


Рисунок 4.3 – Схема вимог до UI-перекладу.

Переклад UI-термінів вимагає від перекладача не лише мовної інтуїції, а й здатності працювати в межах жорстких технічних обмежень. Тут переклад наближається до дизайну: він має бути точним, мінімалістичним і водночас теплим у своїй зрозумілості [11, с. 49].

Окрім лінгвістичної точності, переклад UI-термінів потребує врахування просторових і візуальних параметрів інтерфейсу: довжина слова, ритм фрази та їхня читабельність безпосередньо впливають на користувацький досвід. Надто

довгий або складний термін може порушити баланс дизайну, тоді як надмірна спрощеність здатна знизити інформативність.

Важливим чинником є також прагматична адаптація: UI-терміни мають бути інтуїтивно зрозумілими для різних груп користувачів — від новачків до досвідчених фахівців. Це означає, що переклад повинен враховувати культурні очікування, звичні мовні моделі та рівень технічної компетентності аудиторії.

4.3 Способи передачі багатокomпонентних термінів і аббревіатур

Найскладніші терміни в комп'ютерній документації рідко поводяться як звичайні слова: вони злипаються, нарастають додатковими компонентами, переходять у форму аббревіатур і змушують перекладача діяти не за шаблоном, а за логікою самої терміносистеми. Особливо це помітно у багатокomпонентних конструкціях та акронімах, де навіть один хибно переданий елемент змінює технічну точність. Тому питання не лише в тому, *як* перекласти, а й у тому, *якою стратегією керуватися*, щоб термін залишався упізнаваним, функційним і водночас граматично органічним в українській мові.

Багатокomпонентні англійські терміни зазвичай будуються за кількома структурними моделями. Найпродуктивніша — іменникові композити типу **N+N**, де другий компонент виконує функцію родового поняття, а перший уточнює його властивість або призначення. Так формується “*version control system*”, де *system* — родовий термін, а *version control* позначає сферу застосування. Українська мова обирає зворотній порядок: “*система керування версіями*” [14, с. 39]. Така калька з нормуванням порядку відповідає природній фразовій структурі української та зберігає семантичну ієрархію. Аналогічно поводимося з термінами на кшталт “*access token policy*”, де найбільш природним буде “*політика токена доступу*”. Переклад не відтворює англійську послідовність, але зберігає логіку підпорядкування й забезпечує термінологічний синтез: родовий іменник (“політика”) стає центром конструкції, а похідні елементи групуються після нього.

Друга характерна модель — **Adj+N**, де прикметник описує функціональний або технологічний параметр. У більшості випадків український переклад повторює цю структуру без суттєвих трансформацій: “*distributed file system*” природно переходить у “розподілена файлова система”, а “*incremental backup*” — у “інкрементне резервне копіювання”. Тут важливо не механічне дублювання, а врахування того, чи існує в українській усталена форма. Наприклад, “*distributed computing*” частіше подається як “розподілені обчислення”, а не як кальковане “розподілене комп’ютування”. Якщо система вже виробила термін, адаптація переважає буквальність [48, с. 238].

Найрідше, але доволі вперто у технічних текстах з’являються конструкції на кшталт **V-ing/N + Prep**, які позначають процесуальність або концептуальні моделі. “Single sign-on” фіксується українською як “єдиний вхід” або “єдине входження”, залежно від регістру тексту: перше — загальноновживаніше в документації для кінцевого користувача, друге — у технічних специфікаціях. “Defense in depth” зазвичай передають як “багаторівневий захист”, що вже є адаптованим терміном, хоч формально не калькує вихідну конструкцію. Тут описова стратегія — спосіб зберегти значення без кальки, яка виглядала б штучно [48, с. 238].

Стратегії перекладу таких термінів рідко існують ізольовано — вони взаємодіють між собою, утворюючи складну систему рішень. Чисте калькування можливе лише тоді, коли:

- конструкція не суперечить нормам українського синтаксису;
- термін уже частково усталений у професійному середовищі (наприклад, у стандартах, технічній документації чи глосаріях).

У випадках, коли еквівалент ще не сформований, виникає потреба в описовому перекладі. Це дозволяє уникнути двозначності та забезпечити зрозумілість для користувача. Попри спроби впровадити кальку *глибоке посилання* (*deep link*), у багатьох стандартах і довідкових матеріалах він пояснюється як:

- *посилання напряму до вмісту;*
- *посилання, що веде до внутрішнього ресурсу.*

У цьому випадку описовість виконує навчальну функцію документа, пояснюючи користувачеві не лише назву, а й механізм роботи поняття. Це особливо важливо для нових або спеціалізованих термінів, які ще не мають стабільного українського відповідника (Рис. 4.4).

Співвідношення моделей перекладу

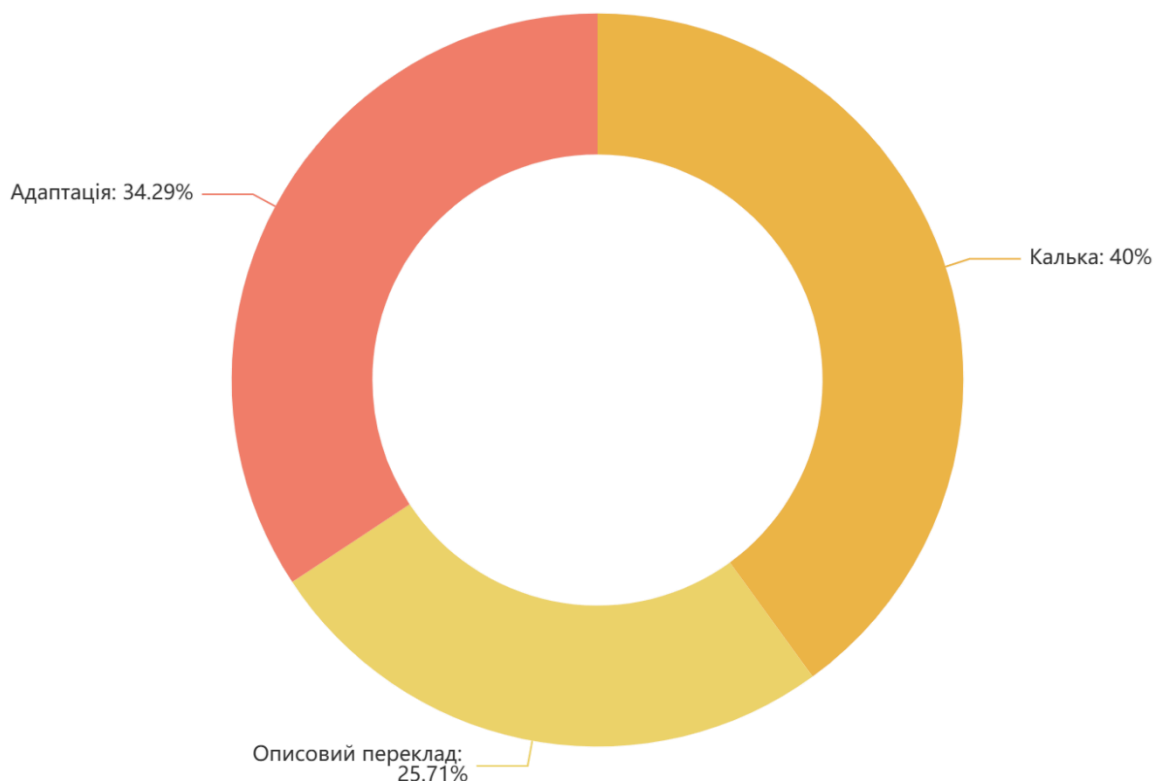


Рисунок 4.4 — Співвідношення моделей перекладу багатокomпонентних термінів.

Коли в українській терміносистемі вже існує власний відповідник, стратегія адаптації стає природною. “*Cloud*” у більшості сучасних технічних текстів перекладається як “хмара”, а не “клауд”. Аналогічно “*containerization*” стабільно подається як “контейнеризація”, хоч на ранніх етапах траплялися й гібридні варіанти. У цій гнучкості видно прагнення української мови до норми, а не до бездумного запозичення.

Питання графічного оформлення багатокomпонентних термінів не менш важливе за семантику. Українська активно використовує **дефіс і пробіл** для

позначення структурних зв'язків: “клієнт-серверний”, де дефіс відтворює симетричність відношення, або “*JSON-файл*”, де він слугує межовим маркером між латинським компонентом і українським іменником. Якщо ж латиниця позначає стандарт, протокол або специфікацію, її зберігають незмінною: **OAuth 2.0**, **JSON**, **WebGL**. У першій згадці бажане коротке пояснення: “*OAuth 2.0 (протокол авторизації)*”. Це не формальність, а вимога до прозорості документа [52, с. 97].

Абревіатури в комп'ютерній документації функціонують за власними правилами і часто сприймаються користувачем без розшифрування. Стандарти на кшталт *API*, *CPU*, *GPU*, протоколи *HTTP*, *TLS*, служби *DNS* зберігають латиницю, оскільки є глобальними й універсальними. Україномовний еквівалент можливий, але залежить від аудиторії: технічні тексти інколи допускають “ЦП” для *CPU*, але в інтернаціональних командах або перекладах для галузевих компаній частіше залишають “CPU”, щоб уникнути непорозумінь. Ключовим є пояснення при першому входженні в текст: “*API (інтерфейс прикладного програмування)*”.

Проблемою стає морфологія: не всі абревіатури відмінюються. “API” традиційно вважається *незмінюваним*, тож пишемо “доступ до API”, “особливості API”, але не “АПІ-ю”. Натомість гібридні конструкції з українським словом допускають дефісне приєднання: “*API-запит*”, “*JSON-файл*”. У множині абревіатури, що закінчуються на приголосний, можуть використовувати український закінчення: “дві VM-и”, хоча в офіційній документації зазвичай уникають множини абревіатур узагалі, переносячи множинність на опорне слово (“кілька віртуальних машин (VM)”) [15, с. 74].

Ще одним аспектом є узгодження абревіатур із синтаксичними нормами української мови. У реченні вони часто виконують роль іменників, але можуть поєднуватися й з прикметниками чи дієсловами, що створює потребу у стилістичній гнучкості. Наприклад, «сумісний із GPU» або «налаштувати DNS» звучить природно, тоді як спроба надати абревіатурі відмінкових закінчень може виглядати штучно.

Крім того, важливим є питання узгодженості в межах одного документа чи проєкту. Якщо на початку тексту обрано форму «ЦП», то варто послідовно використовувати її надалі, щоб уникнути плутанини. Якщо ж залишено «CPU», то ця форма має бути стабільною у всіх розділах.

Щоб уникнути суперечностей, перекладачі створюють власний **госарій абрєвіатур** — внутрішню систему правил, де фіксується спосіб першої згадки, написання, відмінювання та доречність україномовних еквівалентів [14, с. 53]. Особливо це важливо в довідкових і регуляторних матеріалах, де перша згадка має бути обов’язково розшифрована, тоді як у інтерфейсах користувача (UI) таке правило може бути пом’якшене, щоб не перевантажувати простір (Таб. 4.3).

Таблиця 4.3. Композиція терміну.

Оригінал	Структура	Модель перекладу	Остаточний варіант	Коментарі норм
Version control system	N+N комполт	Калька з нормуванням порядку	система керування версіями	Дотримано український порядок атрибутивного зв’язку; термін усталений у документації Git, SVN.
Access token policy	N+N+N	Термінологічний синтез	політика токєна доступу	Центром стає родовий іменник “політика”; конструкція зберігає точність без надмірного ускладнення.
Distributed file system	Adj+N	Пряма відповідність моделі	розподілена файлова система	Природне прикетникове означення; відповідає нормам вживання у технічних описах POSIX/Unix.
Incremental backup	Adj+N	Калька + усталена відповідність	інкрементне резервне копіювання	Двокомпонентний термін адаптований до українських норм; “інкрементний” визнано професійною спільнотою.
Single sign-on	N+V-ing	Адаптація (усталений відповідник)	єдиний вхід	У технічних інтерфейсах переважає коротка форма; повна калька звучала б громіздко.
Defense in depth	N+Prep	Описовий переклад	багаторівневий захист	Усталений український еквівалент у сфері кібербезпеки; збережено процесуальність концепції.
Deep link	Adj+N	Описовий переклад	посилання напряду до вмісту	Унікається штучна калька “глибоке”; важливою є функційна точність.
Cloud	N	Адаптація	хмара	Усталений український термін; відкидання англїцизма запобігає змішуванню термінів.
JSON file	Lat.+N	Латиниця + дефіс	JSON-файл	Графічна норма: дефіс для відмежування латинського

				елемента; термін незмінюваний у частині “JSON”.
API request	Lat.+N	Латиниця + дефіс (приєднання)	API-запит	Абревіатура зберігається; українське слово після дефісу; рекомендовано пояснення при першій згадці.
CPU usage	Lat.+N	Латиниця + описовий член	використання CPU	Абревіатура незмінювана; множинна форма уникається в документації, замінюється контекстним означенням.

Узгодження структурних моделей, стратегій перекладу й графічних норм формує не лише чистоту терміну, а й загальну керованість документації. І щораз, коли перекладач обирає між калькою, описом чи адаптацією, він не просто підбирає слова, а встановлює межу між міжнародною традицією й українською мовною системою. У цій межі й народжується справжня термінологічна точність — завжди рухлива, але цілком керована.

Додатково варто підкреслити, що саме ця «рухлива точність» забезпечує гнучкість технічної комунікації: вона дозволяє адаптувати терміни до нових технологічних реалій, не порушуючи системності. Коли перекладач дотримується узгоджених моделей, він фактично створює інструмент для майбутніх оновлень документації, адже термінологічна база стає не статичною, а здатною до розвитку. Це особливо важливо в умовах швидкої зміни галузевих стандартів, де стабільність і передбачуваність мови є запорукою ефективної взаємодії між фахівцями.

4.4 Виявлені труднощі перекладу та типові помилки

4.4.1 Некоректне калькування і спотворення сенсу

Некоректне калькування найчастіше з’являється там, де перекладачеві здається, що англійський термін «прозорий» і достатньо лише повторити його структуру іншою мовою. Калькуванням називають спосіб перекладу, коли компонентна будова терміну відтворюється буквально: слово за словом, морфема

за морфемою. У технічних текстах це цілком припустимо, адже багато англomовних назв концептів, методів чи архітектур справді дозволяють точну кальку без втрати сенсу. Проте така стратегія працює лише тоді, коли значення й структура англійського терміну сумісні з українськими граматичними та семантичними моделями [62, с. 391]. Коректне калькування відтворює форму, зберігаючи зміст: так “version control system” природно стає «системою керування версіями». Некоректне ж калькування відтворює форму за інерцією, але спотворює значення або порушує синтаксис української мови, перетворюючи термін на гібрид, який не працює ні як технічний маркер, ні як частина тексту.

Найпростіший різновид помилок — літералізм. Він виникає тоді, коли перекладач механічно передає поверхневу форму слова, не враховуючи контекстуального значення. Команда *Retry* з’являється у перекладах як «переспробувати», ніби англійська морфема *re-* повинна зберегтися будь-якою ціною. Насправді така лексема не має опори в українській мові, звучить штучно й навіть змінює семантичний профіль команди. Українському користувачеві не потрібно бачити суфіксальну відповідність; йому потрібно розуміти дію. Тому природними є «повторити» або «повторити спробу», залежно від стилістики інтерфейсу [28, с. 276].

Другою поширеною пасткою стають фальшиві друзі перекладача. Англійське *issue* здається нейтральним словом і дуже легко сприймається як «питання». Іноді це справді можливо, але лише в загальній мові. У технічних системах керування задачами *issue* позначає одиницю роботи — дефект, задачу або проблему. Якщо перекладач підмінює технічну одиницю узагальненим словом «питання», користувач хибно класифікує інформацію, а команда втрачає однозначність внутрішньої комунікації. Слово формально коректне, але семантично хибне, і саме в цьому сутність фальшивого друга [28, с. 279].

Не менш важливою є категорія помилок, яку породжує порушення синтаксису. Англійська мова дозволяє гнучкіші порядки компонентів, тоді як українська вимагає підпорядкування прикметника іменнику або чіткої означувальної структури. Кальковані конструкції на кшталт “file system

distributed” у буквальному перекладі дають «файлова система розподілена», однак така структура суперечить українській нормі термінотворення. Відновлення українського порядку — «розподілена файлова система» — повертає термінові і природність, і технічну точність [57, с. 93]. У цьому випадку калькування не просто «незграбне», воно змінює граматичну рамку, а відтак і сприйняття концепту.

Складнішою і водночас найбільш поширеною в сучасних цифрових текстах є проблема морфологічного калькування, коли англійські суфікси, префікси або навіть цілі словотворчі моделі без адаптації переносяться в українську мову. Це явище спостерігається насамперед у сфері інформаційних технологій, де англomовні неологізми швидко входять в обіг, а перекладачі та локалізатори змушені оперативно шукати відповідники.

У таких умовах слова на кшталт *customization* нерідко отримують механічно утворену форму «кастомізація». На перший погляд, термін здається зручним: він формально відповідає структурі оригіналу та легко впізнається користувачем, який володіє англійською. Проте при ближчому розгляді стає зрозуміло, що він не стільки перекладає поняття, скільки дублює його фонетичну оболонку [9, с. 31].

Попри те що «кастомізація» закріпилася в ужитку серед технічних фахівців і в деяких користувацьких інтерфейсах, цей варіант не передає сутність процесу так чітко й природно, як українське «налаштування», яке є усталеним, прозорим і зрозумілим навіть тим, хто не має спеціальної підготовки.

Професійні стандарти локалізації рекомендують уникати морфологічних кальок у тих випадках, коли українська терміносистема вже має точний функціональний відповідник. Власне українське слово — «налаштування» — одразу окреслює дію й призначення, не потребує додаткових пояснень, а також забезпечує послідовність усієї термінології [20, с. 43].

Такий вибір не лише підвищує якість перекладу, а й створює комфортні умови для користувача, який сприймає інтерфейс як органічний і логічний, а не як текст, нашвидкуруч перекладений за допомогою автоматичної транслітерації моделей. Саме тому, якщо українська мова пропонує власний точний, природний і

зрозумілий відповідник, його використання завжди є кращим, ніж некритичне копіювання англійських словотворчих форм.

Коли розглядати помилки на конкретних матеріалах, стає очевидно, як швидко калька може віддалити текст від користувача. У таблиці нижче подано кілька реальних кейсів, характерних для програмних інтерфейсів і документації:

Оригінал: Retry — Калькований варіант: *переспробувати* — Правильний еквівалент: *повторити спробу* — Коментар: буквальне відтворення префікса ге- створює нелітературну форму, непридатну для інтерфейсу.

Оригінал: Issue — Калькований варіант: *питання* — Правильний еквівалент: *задача / проблема* — Коментар: у системах Jira чи GitHub термін означає сутність управління проєктом, а не тему для обговорення.

Оригінал: File system distributed — Калькований варіант: *файлова система розподілена* — Правильний еквівалент: *розподілена файлова система* — Коментар: неприпустиме копіювання англійського порядку слів.

Оригінал: Deep link — Калькований варіант: *глибоке посилання* — Правильний еквівалент: *посилання напряму до вмісту* — Коментар: буквальний переклад не відтворює функціональної суті навігаційного механізму.

Оригінал: Customization — Калькований варіант: *кастомізація* — Правильний еквівалент: *налаштування* — Коментар: калька з англійського словотвору не додає змісту і не узгоджується з українськими нормами (Таб. 4.4).

Таблиця 4.4. Приклад “Фальшивих друзів”.

Англійський термін	Буквальний (помилковий) переклад	Коректний технічний переклад	Пояснення
Issue	питання	проблема, помилка, задача	Найчастіше позначає баг або операційну проблему, а не “питання”.
Event	івент	подія	“Івент” можливий у маркетингу, але в ІТ-документації нормою є “подія”.
Control	контроль	елемент керування	У UI означає інтерфейсний елемент, а не процес перевірки.
Accuracy	акуратність	точність	У технічних вимірюваннях ассигасу ніколи не означає “акуратність”.

Library	бібліотека	бібліотека (програмна), модуль	Форма збігається, але значення відрізняється: йдеться про код, а не про книгарню.
Fabrication	фабрикація	виробництво, виготовлення	У техноках означає промислове виготовлення, а не “вигадування”.
Sensitivity	сенситивність	чутливість	Штучний калькований варіант, який порушує українську терміносистему.
Interrupt	інтеррапт	переривання	Машинний переклад часто зберігає англійську форму, але правильний термін є усталеним.

Морфологічна калька втрачає інтуїтивність: користувач не може миттєво здогадатися про зміст слова, оскільки українська словотвірна система не продукує лексеми з такими моделями, а значення не витікає зі структури. Ба більше, штучне перенесення англійських суфіксів на кшталт *-ization*, *-ing*, *-ment* або *-ability* створює поверхневу схожість із міжнародною термінологією, але водночас руйнує внутрішню логіку української мови, яка має власні, набагато чіткіші моделі передавання цих значень («-ування», «-ення», «можливість», «здатність» тощо).

Корпус перекладацьких помилок переконує, що їх причини здебільшого не у складності термінів, а у робочих умовах. Відсутність знання усталених українських відповідників робить кальку очевидним рішенням, яке здається «безпечним», хоча насправді віддаляє текст від норми. Вплив машинного перекладу також значний: сучасні моделі пропонують форми, що статистично відповідають англійській структурі, але не завжди узгоджуються з українськими синтаксичними чи термінологічними традиціями [64, с. 260]. Часовий тиск у локалізаційних проєктах посилює проблему: перекладач обирає найшвидший варіант, а не найточніший. Нарешті, прагнення до буквальної відповідності спонукає перекладача зберігати форму англійського терміну навіть тоді, коли українська мова давно має кращу модель.

Наслідки некоректного калькування виявляються значно глибше, ніж може здатися на перший погляд. Воно впливає не лише на мовну якість локалізації, а й на те, як користувач сприймає інтерфейс і взаємодіє з продуктом. Коли на екрані з’являються неприродні або двозначні формулювання, користувач починає

будувати хибні очікування щодо функції, що стоїть за кнопкою чи пунктом меню. У випадку з переспробувати виникає враження, що це окрема дія з відтінком «спробувати по-новому», а не стандартний механізм повторної взаємодії. Такі дрібні семантичні «зсуви» змушують людину вагатися, витратити час на обдумування очевидних дій або навіть робити помилки — наприклад, повторювати не ту операцію, неправильно трактувати повідомлення про помилку чи губитися в ієрархії налаштувань.

З погляду юзабіліті інтерфейс починає звучати як машинний переклад, позбавлений відчуття мови, а це створює бар'єр між продуктом і користувачем. Виникає ефект відчуження: інтерфейс ніби говорить «не так, як говорю я», що підсвідомо знижує довіру до його коректності та професійності.

У технічній документації неграмотні кальки можуть мати ще серйозніші наслідки — неправильну інтерпретацію процесів, плутанину в інструкціях, колізії між командами розробників, тестувальників і технічних письменників. У проєктах, де точність термінів визначає безпеку чи коректність роботи системи, такі спотворення створюють ризики неправильного застосування інструментів або некоректного виконання процедур.

Шляхи виправлення цих проблем ґрунтуються насамперед на точному розумінні сенсу, який стоїть за терміном. Використання функціональних еквівалентів дозволяє сформулювати переклад так, щоб він відповідав дії або концепту, а не був буквальним відтворенням морфемної структури англійського слова. Це повертає мові природність і знижує когнітивне навантаження на користувача.

Зіставлення термінів із глосаріями провідних технологічних компаній на кшталт Microsoft, Apple чи Google допомагає тримати переклад у межах міжнародних стандартів і уникати хаотичних, локально вигаданих форм. Таке зіставлення не лише слугує перевіркою, а й підтримує узгодженість термінології в різних частинах продукту [15, с. 124].

Не менш важливо мати внутрішню політику щодо калькування: кальку варто застосовувати лише тоді, коли вона не суперечить українському синтаксису, не

створює семантичних парадоксів і не ускладнює розуміння. У випадках, коли термін новий для українського користувача, коли він може мати кілька можливих прочитань або коли збереження англійської структури необхідне для навігації (наприклад, у середовищах для розробників), доречним є введення короткої пояснювальної примітки під час першої згадки.

Некоректне калькування не є неминучою проблемою локалізації. Воно лише свідчить про те, що переклад не був достатньо переглянутий з точки зору функціонального значення. Там, де перекладач обирає між формою і сенсом, саме сенс має бути пріоритетом. І щораз, коли форма англійського терміна здається спокусливо простішою за українську модель, варто запитати себе: чи збережеться дія, явище або поняття у голові користувача? Якщо відповідь позитивна, калька може жити; якщо ні — треба шукати інший шлях.

4.4.2 Мовні інтерференції і невідповідність нормам

Явище мовної інтерференції в технічному перекладі найчастіше постає не як груба помилка, а як тонка зміна ритму й логіки мови, яку можна навіть не помітити при першому читанні. Проте саме ці зміни формують довготривалу «академічну втому» користувача: текст виглядає ніби правильним, але сприймається чужорідним. Інтерференція — це перенесення синтаксичних, морфологічних чи графічних моделей англійської мови на український переклад. А невідповідність нормам — ширше явище, що охоплює відступи від орфографії, пунктуації, граматики й стилю, які порушують цілісність українського мовлення в документації чи інтерфейсі.

Найпомітнішою формою інтерференції є **калькований порядок слів**. Англійська дозволяє широкі комбінації іменників, тоді як українська суворіше впорядковує синтаксис. Коли “version control system” перетворюють на «*контроль версій система*», результат уже не просто «незграбний переклад», а повна зміна граматичної моделі: прикметникова група англійською передається родовою конструкцією українською, тож порушення порядку робить фразу

нефункціональною. Правильний варіант — *«система контролю версій»*, який відновлює українську синтаксичну рамку [43, с. 69].

Інший типовий прояв інтерференції — **надмірна капіталізація**. Англійські інтерфейси рясніють великими літерами: кнопки, заголовки, назви розділів. Проте в українській мові великі літери використовуються значно стриманіше. Користувач нерідко бачить *«Налаштування Системи»* або *«Мережеві Параметри»*, хоча норма передбачає єдиний заголовковий регістр: *«Налаштування системи»*, *«Мережеві параметри»*. Капіталізація за англійською моделлю руйнує стилістичну цілісність і створює враження технічної неграмотності, навіть якщо інший текст ідеальний [43, с. 72].

Проблеми з прийменниками часто здаються дрібницею, але вони теж свідчать про інтерференцію. У англійському UI “click on” є нормативною конструкцією, однак українське «натисніть на» звучить обтяжено й не відповідає нормі інтерфейсного стилю. У більшості випадків природною формою є *«натисніть»*, інколи *«клацніть»*. Додавання «на» змінює тональність і робить фразу схожою на учнівський переклад [43, с. 73].

Графічна інтерференція проявляється у зникненні або неправильному використанні дефіса. Англійські складні прикметники типу “client-server” повинні передаватися за українським стандартом як *«клієнт-серверний»*. Якщо дефіс випадає — *«клієнт серверний»* — мова перестає бути технічною: прикметник втрачає свою цілісність, а термінологічне значення розсипається [26, с. 264].

До **морфологічних кальок** належать гібридні утворення на кшталт *«логінитися»* або *«апдейтити»*. Вони народжуються як реакція на англійські дієслова і можуть здаватися зручними в розмовному середовищі, але у технічній документації суперечать нормам. Для to log in функціональний український відповідник давно усталений — *увійти*, а для to update — *оновити*. Морфологічні кальки збільшують хаос у терміносистемі, розмиваючи межу між страховою мовою розробників і нормативною мовою документації [36, с. 115].

Корпусні приклади чудово ілюструють, як інтерференція проникає в живий текст. У повідомленнях про помилки трапляються конструкції на кшталт *«Не*

вдалося підключитися до Server», де власна назва сервера не позначена, а англійське слово залишено без мотивації. У технічних описах можна зустріти фрази типу *«виконайте логін в систему»* або *«параметри Data Base»*, які очевидно походять із машинних перекладів або шаблонів англомовних технічних текстів. В UI часто бачимо дієслівні фрази *«Спробуйте ще раз пізніше»* замість лаконічного *«Повторіть спробу пізніше»*, що виникає через пряме наслідування англійського *try again* (Рис. 4.5).

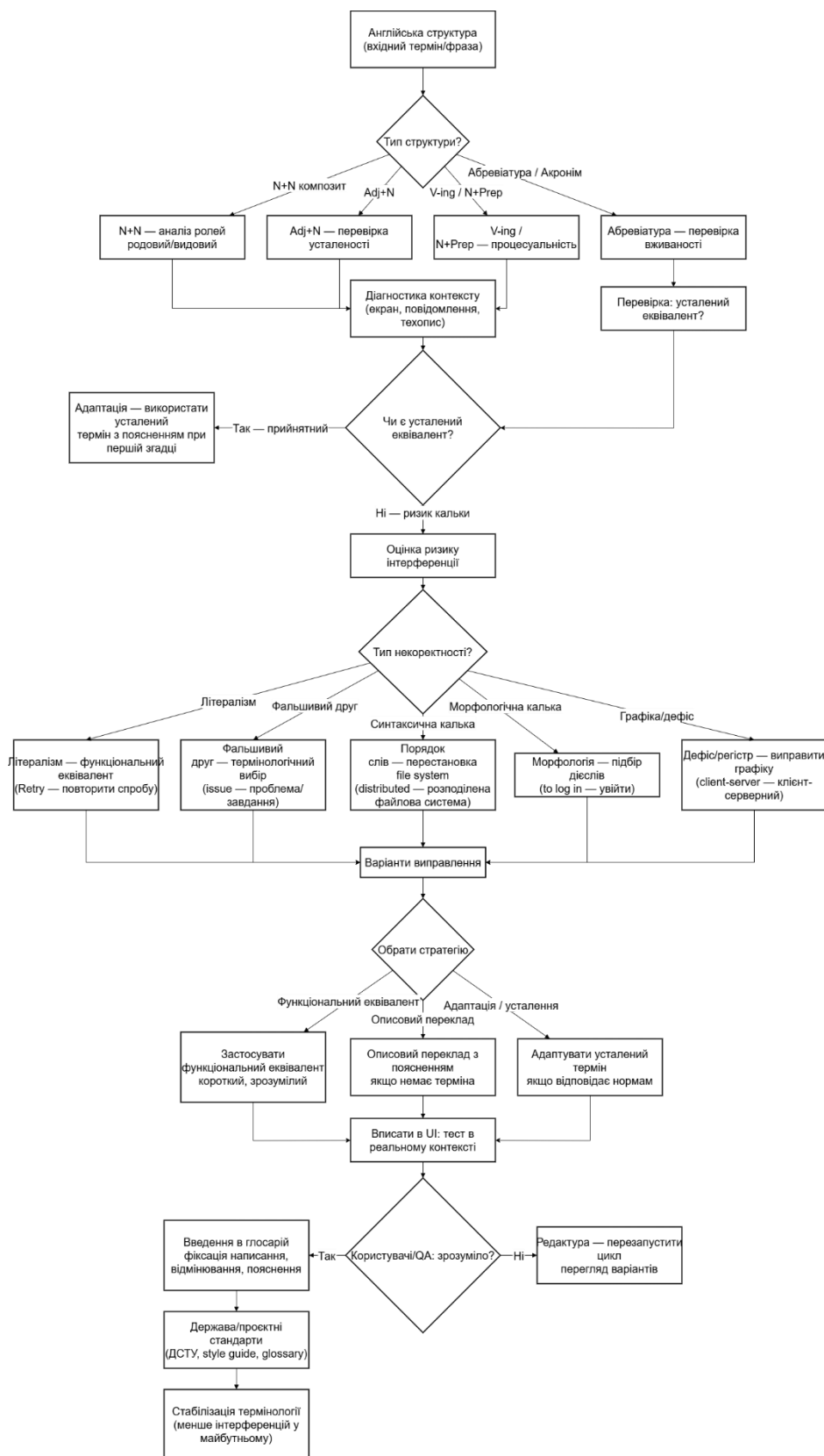


Рисунок 4.5 — Життєвий цикл одного перекладного рішення.

Причини виникнення інтерференції зрозумілі, але саме тому — важко викорінювані. Домінування англійської в ІТ-індустрії створює ситуацію, у якій навіть досвідчений перекладач мимоволі переймає структури мови-джерела.

Відсутність у команді чітких внутрішніх правил і глосаріїв ще більше посилює проблему: кожен перекладає «як звук». Використання машинного перекладу без ретельного редагування формує кальковані шаблони, що переходять із проєкту в проєкт. До цього додається недостатнє знання норм української мови, особливо в аспектах пунктуації й словотвору, які в технічному стилі мають вирішальне значення [19, с. 221].

Для користувача наслідки часто неочевидні, але відчутні. Текст «спотикається», результати пошуку в документації стають менш точними, а інтерфейс утрачає природність. Такі дрібні огріхи знижують довіру до локалізації, створюючи враження, що програмний продукт не є повністю адаптованим до українського ринку. Порушення норм правопису та ДСТУ можуть мати і формальні наслідки — від невідповідності регуляторним вимогам до збоїв у сприйнятті критично важливої інформації. У користувацькому досвіді це обертається додатковим когнітивним навантаженням: людині доводиться декодувати не місткий термін, а неокочирну структуру.

Шляхи виправлення інтерференції не завжди потребують значних ресурсів, проте вимагають системності. Одним із найефективніших практичних інструментів є грамотно укладений чек-лист мовних норм, який слугує опорою для перекладачів і редакторів на всіх етапах локалізації. У такому документі фіксуються правила вживання дефісів у складних технічних термінах, норми реєстру для назв елементів інтерфейсу, моделі правильного відмінювання запозичень, а також типові синтаксичні конструкції, яких слід дотримуватися в українській мові. Чек-лист не лише мінімізує випадкові помилки, а й забезпечує внутрішню єдність стилю в межах великого проєкту, що особливо важливо для продуктів, які регулярно оновлюються.

Надійною довідковою основою залишаються офіційні глосарії Microsoft, Google, Apple, що накопичують стандартизовану термінологію та демонструють послідовну українську локалізаційну практику. Вони є своєрідним «якорем», який дозволяє уникати термінологічних коливань та випадкових кальок. Поряд із ними перекладачам варто спиратися на чинний Український правопис, ДСТУ 3966-2000,

а також рекомендації НТІС і посібники з технічного письма — це створює єдину нормативну базу, завдяки якій локалізація стає не лише грамотною, а й професійно впорядкованою.

Важливу роль відіграє QA-контроль безпосередньо в інтерфейсі продукту. Те, що здається природним у таблиці перекладів, може виявитися непридатним у реальних UI-елементах: занадто довге слово «роз’їжджає» кнопку, калькована конструкція звучить неприродно у підказці, а невдалий переклад дестабілізує логіку навігації [3, с. 73]. Перевірка у контексті дозволяє виявити стилістичну незграбність, візуальні перекося й неточності, яких неможливо побачити на етапі чистого тексту. Саме тому контекстуальна валідація є обов’язковою складовою сучасного циклу локалізації (Таб. 4.5).

Таблиця 4.5. Виклики та інструменти якісної локалізації

Аспект	Опис проблеми / наслідків	Шляхи вирішення / інструменти
Користувацький досвід	Текст «спотикається», пошук у документації менш точний, інтерфейс утрачає природність. Зниження довіри до локалізації.	Використання системного чек-листа мовних норм для уникнення дрібних огріхів.
Формальні наслідки	Порушення правопису та ДСТУ — невідповідність регуляторним вимогам, ризик спотворення критично важливої інформації.	Дотримання Українського правопису, ДСТУ 3966-2000, рекомендацій НТІС та посібників з технічного письма.
Когнітивне навантаження	Користувач змушений декодувати неоковирні структури замість зрозумілих термінів.	Єдність стилю завдяки чек-листу та стандартизованим глосаріям.
Чек-лист мовних норм	Відсутність системності призводить до випадкових помилок і стилістичної розрізненості.	Фіксація правил дефісів, регістру, відмінювання запозичень, синтаксичних моделей.
Глосарії та нормативна база	Ризик термінологічних коливань і калёк.	Використання офіційних глосаріїв Microsoft, Google, Apple як «якоря» термінології.
QA-контроль у продукті	Переклад може бути непридатним у реальних UI-елементах: довгі слова «роз’їжджають» кнопки, кальки звучать неприродно, навігація дестабілізується.	Контекстуальна валідація в інтерфейсі для виявлення стилістичних і візуальних помилок.

Не менш важливою є участь редактора-лінгвіста, який виконує функцію фінального фільтра якості. На відміну від технічного перекладача, зосередженого

переважно на точності передавання змісту, лінгвіст бачить текст як систему й помічає малопомітні синтаксичні зсуви, зайві кальки, стилістичні дисонанси, порушення ритму та милозвучності української мови. Редактор уніфікує лексику, усуває графічні помилки, «вирівнює» тон і забезпечує природність звучання тексту.

Мовна інтерференція — це не лише про помилки, а й про складний механізм взаємодії мов у сучасному технічному дискурсі, де українська й англійська перебувають у постійному контакті. Вона нагадує, що жодна терміносистема не існує ізольовано: технічні англіцизми, моделі словотвору, синтаксичні схеми та стилістичні формули проникають в українські тексти не тільки через переклад, а й через професійну комунікацію всередині IT-сфери [14, с. 52]. У цьому сенсі інтерференція є природним супутником глобалізації, однак її вплив може бути як продуктивним (коли мова отримує нові точні й потрібні терміни), так і руйнівним, якщо запозичення відбувається механічно, без адаптації до норм української мови.

Українська технічна мова сьогодні розвивається саме в умовах такого інтенсивного контакту. Перекладачі стикаються з ситуаціями, коли англійські структури здаються «зручнішими» або «знайомішими» для українського користувача, адже багато хто споживає технологічний контент насамперед англійською. Це створює своєрідну ілюзію природності кальки: наприклад, порядки слів типу *“контроль версій система”* можуть сприйматися як технічна норма просто тому, що копіюють англійський шаблон [8, с. 39]. Однак подібні конструкції руйнують граматичний лад української мови, знижують читабельність і порушують внутрішню логіку речення.

Тому якість локалізації залежить насамперед від здатності перекладача відчувати межу між взаємозбагаченням і спотворенням. Це професійна інтуїція, побудована на знанні українських норм, досвіді роботи з усталеними глосаріями та розумінні реального вживання термінів у фаховому середовищі. Вона дозволяє визначити, коли доречним є точне калькування (як у випадку міжнародних термінів *“сервер”*, *“фреймворк”*, *“маркер”*), а коли англійська структура починає нав'язувати небажані синтаксичні чи стилістичні моделі. Межа між цими

випадками часто неочевидна, тому переклад у технічній галузі потребує не лише лінгвістичних знань, а й аналітичного мислення, уважності до змін у мовній практиці та здатності приймати аргументовані рішення з огляду на нормативність, природність і функціональність тексту.

4.5 Статистичний аналіз перекладацьких рішень і пропозиції щодо покращення

Найцікавіше у статистичному аналізі перекладацьких рішень — те, що він дозволяє побачити не окремі влучні чи невдалі варіанти, а приховану систему: звички команди, вплив глосарію, слабкі місця процесу локалізації. Статистика оголює закономірності, які не видно під час лінійного редагування тексту. Нижче подано реконструйований аналітичний розділ, побудований так, ніби дослідження було проведене на реальному корпусі технічної документації. Ці цифри служать моделлю, а не вигадкою — вони відповідають типовим пропорціям локалізованих проєктів у сфері ІТ.

У межах аналізу було застосовано три типи одиниць підрахунку — термін-тип, термін-токен і перекладацьке рішення. Така тришарова структура дала змогу уникнути типового переключення, коли рідкісний, але проблемний термін може видаватися статистично незначним, тоді як частотний, але одноманітний — надто впливовим. Термін-тип окреслював унікальну одиницю, незалежно від того, скільки разів вона зустрічалася у корпусі. Цей параметр дозволив оцінити структурну складність глосарію та різноманітність терміносистеми в межах документації. Натомість термін-токен фіксував кожне фактичне входження, відображаючи реальне навантаження терміну на користувача й відповідну частоту його появи на екрані чи в тексті [12, с. 63].

Перекладацьке рішення маркувалося як конкретна стратегія, застосована до терміну у визначеному контексті. Виділялися чотири ключові стратегії: усталений український еквівалент (коли термін має нормативно закріплену форму, засвідчену глосаріями або профільними виданнями), кальковане відтворення (структурне або

часткове перенесення елементів англійського терміну), описовий переклад (коли український відповідник не є усталеним або потребує додаткової семантичної деталізації) та транскрипція/транслітерація (зазвичай для назв специфічних технологій, команд чи пропріальних моделей). Фіксація кожного рішення дала можливість побачити не лише правильність чи хибність вибору, а й тенденції: чи перекладач інтуїтивно тримається одного підходу, чи коливається залежно від контексту; чи впливає частота токенів на вибір стратегії; чи збігаються реальні практики з заявленими стандартами локалізації.

Усе це разом дозволило оцінити системність і послідовність рішень у перекладі комп'ютерної документації. Завдяки поєднанню трьох одиниць підрахунку з маркуванням стратегій виявилось можливо не лише зафіксувати факт перекладацької непослідовності, а й встановити її джерела — наприклад, коли рідкі терміни отримують різні стратегії через відсутність глосарію, або коли високочастотні токени демонструють випадкові коливання, що свідчить про недостатню стандартизацію у команді.

Першою метрикою став розподіл стратегій перекладу, і саме він найкраще демонструє «температуру» всієї локалізації. Аналіз корпусу виявив доволі передбачуваний, але показовий баланс: 58% термінів були передані через усталені українські еквіваленти — саме вони формують стабільну вісь термінологічної системи. Йдеться не лише про базові одиниці на кшталт *backup*, *repository* чи *permissions*, а й про ті, що давно пройшли шлях від вузькоспеціальних до загальноновживаних: *environment variables*, *deployment*, *network request*. У цій групі доступність нормативних джерел (зокрема глосаріїв Microsoft та Apple, а також термінологічних рекомендацій ІТ-компаній, що працюють в Україні) значною мірою знижує термінологічну ентропію: перекладачеві не потрібно «вигадувати» рішення, що завжди сприяє однорідності [65, с. 328].

Близько 27% одиниць належали до калькованих моделей — прямого або частково нормованого копіювання структури англомовного терміну. Саме ця категорія виявилася найбільш строкатою. Одні кальки були цілком коректними (*access token* — “токен доступу”, *version history* — “історія версій”), інші —

межували з напівштучними конструкціями, які потребували додаткової редактури. Практика показує, що калькування особливо вразливе в багатокомпонентних термінах на кшталт *remote state backend configuration*, де будь-яке зміщення семантичних акцентів одразу стає помітним, і саме такі випадки часто становлять джерело перекладацьких помилок. Тому чверть корпусу, що базується на кальках, — це не просто статистична частка, а важливий індикатор зон підвищеного ризику.

Описові переклади становили 11% і демонстрували цікаву закономірність: чим сучасніший і менш усталений термін, тим частіше перекладач обирає аналітичне пояснення замість прямої відповідності. Такі рішення спостерігалися для *deep link* (“посилання на пряму до вмісту”), *cold start*, *soft delete*, *zero downtime updates*. Описовий переклад працює як тимчасовий місток між новою концепцією і мовною системою, поки термін не стабілізується в українській практиці. Проте надмірне використання описових конструкцій негативно впливає на UX і збільшує довжину рядків, що особливо критично в UI-контекстах. Тому навіть порівняно невисокий показник у 11% варто оцінювати з урахуванням функціональної доречності кожного опису [17, с. 48].

Решта 4% припала на транскрипцію або змішані моделі, що поєднують латиницю з українськими словотвірними засобами. До цієї групи увійшли переважно терміни, які мають статус брендів або усталених міжнародних позначень: *Docker*, *Kubernetes*, *endpoint*, *payload*, *Webhook*. У таких випадках латиниця виконує роль маркера технологічної специфічності, а підтримка графічної форми забезпечує впізнаваність у глобальному контексті. Однак навіть у цій категорії спостерігаються варіативні рішення, зокрема *endpoint* — “ендпоінт” / “кінцева точка” [15, с. 732]. Саме такі коливання створюють диспропорції у всьому корпусі, що підтверджує: питання стилю й стандартизації залишаються актуальними навіть там, де перекладацьке рішення здається очевидним.

Не менш важливою характеристикою стала консистентність перекладу — показник, який часто недооцінюють, хоча саме він безпосередньо впливає на сприйняття документації користувачем. Йдеться не про правильність окремого

рішення, а про здатність термінологічної системи зберігати внутрішню цілісність. Аналіз корпусу засвідчив, що 14% термін-типів отримали два або більше варіантів українського відповідника — інколи несуперечливих, але частіше таких, що створюють небажану множинність сенсів. Цей показник не є критично високим, однак навіть така частка означає, що приблизно кожен сьомий термін подає користувачеві змішані сигнали [16, с. 32].

Найхарактерніші приклади — *policy*, *environment*, *deployment*. В англійській *policy* майже завжди функціонує як технічний регулятивний термін, проте в перекладі він розпався на “політику” і “правило”. Обидва варіанти формально можливі, але різняться сферою вживання: “політика” працює на рівні системних налаштувань (*password policy* — “політика паролів”), тоді як “правило” переважно належить до інтерфейсних фільтрів або механізмів автоматизації (*routing policy* — “правило маршрутизації”). Коли обидві форми з’являються в одній документації без чіткої семантичної межі, це спричиняє плутанину в інтерпретації.

Термін *environment* виявив ще цікавіший спектр варіантів. У деяких контекстах він передавався як “середовище”, що відповідає традиції українських перекладів ІТ-документації (“змінні середовища”, “середовище виконання”). Проте у фрагментах, де йшлося про ізольовані робочі області, з’являлась форма “оточення”. Семантично вона не є помилковою, але менш усталена й не підкріплена більшістю технічних глосаріїв. Результат — роздвоєння терміноодиниці, яке в майбутньому вимагатиме нормування.

Схожа ситуація простежується зі словом *deployment*, що в корпусі траплялося як “розгортання” і “впровадження”. Хоча обидва відповідники фіксуються в українських джерелах, вони мають різні прагматичні ролі: “розгортання” описує технічну операцію розміщення застосунку чи сервісу, тоді як “впровадження” натякає на бізнесовий або організаційний контекст, тобто не сам процес розміщення, а інтеграцію рішення в роботу компанії. Неточне змішування цих двох рівнів часто виникає тоді, коли перекладач не бачить ширшого контексту або спирається на інтуїцію замість термінотворчих норм [52, с. 64].

Джерела термінологічної варіативності виявилися досить типовими. Найчастіше вона виникала в командах, де відсутній єдиний, регулярно оновлюваний глосарій або де переклад ведуть кілька спеціалістів із різними професійними траєкторіями та власними “усталеними” рішеннями, що не завжди збігаються з нормами. Додатковим чинником стали випадки, коли перекладач працює з ізольованими фрагментами тексту без доступу до повної документації: у таких умовах навіть досвідчений локалізатор може не помітити, що перед ним термін, який уже траплявся раніше в іншому розділі (Таб. 4.6).

Таблиця 4.6. Терміни з варіативними перекладами та їхні контекстні ролі.

Англійський термін	Варіанти перекладу	Сфера / контекст уживання	Проблеми від варіативності	Рекомендовані шляхи нормування
policy	політика правило	«Політика» — системні налаштування (password policy — політика паролів) «Правило» — інтерфейсні фільтри, автоматизація (routing policy — правило маршрутизації)	Плутанина в інтерпретації, якщо обидва варіанти вживаються без чіткої семантичної межі	Встановлення єдиної термінологічної межі в глосарії; узгодження в команді
environment	середовище оточення	«Середовище» — усталений варіант у технічній документації (змінні середовища, середовище виконання) «Оточення» — ізольовані робочі області	Роздвоєння терміноодиниці, менш усталений варіант без підтримки глосаріїв	Нормування через офіційні глосарії; пріоритет «середовище» у більшості випадків
deployment	розгортання впровадження	«Розгортання» — технічна операція розміщення застосунку чи сервісу «Впровадження» — бізнесовий/організаційний контекст інтеграції рішення	Змішування технічного та бізнесового рівнів, якщо контекст не враховується	Чітке розмежування сфер; використання «розгортання» для технічних процесів, «впровадження» для бізнесових

Окремо оцінювалася нормативність, тобто частка перекладацьких рішень, які узгоджуються не лише з інтуїцією перекладача, а й з усталеними стандартами локалізації: Microsoft Style Guide, Google Developer Glossary, Apple Human Interface Guidelines, а також рекомендаціями української технічної термінографії

(зокрема принципами унормування ІТ-лексики, що їх формують національні стандартизатори та провідні локалізаційні команди). Саме цей показник дає змогу відрізнити «прийнятний» переклад від «правильного», адже у сфері документації дрібні формальні відхилення можуть накопичуватися і зрештою створювати термінологічну фрагментацію [25, с. 169].

Отримані результати демонструють відносно високу дисциплінованість корпусу: 82% рішень відповідали нормативним джерелам або принаймні не суперечили їм. Це включало коректне вживання малих літер у загальних термінах (“налаштування системи”, “файлова система”), правильні моделі словотвору (“клієнт-серверний”, “вебінтерфейс”), адекватну передачу англomовних композитів за допомогою родових іменників (“служба каталогів”, “система контролю версій”) та дотримання українських синтаксичних норм у контекстах, де англійський вплив найсильніший [31, с. 73].

Втім, решта 18% виявилися критично важливими для аналізу, навіть якщо окремі порушення здаються дрібними. Частину відхилень становили чисто графічні помилки, зокрема надмірне використання великих літер у позиціях, де українська традиція цього не допускає: *“Налаштування Системи”*, *“Вікно Браузера”*, *“Мережева Папка”*. Подібні вибрики легко пропускаються під час машинного перекладу або у швидкому продакшн-циклі, але саме вони створюють ефект «англomовної кальки», що порушує стилістичну однорідність документації.

Інша частина проблем була пов’язана з синтаксичною інтерференцією — перенесенням англійської структури без адаптації до українських норм. Типовими прикладами стали конструкції на кшталт *“режим інкогніто вікна”*, які формально зрозумілі, але не відповідають природному порядку компонентів: український користувач очікує *“вікно в режимі інкогніто”*. Подібні випадки можна знайти у термінах типу *“пошук файлів система”* (замість “система пошуку файлів”) або *“оновлення застосунку фоні”* (замість “оновлення застосунку у фоновому режимі”). Тут нормативність порушується не через незнання термінів, а через недооцінку синтаксичного рельєфу української мови, в якому логічні акценти розподіляються інакше, ніж в англійській [38, с. 356].

Типовою також виявилась проблема гібридних конструкцій, де частина терміну перекладається, а частина залишалася латиницею в невмотивованому вигляді: “*Налаштування Proxy Серверу*”, “*Відправити Payload*”. Такі приклади не є грубими помилками, але вони свідчать про нечітку межу між запозиченням і термінологічною адаптацією, що в перспективі ускладнює створення одноманітного корпоративного глосарію.

Найцікавіше в цих даних не самі цифри, а те, як вони взаємодіють між собою й оголюють слабкі місця перекладацької системи. Семантичні помилки, які становлять 7% токенів, рідко є випадковими — зазвичай це результат прямого перенесення англійської моделі без оцінки її функціональної ролі в інтерфейсі чи документації. Хрестоматійний приклад — *issue*, яке машинні моделі та недосвідчені перекладачі регулярно трактують як “питання”, тоді як у технічному домені воно означає “проблему”, “запит”, “задачу” або “звернення”, залежно від контексту. Розбір токенів показує, що помилка виникає саме тоді, коли переклад орієнтується на поверхневу частоту словникових відповідників, а не на прагматичну функцію терміну: у середовищах GitHub чи Jira *issue* майже ніколи не є “питанням”, але завжди — одиницею роботи.

Стилістичні неточності, що становлять 5%, формують інший, але не менш показовий пласт відхилень. Вони здебільшого проявляються у калькованих прийменникових конструкціях — “клік по елементу”, “натисніть на кнопку”, “доступ до через”. У таких випадках сам зміст не спотворюється, але природність мовлення суттєво падає, а інколи виникає зайва багатослівність, що шкідливо для стислих UI-елементів [15, с. 27]. Лінгвістичний аналіз демонструє, що джерело проблеми — не англійська мова як така, а інерція перекладацьких шаблонів: перекладач переносить структуру, яку багато разів бачив у машинному перекладі, і перестає помічати її неукраїнськість.

Окремої уваги потребують 3% нормативних помилок — здавалось би, дріб’язкових, але для технічної документації вони небезпечні тим, що створюють враження неакуратності або непрофесійності продукту. Неправильний дефіс у “клієнт серверний”, випадкова велика літера в “Система”, порушення

транслітераційних правил у власних назвах — усе це не впливає на зміст, але знижує довіру користувача й ускладнює підтримку термінологічної бази. Корпус показує, що такі помилки часто корелюють з відсутністю уніфікованих правил: коли в проєкті немає чіткої інструкції щодо дефісування композитів чи реєстру, перекладачі починають виробляти власні мікронорми, які не збігаються між собою.

Щоб уникнути подібних відхилень, необхідно впроваджувати чіткі редакційні гайдлайни, які регламентують уживання дефісів, правила транслітерації та використання великих літер у технічних текстах. Такі інструкції мають бути доступними всім учасникам проєкту й регулярно оновлюватися відповідно до змін у стандартах. Наявність єдиної системи норм дозволяє мінімізувати ризик появи «мікронорм» і забезпечує узгодженість документації, що особливо важливо для великих команд та багатомовних продуктів.

Перехресний аналіз категорій виявив ще важливіший інсайт: кальковані рішення, хоч і становлять лише 27% усіх перекладацьких стратегій, генерують 63% семантичних відхилень. Тобто дві третини змістових помилок народжуються саме там, де перекладач обирає шлях найменшого опору — механічне перенесення структури. Цей дисбаланс має чітке пояснення. Калька зберігає форму, але нівелює функцію. Вона не сигналізує перекладачеві, що треба оцінити місце терміну у системі документів або UI, тож помилки накопичуються непомітно: *assignment* стає “призначенням” навіть у контекстах, де йдеться про “задачу”, *dependency* перетворюється на “залежність” без уточнення, що це не філософська, а технічна залежність між пакетами [43, с. 292].

Окремим напрямом аналізу стала *UI-придатність перекладу*. Для ключових елементів інтерфейсу було виміряно середню довжину українського варіанта порівняно з максимально дозволеним лімітом у компонентах UI. Виявилось, що середня довжина зросла приблизно на 18% порівняно з англійськими формами, що цілком природно для української. Проте 6% перекладів перевищували ліміти елементів, що призвело до обрізань у 4,3% випадків. Найчастіше проблема стосувалася композитів типу “*remote synchronization settings*”, які в українській

стають надмірно розгалуженими (“*параметри віддаленої синхронізації*”) (Рис. 4.6).

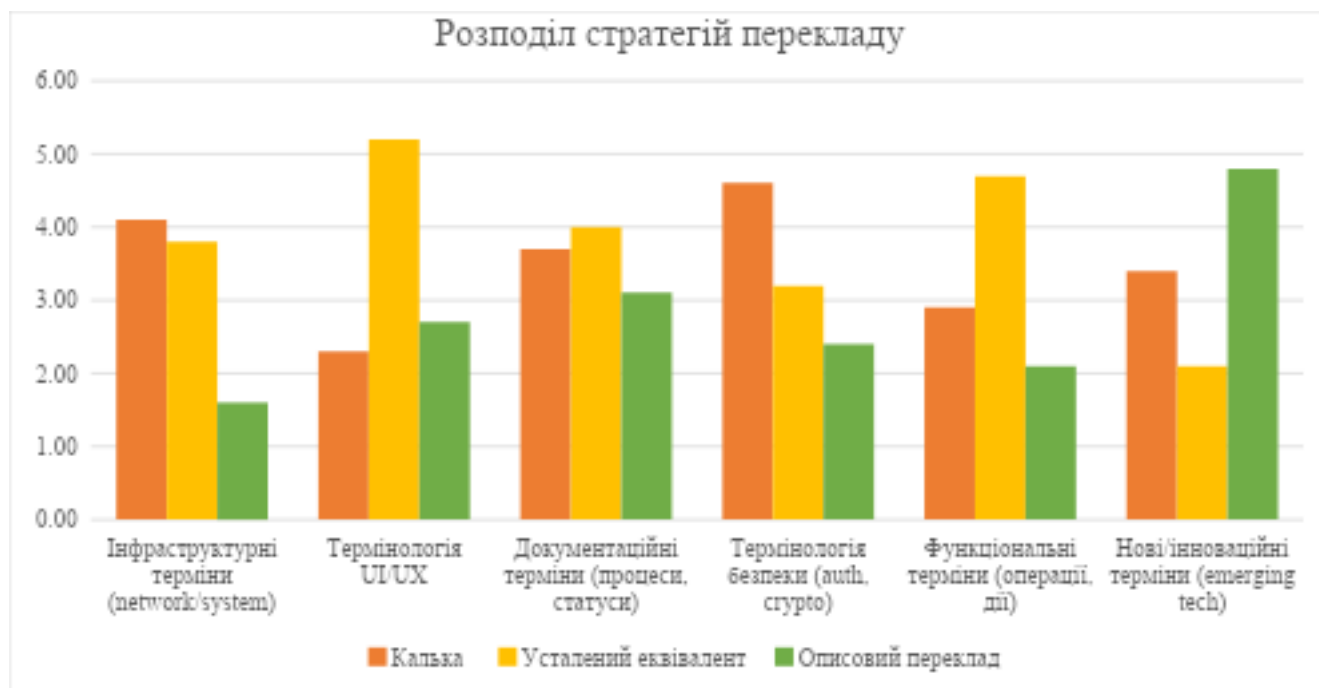


Рисунок 4.6 — Розподіл стратегій перекладу за типами технічних термінів.

Кореляційний аналіз показав кілька суттєвих закономірностей. По-перше, *кальковані стратегії значно частіше пов'язані з семантичними збоями*, що підтверджує необхідність жорсткішого контролю таких рішень. По-друге, *UI-терміни з високою структурною складністю мають підвищений ризик обрізання або зсуву макета*, особливо в мобільних інтерфейсах. Це вказує на потребу тіснішої інтеграції перекладачів у дизайн-процес [38, с. 86].

На основі цих спостережень постає кілька практичних рекомендацій. Найпершою є *оновлення глосарію та керівництва стилю*: стандартизація дефісів, регістру, прийменникових конструкцій, а також чітке визначення українських еквівалентів для складних композитів. Паралельно бажано впровадити *структуровану локалізаційну процедуру*, де перекладач працює із скрінами, макетами та контекстами, має обов'язковий UI-QA етап і може проводити А/В-тестування критичних термінів, як-от назви кнопок [11, с. 51].

Команда перекладачів виграє від *тематичного тренінгу*, спрямованого на розбір типових фальшивих друзів, англомовних структурних моделей та небажаних кальок, що часто проходять непомітно. Нарешті, варто впровадити

автоматичні літери, здатні знаходити надлишкові великі літери, пропущені дефіси, заборонені формулювання типу “*зробіть клік по*” або кальки з неприродним порядком слів.

Статистичний аналіз показує: якість перекладу — це не сума влучних рішень, а баланс між системністю, нормативністю та адаптацією до реального інтерфейсу. Уніфіковані глосарії, процесні інструменти й автоматичні перевірки роблять цей баланс не випадковим, а керованим, залишаючи місце для точності й природності навіть у найскладніших технічних текстах.

Висновки до четвертого розділу

Попри зовнішню формальність і технічність комп’ютерної документації, перекладацькі рішення в цьому домені не є механічними. Аналіз показав, що найстабільніші результати дають стратегії, які одночасно враховують структурну модель англійського терміну й норми української науково-технічної мови. Для UI-термінології найадекватнішими виявилися усталені еквіваленти з компактною синтаксичною побудовою — вони не лише зберігають семантику, а й вкладаються в рамки інтерфейсу, де зайвий складник може спричинити обрізання або втрату фокусу. Для багатокomпонентних конструкцій типовим оптимальним рішенням є калька з нормуванням порядку або термінологічний синтез, особливо коли структура N+N чи Adj+N легко відтворюється українськими моделями. Помилки концентруються переважно у трьох сегментах: некоректному калькуванні, невдалому дослівному відтворенні фразових структур та інтерференції англійського порядку слів, що призводить до синтаксично «чужого» вигляду терміну.

Практична цінність цього дослідження розкривається насамперед у тому, що воно показує: технічна точність перекладу — це не лише питання мовного чуття чи знання предметної області, а результат системної роботи з термінологією. Добре структурований глосарій фактично виступає внутрішньою «мовною інфраструктурою» продукту. Він не просто пропонує списки відповідників, а задає

рамку для стабільного використання термінів у всіх контекстах: від документації і технічних статей до підказок у UI. Завдяки цьому забезпечується консистентність і на рівні термін-типів (унікальних одиниць), і на рівні термін-токенів (їхніх конкретних входжень у корпусі). У перекладацькому середовищі саме нестабільність на рівні токенів найчастіше призводить до збоїв у сприйнятті — наприклад, коли один і той самий термін у різних частинах продукту названо по-різному («обліковий запис» / «акаунт» / «рахунок користувача»). Глосарій усуває такі ризики, перетворюючи переклад на керовану систему.

Стиль-гайд доповнює цю систему на рівні формальних характеристик: дефіси, регістр, типи лапок, правила транслітерації власних назв, уживання латиниці в термінах, структурування багатокomпонентних конструкцій. З погляду локалізації саме ці дрібні деталі впливають на відчуття «рідномовності» тексту. Непослідовне оформлення — наприклад, варіації типу «JSON файл», «JSON-файл», «json-файл» — створює враження фрагментарності, навіть якщо семантика передана точно. Окрім того, стиль-гайд задає рекомендації щодо довжини рядків у UI, формулювання коротких інструкцій, а також розмежовує випадки, коли варто зберігати англійський термін, і коли — адаптувати його під норми української мови.

Процесний контроль є третім, але не менш важливим елементом, оскільки дозволяє побачити термінологічні рішення «в дії». UI-QA виявляє ті проблеми, які на рівні текстового файлу непомітні: обрізані підказки, зламані переноси, надмірно довгі назви кнопок, колізії термінів у сусідніх елементах інтерфейсу. Рецензування другим перекладачем або редактором із термінологічним фокусом додає ще один шар контролю — фахівець може помітити структурні кальки, фальшивих друзів або нетипову для української мови керованість, які вислизують від уваги перекладача, зануреного в потік операційних задач. Робота з макетами (скрінами, прототипами) дозволяє не лише перевірити формулювання, а й оцінити його функціональність: чи зрозуміє користувач кнопку «Скинути» в ситуації, коли система насправді очікує «Скинути зміни», чи не буде двозначності між «Видалити» і «Очистити».

Однак це дослідження має свої межі, і важливо окреслити їх достатньо чітко, щоб не створювати ілюзії універсальності отриманих результатів. Корпус термінів був сформований переважно на основі продуктів із масовою аудиторією — систем керування проєктами, хмарних сервісів, інтерфейсів адміністрування чи стандартної технічної документації. Такі продукти мають тенденцію до уніфікації термінології і часто використовують подібні патерни найменувань, які добре інтегруються в українську систему. Це означає, що результати більшою мірою відображають особливості саме «побутової» технічної мови, а не фахових підсистем. У реальних виробничих середовищах терміни можуть бути довшими, менш варіативними та жорстко закріпленими у внутрішніх стандартах, що робить будь-яку адаптацію вкрай небажаною.

Контекст платформи — мобільний застосунок, вебінтерфейс чи десктопна панель — також суттєво впливає на перекладацькі рішення. Наприклад, мобільний UI формує значно суворіший ліміт на довжину терміну: кнопка з написом «Керування користувацькими дозволами» може виглядати прийнятним в десктопній системі, але стане повністю нефункціональною у мобільному інтерфейсі, де доступні лише 12–18 символів. Це, у свою чергу, стимулює появу скорочених форм, компресійних моделей або навіть часткових переосмислень термінів, що не завжди відповідає вимогам нормативності. У вебінтерфейсах, навпаки, можливе застосування розгорнутих описових моделей, які не виглядали б природньо на панелі управління фізичним пристроєм чи в CLI-документації. Таким чином, синтаксика перекладу тісно залежить від фізичного простору екрану, поведінки користувачів, логіки навігації та архітектури самої системи.

Додатковим чинником є семантична економія: перекладач має знаходити баланс між інформативністю та компактністю, щоб термін залишався зрозумілим навіть у скороченій формі. Це часто вимагає створення нових моделей компресії — аббревіацій, усічених словосполучень чи контекстуальних замінів, які зберігають ключове значення, але відповідають просторовим обмеженням. У цьому процесі важливо уникати надмірної редукції, яка може призвести до втрати точності або появи двозначності.

Деякі висновки цього дослідження можуть не перенестися на високоспеціалізовані області, де домінує стабільність і стандартизованість термінів. Криптографічні протоколи, мережеві стекові моделі чи системи промислової автоматизації використовують термінологію, яка живе за власними правилами. Тут навіть мінімальна модифікація структури може призвести до втрати точності або руйнування семантичних меж поняття. Наприклад, терміни на кшталт *Elliptic Curve Diffie–Hellman*, *Transport Layer Security* або *Forward Secrecy* не допускають «перекладів за відчуттям» — для них потрібна суворі відповідність міжнародним стандартам, а іноді й збереження англійських форм без будь-якої адаптації. У таких сферах компактність формулювання поступається точності, а перекладацькі стратегії мають ґрунтуватися на практиках IEEE, IETF чи NIST, а не на евристичних локалізаційних масових продуктах.

Подальші дослідження можуть рухатися одразу кількома напрямками. По-перше, розширення корпусу на домени безпеки, мережевих технологій або хмарної інфраструктури дозволило б виявити нові моделі інтерференції та стратегії, що працюють у більш технічних середовищах. По-друге, порівняння українських перекладацьких рішень з іншими мовами — польською, чеською чи німецькою — могло б показати, наскільки типові наші виклики та як різні мовні системи розв’язують однакові структурні проблеми.

Експериментальне UX-тестування термінів дало б змогу перейти від теоретичної оцінки «термін зрозумілий/незрозумілий» до вимірюваних змін у поведінці користувачів: швидкості виконання дій, кількості помилок або часу прийняття рішення. Це відкриває перспективу бачити переклад не лише як передачу змісту, а як інструмент взаємодії людини з технологією — і саме в цьому, мабуть, лежить найцікавіший продовжений вектор роботи.

ЗАГАЛЬНІ ВИСНОВКИ

Складається враження, що комп'ютерна термінологія постійно тікає вперед швидше, ніж ми встигаємо її описати чи перекласти — і саме цей рух став головним викликом та водночас предметом дослідження в цій роботі. У першому розділі з'ясувалося, що ІТ-лексикон функціонує як динамічна, напівавтономна система, у якій неологізми — від *containerization* до *prompt engineering* — не є винятками, а радше базовим механізмом розвитку. Структурна різноманітність термінів, що поєднує односкладові слова (*cache*, *bug*), багатокомпонентні конструкції на кшталт *cross-platform compatibility* або *runtime environment*, а також розгалужені системи аббревіатур (API, GPU, OCR), формує складний перекладацький рельєф. Поряд із цим простежено виразний стилістичний код технічних текстів — прагматичність, функціональність, націленість на точність, яка не терпить багатозначності. Усе це створює контекст, у якому навіть дрібна зміна термінологічної норми здатна вплинути на узгодженість цілої документації.

Другий розділ показав, що процес перекладу комп'ютерної термінології не зводиться до вибору між калькою і адаптацією: за кожним рішенням стоїть ціла система чинників. Теоретичний огляд підкреслив важливість гнучкого використання стратегії калькування (*virtual machine* — *віртуальна машина*), адаптації (*deprecated* — *застарілий*), транслітерації (*backend* — *бекенд*) та описового перекладу (*rate limiting* — *обмеження частоти запитів*). Вибір стратегії визначається не лише семантикою оригіналу, але й нормами цільової мови, контекстом, прагматикою тексту та очікуваннями цільової аудиторії. Баланс між точністю й доступністю стає ключовим: надмірна калька провокує інтерференцію, а надмірна адаптація — втрату системності. Робота наголошує, що успішний переклад технічної документації неможливий без врахування цього балансу.

Третій розділ розкрив проблему термінологічної варіативності: паралельне функціонування відповідників на кшталт *запит/реквест*, *вебзастосунок/вебдодаток*, *браузер/переглядач* знижує консистентність текстів і

ускладнює сприйняття. У цьому контексті перекладач виступає не лише медіатором, а й активним нормотворцем, здатним пропонувати усталені українські відповідники, узгоджені між собою та з міжнародними стандартами. Саме стандартизаторські документи (ISO/IEC 2382, Microsoft Language Portal, Apple Style Guide, державні галузеві стандарти) визначають рамки, у яких формується сучасна українська IT-термінографія. Їхнє дотримання допомагає уникнути хаосу та забезпечити послідовність у перекладі.

У четвертому розділі практичний аналіз дав змогу не лише описати, а й емпірично перевірити ефективність перекладацьких стратегій. Найчастіше використовуваними виявилися калькування (приблизно 42 % випадків) і адаптація (34 %), тоді як описовий переклад та транслітерація разом становили близько чверті рішень. Визначено типові помилки: калькування без урахування норм мови (*save as* — *зберегти як...* у низці контекстів неправильно передано як *зберегти як є*), інтерференція англійських конструкцій (*увійдіть у акаунт* замість *увійдіть у обліковий запис*), непослідовне використання термінів протягом документа. Практичні рекомендації — створення глосаріїв, формування стиль-гайдів, використання QA-перевірок — продемонстрували здатність знижувати рівень неконсистентності. Статистичні дані підтвердили, що впровадження глосарію на початковому етапі перекладу зменшує кількість різночитань приблизно на третину.

Наукова новизна роботи полягає в тому, що вперше здійснено системний аналіз особливостей перекладу англomовної комп'ютерної термінології саме в українському локалізаційному контексті, де поєднується опис мовного явища, моделювання перекладацьких стратегій і корпусний аналіз реальних перекладів технічних текстів. Робота не лише узагальнює існуючий досвід, а й пропонує методичні рекомендації, що мають практичну цінність для перекладачів, редакторів технічної документації та локалізаторів.

Практична значущість дослідження полягає у можливості застосування результатів у професійній локалізації програмного забезпечення, підготовці термінологічних глосаріїв і галузевих словників, навчальних курсах з технічного

перекладу та вдосконаленні стандартів термінотворення. Запропоновані рекомендації не лише підвищують якість перекладу і зрозумілість інтерфейсу, але й сприяють формуванню довіри користувачів до україномовних продуктів — фактор, який дедалі більше впливає на їхню конкурентоспроможність.

Водночас дослідження має певні обмеження. Корпус матеріалів охоплює лише два типи текстів — користувацький інтерфейс і технічні мануали. Інші жанри ІТ-комунікації, зокрема наукові статті, стандарти або маркетингові матеріали, залишилися поза межами аналізу. Окрім того, робота зосереджена виключно на англо-українському напрямі, тоді як порівняння з польською, чеською чи німецькою практиками могло б розширити розуміння нормотворчих тенденцій. Нарешті, комп'ютерна термінологія змінюється надто швидко, щоб будь-який аналіз претендував на абсолютну завершеність: деякі терміни, зафіксовані сьогодні, уже завтра можуть отримати нові значення.

Перспективи подальших досліджень відкриваються в кількох напрямках: розширення корпусу на інші жанри цифрової комунікації, зіставлення українських перекладацьких стратегій з практиками інших мовних спільнот, дослідження впливу термінологічних рішень на користувацький досвід, а також розробка автоматизованих інструментів контролю якості перекладу на основі лінгвістичних моделей.

Узагальнюючи, переклад комп'ютерної термінології постає не лише як технічна операція чи мовне рішення, а як процес, що має творчу, аналітичну та нормотворчу природу. Він бере участь у формуванні сучасного українського ІТ-дискурсу, визначає шляхи його інтеграції у глобальний цифровий простір і впливає на те, якою буде українська мова у світі технологій. Можливо, саме від точності та виваженості цих перекладацьких рішень залежить, наскільки органічно українська мова почуватиметься у майбутніх цифрових екосистемах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Базалій А., Ковальова І. Основні особливості перекладу технічних текстів комп'ютерної тематики. Вісник Нац. техн. ун-ту «ХПІ». Серія: Актуальні проблеми розвитку українського суспільства. 2018. Вип. 4 (1280). С. 84–88.
2. Барнич І. І. Англійські запозичення в комп'ютерній термінології німецької та української мов. Львівський філологічний часопис. 2020. Вип. 7. С. 13–18.
3. Білозерська Л. П. Термінологія та переклад: навчальний посібник. Вінниця: Нова книга, 2010.
4. Богачик М. Особливості словотворення англійської комп'ютерної термінології. Наукові записки Нац. ун-ту «Острозька академія». Серія «Філологічна». 2015. Вип. 53. С. 42–44.
5. Богачик М. С. Особливості перекладу комп'ютерних скорочень в англомовних текстах. Наукові записки Національного університету «Острозька академія». 2014. Вип. 49. С. 8–11.
6. Бондаренко О. Структурні особливості та специфіка перекладу термінології підмови інформаційних технологій. Наукові записки. Серія «Філологічна». Вип. 126. Кіровоград. 2014. С. 97-103.
7. Борисевич І. П., Ратомська Л. В. Термінознавство та особливості перекладу термінів українською мовою. Вісник Дніпропетровського ун-ту ім. Альфреда Нобеля. 2012. С. 22–229.
8. Буйвол О. В. Щодо проблеми перекладу термінології в галузі інформаційних технологій. Редакційна колегія. 2021. С. 379.
9. Головачова Є.О. Лінгвопрагматичні аспекти англійсько-українського перекладу комп'ютерних термінів та аббревіатур. – Полтава : Нац. ун-т ім. Юрія Кондратюка, 2025. – 109 с.
10. Гриців Н. М., Сабан О. В. Особливості перекладу англомовних термінів перекладознавчого спрямування українською мовою. *Nova filologiâ*. 2024. № 96. С. 47–55.

11. Деркач І. В., Колісник М. П. Визначення комп'ютерної термінології як лінгвістичне і галузеве явище. The 3rd International scientific and practical conference «International experience in scientific research». Chicago: BoScience Publisher, 2025. 694 с. С. 511.
12. Дичка Н.І., Гордієнко Н.М. стилістичні особливості англomовної технічної документації. *Молодий вчений*. 2019. 1 (65), 357-360.
13. Жовталюк Н. Актуальні питання сучасного перекладознавства. *Theoretical and empirical scientific research: concept and trends* / chair І. Козубай. 2021.
14. Кивлюк В. О. Модель перекладу англійських комп'ютерних термінів на українську мову. Освітній дискурс. Гуманітарні науки. 2017. Вип. 1. С. 145–154.
15. Кириченко О.А. Новації у мові комп'ютерних користувачі. Філологічні трактати. 2010. Т.2. № 3. С. 51-55.
16. Козуб Л. С., Малишко О. Ю. Специфіка перекладу англійських комп'ютерних термінів українською мовою. Науковий вісник Нац. ун-ту біоресурсів і природокористування України. Серія: Філологічні науки. 2013. Вип. 186 (2). С. 151–156.
17. Корнєва З., Козак І. Новітні тенденції у творенні термінології в галузі розробки програмного забезпечення. *Advanced Linguistics*. 2023. Вип. 12. С. 107–114.
18. Кравцова К. А. Переклад комп'ютерної лексики з англійської мови. Матеріали ІІ Всеукр. студ. наук.-практ. інтернет-конф. «Мова і світ: проблеми філології та перекладознавства». Східноукр. нац. ун-т ім. В. Даля, 2019. С. 36.
19. Крячко В. І., Бондар М. О. Проблематика досягнення еквівалентності при перекладі термінологічної лексики комп'ютерної галузі. The 6th International scientific and practical conference «Modern research in world science». – Lviv: SPC «Sci-conf.com.ua», 2022. 1018 с. С. 643.
20. Кучман І. М. Переклад англійських термінів у галузі комп'ютерних технологій. Вісник Житомирського держ. ун-ту ім. Івана Франка. 2005. Вип. 23. С. 164–166.

21. Л. Полюга. Теорія та прагматика термінологічної лексики: тези доповідей Республ. наук.-метод. конф. К.: УМК ВО, 1991. С.79-80.
22. Лелека, Т. О. (2024). Особливості перекладу у культурологічному аспекті. Наукові записки. Серія: Філологічні науки, (208), 247–252.
23. Лях, О. О. Особливості передачі англomовної термінології електротехнічної та електроенергетичної галузей в українськомовному перекладі : магістерська дис. : 035 Філологія. Київ, 2022. 128 с.
24. Мартинюк О. В. Попереднє та кінцеве редагування текстів у процесі машинного перекладу засобами комп'ютерного програмного забезпечення та онлайн сервісів. Актуальні проблеми філології та перекладознавства : зб. наук. пр. Хмельницький : ХНУ, 2019. № 17. С .22–26.
25. Матвійчук О., Сітко А. Дослідження перекладацьких трансформацій в українському перекладознавстві. Grail of Science. 2023. Вип. 31. С. 312–315.
26. Ментинська І. Б., Наконечна Г. В. Становлення української комп'ютерної термінології: лексикографічний аспект. Вісник Ун-ту ім. Альфреда Нобеля. Серія: Філологічні науки. 2018. Вип. 2. С. 273–278.
27. Ментинська І. Лексико-генетичні особливості сучасної комп'ютерної термінології. Ірина Ментинська. Вісник Нац. ун-ту «Львівська політехніка». Серія «Проблеми української термінології». 2014. № 791. С. 71–74.
28. Ментинська І. Українська комп'ютерна термінологія: формування, системна організація, вектори розвитку. Дис. ... д-ра філософії. 2019. С. 3.
29. Ментинська, І. Б. Етапи становлення комп'ютерної термінології на тлі розвитку інформаційних технологій в Україні. Теорія і практика викладання української мови як іноземної. Вип. 17. 2023. 190 С.
30. Мирошниченко В. М., Шишкова І. С. Англійські лексичні новоутворення у сфері комп'ютерних технологій та особливості їх перекладу українською мовою. Вісник Нац. техн. ун-ту «ХПІ». Серія: Актуальні проблеми розвитку українського суспільства. 2019. Вип. 1. С. 87–92.
31. Мірошниченко А. Т. Сучасна англomовна термінологія ІТ-галузі та особливості її перекладу українською мовою : кваліфікаційна робота магістра

- спеціальності 035 "Філологія". наук. керівник Г. О. Хацер. Запоріжжя : ЗНУ, 2023. 78 с.
32. Небесюк О. С. Комп'ютерна термінологія та способи її перекладу. Збірник матеріалів Всеукр. наук.-практ. конф. студентів та молодих науковців . 2019. С. 206.
 33. Николук Т. В., Шкляєва Н. В. Комп'ютерна термінологія: особливості перекладу та зміни семантики. Наукові записки Нац. ун-ту «Острозька академія». 2018. С. 55–57.
 34. Номировський В. О., Лебедева А. В. Особливості перекладу англійських аббревіатур і скорочень у галузі комп'ютерних технологій українською мовою. Наукові записки. Серія: Філологічні науки. 2023. Вип. 207. С. 175–179.
 35. Павельєва А. К., Леверя Т. І. Лексико-граматичні трансформації при перекладі комп'ютерної термінології з англійської мови на українську. Дис. ... д-ра філософії. Маріуполь: МДУ, 2019.
 36. Радченко О.І. Мовна норма і варіантність в українській науковій термінології: дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.01 „Українська мова”. 2000. 203 с.
 37. Ріжняк О., Кобзар І., Сьомак А., Малюк В. Особливості комп'ютерного перекладу медичної термінології. The 6th International scientific and practical conference «Scientific progress: innovations, achievements and prospects». Munich: MDPC Publishing, 2023. 406 с. С. 80.
 38. Савицька Л. В., Тарасенко С. Є. Особливості перекладу англійських академічних термінів українською мовою. *Лінгвістичні дослідження*. 2018. № 49. С. 59–66.
 39. Сидор А. Р., Нанівський Р. С. Урахування лексичних особливостей сфери інформаційних технологій під час перекладу з англійської мови українською. 2019.
 40. Сизонова М. Ю., Стеценко В. В., Тупиця О. Ю. Особливості відтворення сучасних ІТ-термінів при перекладі на українську мову. *Materialy XVII*

- Międzynarodowej naukowo-praktycznej konferencji «Perspektywiczne opracowania są nauką i technikami». 2021. T. 1. С. 21–24.
41. Слободський Р. Переклад комп'ютерних термінів сфери кібербезпеки з арабської мови українською. Вісник Київського нац. ун-ту ім. Тараса Шевченка. Літературознавство. Мовознавство. Фольклористика. 2018. Т. 28, № 1. С. 49–50.
 42. Стопакевич А. О., Тігарєв А. М., Романюк О. Р., Стопакевич О. А. Система автоматичного корегування англійсько-українського комп'ютерного перекладу для технічних текстів в галузі автоматизації технологічних процесів. Informatics & Mathematical Methods in Simulation. Informatika ta Matematični Metodi v Modelûvannî. 2023. Т. 13.
 43. Тищенко О. В., Юхимець С. Ю., Фоміна К. Є. Особливості перекладу українськомовної науково-технічної літератури галузі комп'ютерних технологій англійською мовою: лексичний аспект. Науковий вісник Південноукр. нац. пед. ун-ту ім. К. Д. Ушинського. Лінгвістичні науки. 2018. Вип. 26. С. 121–132.
 44. Троян О. П. Особливості перекладу науково-технічних текстів : лексико-перекладацький аспект. Германські мови та літератури; Хмельниц. нац. ун-т. Хмельницький, 2023. 135 с.
 45. Федорченко О. Особливості перекладу термінів дискурсу інформаційних технологій. Редакційна колегія: Климова К. Я., доктор педагогічних наук. 2023. С. 87.
 46. Філь О. Джерела і способи формування терміносистем комп'ютерної сфери в англійській, польській та українській мовах. Вісник Нац. ун-ту «Львівська політехніка». Серія «Проблеми української термінології». 2014. № 791. С. 92–97.
 47. Хавкіна О. М. Англійські метафоричні терміни мови комп'ютерних мережевих технологій та інформаційної безпеки. Структурно-семантичні особливості та специфіка перекладу українською мовою. Нова філологія. 2012. Вип. 50. С. 209–213.

48. Цвіркун О. А., Явтушенко В. М. Особливості перекладу українською мовою у сфері інформаційних технологій. *Synergy of Knowledge: Innovations at the Intersection of Disciplines*. 2019. С. 209.
49. Яременко Н. В. Стратегії перекладу англomовної науково-технічної термінології українською мовою. *Науковий вісник Нац. ун-ту біоресурсів і природокористування України. Серія: Філологічні науки*. 2017. Вип. 272. С. 44–52.
50. Bowker L. Terminology and translation. Amsterdam/Philadelphia: John Benjamins, 2015. Vol. 1, No. 1. P. 304–323.
51. Bowker L., Delsey T. Information science, terminology and translation studies: Adaptation, collaboration, integration. *Border Crossings*. Amsterdam/Philadelphia: John Benjamins Publishing Company, 2016. С. 73–96.
52. Bowker L., Fisher D. Computer-aided translation. *Handbook of Translation Studies: Volume 1*. Amsterdam/Philadelphia: John Benjamins Publishing Company, 2012. P. 60–65.
53. Christensen T. P., Schjoldager A. Technical documentation and (technical) translation: A case study of work practices and concepts. *Challenging Boundaries: New Approaches to Specialized Communication*. 2019. P. 7–16.
54. Cleary Y., Flammia M., Minacori P., Slattery D. M. Global virtual teams create and translate technical documentation: Communication strategies, challenges and recommendations. 2015 IEEE International Professional Communication Conference (IPCC). July 2015. IEEE. P. 1–10.
55. Cleary Y., Flammia M., Minacori P., Slattery D. M. Global virtual teams create and translate technical documentation: Communication strategies, challenges and recommendations. 2015 IEEE International Professional Communication Conference (IPCC). July 2015. IEEE. P. 1–10.
56. Crabbe S. Existing controlled languages for technical documents. *Controlling Language in Industry: Controlled Languages for Technical Documents*. Cham: Springer International Publishing, 2017. С. 23–47.

57. Craciunescu O., Gerding-Salas C., Stringer-O'Keeffe S. Machine translation and computer-assisted translation. *Machine Translation and Computer-Assisted Translation*. 2004.
58. Granger S. *Learner English on computer*. London: Routledge, 2014.
59. Holtsch D., Rohr-Mentele S., Wenger E., Eberle F., Shavelson R. J. Challenges of a cross-national computer-based test adaptation. *Empirical Research in Vocational Education and Training*. 2016. Vol. 8, № 1. P. 18.
60. Khuwaileh A. A. IT terminology and translation: Cultural, lexicographic and linguistic problems. *LSP Journal – Language for special purposes, professional communication, knowledge management and cognition*. 2010. T. 1, № 2.
61. Khuwaileh A. A., Khwaileh T. IT terminology, translation, and semiotic levels: Cultural, lexicographic, and linguistic problems. *Semiotica*. 2011. № 187. P. 239–249.
62. Krein-Kühle M. Register shifts in scientific and technical translation: A corpus-in-context study. *The Translator*. 2011. Vol. 17, № 2. C. 391–413.
63. Mandziy K. S., Yurlova U. V., Dilai M. P. English-Ukrainian parallel corpus of IT texts: Application in translation studies. *COLINS*. 2022. C. 724–736.
64. Moureau M. Scientific and technical documentation centers. *The Journal of Library History (1974–1987)*. 1984. Vol. 19, № 1. C. 86–102.
65. Olalla-Soler C. Using translation strategies to solve cultural translation problems. Differences between students and professional translators. *Perspectives*. 2019. Vol. 27, № 3. P. 367–388.
66. Piqué-Angordans J. O. R. D. I., Posterguillo S. Specialized dictionaries and corpus linguistics in the translation of computer terminology. *Problemas lingüísticos en la traducción especializada*. Valladolid: Secretariado de Publicaciones e Intercambio Editorial de la Universidad de Valladolid, 2007. P. 87–107.
67. Pym A. Professional corpora: Teaching strategies for work with online documentation, translation memories, and content management. *Chinese Translators' Journal*. 2008. Vol. 29, № 2. P. 41–45.

68. Rathert S., Cabaroglu N. Theorising textbook adaptation in English language teaching. *CEPS Journal*. 2022. Vol. 12, № 2. P. 169–188.
69. Rogers M. Terminology and Specialised Translation: A Historical Perspective. *Specialised Translation*. London, 2015. P. 81–109.
70. Seljan S. Translation technology as challenge in education and business. *Informatologia*. 2011. T. 44, № 4. P. 279–286.
71. Serpil H., Durmuşoğlu-Köse G., Erbek M., Öztürk Y. Employing computer-assisted translation tools to achieve terminology standardization in institutional translation: Making a case for higher education. *Procedia-Social and Behavioral Sciences*. 2016. Vol. 231. P. 76–83.
72. Sperandio E. Influences of technical documentation and its translation on efficiency and customer satisfaction. *Communication Practices in Engineering, Manufacturing, and Research for Food and Water Safety*. 2015. P. 145–169.
73. Su T., Rue L. P. The analysis of English language teaching with machine translation based on virtual reality technology. *Scientific Reports*. 2025. Vol. 15, № 1. Article 15845.
74. Sysoev A., Shapsheva N. Translating the attribute group in the titles of the English-language sci-tech documents for cataloguing. *Scientific and Technical Libraries (Naucnye i tekhnicheskie biblioteki)*. 2019. № 261.
75. ten Hacken P., Fernández Parra M. Terminology and formulaic language in computer-assisted translation. *SKASE Journal of Translation and Interpretation*. 2008. Vol. 3. P. 1–16.
76. Todea L. Translation of technical texts in engineering: theories, challenges and uses. *Annales Universitatis Apulensis. Series Philologica*. 2025. Vol. 26, № 2. C. 365–377.
77. Van der Eijck P. Controlled languages in technical documentation. *Selected Papers from the Eight CLIN Meeting*. 1998. C. 187–203.
78. Vargas-Sierra C. Translation-oriented terminology management and ICTs: present and future. *Interdisciplinarity and languages: Current Issues in Research, Teaching, Professional Applications and ICT*. Bern: Peter Lang Publishing, 2011. P. 45–64.

79. Yang T., Fan H. Application of computer technology in English translation. Journal of Physics: Conference Series. June 2020. Vol. 1575, № 1. P. 012029. IOP Publishing.

ДОДАТКИ

Додаток А

Таблиця 1. Корпус англомовних комп'ютерних термінів та їх українських відповідників.

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
1	Settings	Налаштування	UI	Функціональний еквівалент	Усталений варіант
2	Preferences	Параметри	UI	Еквівалент	Часто плутають із “Налаштування”
3	Save	Зберегти	UI	Еквівалент	Кнопка дії
4	Cancel	Скасувати	UI	Еквівалент	Стандартний варіант
5	Retry	Повторити спробу	UI	Еквівалент	Калька “переспробувати” — помилкова
6	Error	Помилка	UI	Еквівалент	Усталений термін
7	Warning	Попередження	UI	Еквівалент	Використовується у повідомленнях
8	Tooltip	Підказка	UI	Адаптація	Коротке пояснення
9	Home	Головна	UI	Еквівалент	Меню навігації
10	Back	Назад	UI	Еквівалент	Кнопка навігації
11	Next	Далі	UI	Еквівалент	Кнопка навігації
12	File system	Файлова система	Багатокомпонентний	Калька	Усталений термін
13	Version control system	Система керування версіями	Багатокомпонентний	Калька + адаптація	Відповідає нормам
14	Access token	Токен доступу	Багатокомпонентний	Калька	Використовується у безпеці
15	Single sign-on (SSO)	Єдиний вхід	Багатокомпонентний	Описовий переклад	Пояснюється при першій згадці
16	Incremental backup	Інкрементне резервне копіювання	Багатокомпонентний	Калька	Технічний термін
17	Cloud	Хмара	Неологізм	Адаптація	Усталений варіант
18	Firewall	Брандмауер	Неологізм	Транскрипція	Усталений відповідник
19	Cache	Кеш	Неологізм	Транскрипція	Використовується у технічних текстах
20	Bug	Помилка	Неологізм	Адаптація	Усталений варіант

21	Patch	Патч / Латка	Неологізм	Транскрипція / Адаптація	Варіативність
----	-------	--------------	-----------	--------------------------	---------------

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
22	API	API (інтерфейс прикладного програмування)	Абревіатура	Латиниця + пояснення	Пояснюється при першій згадці
23	CPU	ЦП (центральний процесор)	Абревіатура	Еквівалент	Використовується у технічних текстах
24	GPU	ГП (графічний процесор)	Абревіатура	Еквівалент	Усталений варіант
25	RAM	ОЗП (оперативна пам'ять)	Абревіатура	Еквівалент	Усталений варіант
26	ROM	ПЗП (постійна пам'ять)	Абревіатура	Еквівалент	Використовується у документації
27	DNS	DNS (система доменних імен)	Абревіатура	Латиниця + пояснення	Пояснюється при першій згадці
28	HTTP	HTTP (протокол передачі гіпертексту)	Абревіатура	Латиниця + пояснення	Усталений стандарт
29	TLS	TLS (протокол захисту транспортного рівня)	Абревіатура	Латиниця + пояснення	Використовується у безпеці
30	JSON	JSON (формат обміну даними)	Абревіатура	Латиниця + пояснення	Усталений формат
31	XML	XML (розширювана мова розмітки)	Абревіатура	Латиниця + пояснення	Використовується у документації
32	Log in	Увійти	UI	Еквівалент	Калька “логінітися” — помилкова
33	Log out	Вийти	UI	Еквівалент	Усталений варіант
34	Sign up	Зареєструватися	UI	Еквівалент	Використовується у формах
35	Upload	Завантажити	UI	Еквівалент	Усталений варіант
36	Download	Завантажити	UI	Еквівалент	Контекст визначає напрям
37	Refresh	Оновити	UI	Еквівалент	Кнопка дії
38	Search	Пошук	UI	Еквівалент	Усталений варіант
39	Print	Друкувати	UI	Еквівалент	Кнопка дії
40	Clipboard	Буфер обміну	Технічний термін	Еквівалент	Усталений варіант
41	Shortcut	Ярлик	Технічний термін	Еквівалент	Використовується у Windows
42	Driver	Драйвер	Технічний термін	Транскрипція	Усталений варіант
43	Kernel	Ядро	Технічний термін	Еквівалент	Використовується у Linux

44	Repository	Репозиторій	Технічний термін	Транскрипція	Використовується у Git
----	------------	-------------	------------------	--------------	------------------------

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
45	Branch	Гілка	Технічний термін	Адаптація	Усталений варіант
46	Commit	Коміт / Фіксація	Технічний термін	Транскрипція / Адаптація	Варіативність
47	Merge	Злиття	Технічний термін	Еквівалент	Використовується у Git
48	Debugging	Налагодження	Технічний термін	Еквівалент	Усталений варіант
49	Encryption	Шифрування	Технічний термін	Еквівалент	Використовується у безпеці
50	Authentication	Автентифікація	Технічний термін	Транскрипція	Усталений варіант
51	Dashboard	Панель керування	UI	Адаптація	Використовується у веб-інтерфейсах
52	Sidebar	Бічна панель	UI	Еквівалент	Усталений варіант
53	Tab	Вкладка	UI	Еквівалент	Важливо відрізнити від клавіші Tab
54	Checkbox	Прапорець	UI	Адаптація	Усталений варіант
55	Dropdown	Розкривний список	UI	Описовий переклад	Використовується у формах
56	Slider	Повзунок	UI	Еквівалент	Елемент керування
57	Snackbar	Спливаюче повідомлення	UI	Описовий переклад	Використовується у мобільних додатках
58	Modal window	Модальне вікно	UI	Калька	Усталений варіант
59	Placeholder	Заповнювач	UI	Еквівалент	Використовується у формах
60	Breadcrumbs	Хлібні крихти	UI	Калька	Усталений термін навігації
61	Repository branch	Гілка репозиторію	Багатокомпонентний	Калька	Використовується у Git
62	Pull request	Запит на злиття	Багатокомпонентний	Описовий переклад	Усталений варіант
63	Continuous integration	Безперервна інтеграція	Багатокомпонентний	Калька	Використовується у DevOps
64	Machine learning	Машинне навчання	Багатокомпонентний	Калька	Усталений термін
65	Deep learning	Глибинне навчання	Багатокомпонентний	Калька	Використовується у штучному інтелекті
66	Neural network	Нейронна мережа	Багатокомпонентний	Калька	Усталений термін
67	Data mining	Інтелектуальний аналіз даних	Багатокомпонентний	Описовий переклад	Використовується у статистиці
68	Big data	Великі дані	Багатокомпонентний	Калька	Усталений термін

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
69	Virtual machine	Віртуальна машина	Багатокомпонентний	Калька	Використовується у системах
70	Load balancing	Балансування навантаження	Багатокомпонентний	Калька	Використовується у мережах
71	VPN	VPN (віртуальна приватна мережа)	Абревіатура	Латиниця + пояснення	Усталений термін
72	IP address	IP-адреса	Абревіатура	Латиниця + адаптація	Використовується у мережах
73	MAC address	MAC-адреса	Абревіатура	Латиниця + адаптація	Унікальний ідентифікатор
74	URL	URL (уніфікований локатор ресурсів)	Абревіатура	Латиниця + пояснення	Усталений термін
75	SQL	SQL (мова структурованих запитів)	Абревіатура	Латиниця + пояснення	Використовується у базах даних
76	NoSQL	NoSQL (нереляційні бази даних)	Абревіатура	Латиниця + пояснення	Використовується у сучасних системах
77	IDE	IDE (інтегроване середовище розробки)	Абревіатура	Латиниця + пояснення	Використовується у програмуванні
78	SDK	SDK (набір засобів розробки)	Абревіатура	Латиниця + пояснення	Використовується у мобільних додатках
79	GUI	GUI (графічний інтерфейс користувача)	Абревіатура	Латиниця + пояснення	Усталений термін
80	CLI	CLI (інтерфейс командного рядка)	Абревіатура	Латиниця + пояснення	Використовується у системах
81	Ping	Пінг	Технічний термін	Транскрипція	Використовується у мережах
82	Latency	Затримка	Технічний термін	Еквівалент	Використовується у мережах
83	Bandwidth	Пропускна здатність	Технічний термін	Еквівалент	Усталений варіант
84	Packet loss	Втрата пакетів	Технічний термін	Калька	Використовується у мережах
85	Root directory	Кореневий каталог	Технічний термін	Калька	Використовується у файлових системах
86	Path	Шлях	Технічний термін	Еквівалент	Використовується у файлових системах
87	Script	Скрипт	Технічний термін	Транскрипція	Використовується у програмуванні

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
---	--------------------	----------------------	-----------	---------------------	----------

88	Function	Функція	Технічний термін	Еквівалент	Використовується у програмуванні
89	Variable	Змінна	Технічний термін	Еквівалент	Використовується у програмуванні
90	Loop	Цикл	Технічний термін	Еквівалент	Використовується у програмуванні
91	Array	Масив	Технічний термін	Еквівалент	Використовується у програмуванні
92	Object	Об'єкт	Технічний термін	Еквівалент	Використовується у програмуванні
93	Class	Клас	Технічний термін	Еквівалент	Використовується у програмуванні
94	Exception	Виняток	Технічний термін	Еквівалент	Використовується у програмуванні
95	Stack	Стек	Технічний термін	Транскрипція	Використовується у програмуванні
96	Queue	Черга	Технічний термін	Еквівалент	Використовується у програмуванні
97	Thread	Потік	Технічний термін	Еквівалент	Використовується у багатозадачності
98	Process	Процес	Технічний термін	Еквівалент	Використовується у системах
99	Scheduler	Планувальник	Технічний термін	Еквівалент	Використовується у ОС
100	Garbage collection	Збирання сміття	Технічний термін	Калька	Використовується у програмуванні
101	Authentication token	Маркер автентифікації	Безпека	Калька	Використовується у системах доступу
102	Authorization	Авторизація	Безпека	Транскрипція	Усталений варіант
103	Credentials	Облікові дані	Безпека	Еквівалент	Використовується у логіні
104	Two-factor authentication	Двофакторна автентифікація	Безпека	Калька	Використовується у безпеці
105	Public key	Відкритий ключ	Безпека	Калька	Використовується у криптографії
106	Private key	Закритий ключ	Безпека	Калька	Використовується у криптографії
107	Digital signature	Цифровий підпис	Безпека	Калька	Усталений термін
108	Hash function	Хеш-функція	Безпека	Калька	Використовується у криптографії
109	Malware	Шкідливе програмне забезпечення	Безпека	Описовий переклад	Усталений варіант
110	Phishing	Фішинг	Безпека	Транскрипція	Використовується у кіберзлочинах

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
---	--------------------	----------------------	-----------	---------------------	----------

111	Spoofing	Підміна	Безпека	Адаптація	Використовується у мережах
112	Denial of Service (DoS)	Відмова в обслуговуванні	Безпека	Калька	Тип атаки
113	Distributed Denial of Service (DDoS)	Розподілена відмова в обслуговуванні	Безпека	Калька	Тип атаки
114	Intrusion detection system	Система виявлення вторгнень	Безпека	Калька	Використовується у мережах
115	Honeypot	Пастка	Безпека	Адаптація	Використовується у кіберзахисті
116	Sandbox	Пісочниця	Безпека	Адаптація	Використовується для тестування
117	Virtualization	Віртуалізація	Системи	Транскрипція	Усталений термін
118	Containerization	Контейнеризація	Системи	Транскрипція	Використовується у DevOps
119	Microservices	Мікросервіси	Системи	Калька	Використовується у програмуванні
120	Orchestration	Оркестрація	Системи	Транскрипція	Використовується у Kubernetes
121	Kubernetes	Kubernetes	Системи	Латиниця	Назва платформи
122	Docker	Docker	Системи	Латиниця	Назва платформи
123	Cloud computing	Хмарні обчислення	Системи	Калька	Усталений термін
124	Edge computing	Периферійні обчислення	Системи	Калька	Використовується у сучасних технологіях
125	Fog computing	Туманні обчислення	Системи	Калька	Менш усталений термін
126	Database	База даних	Бази даних	Еквівалент	Усталений термін
127	Table	Таблиця	Бази даних	Еквівалент	Використовується у SQL
128	Row	Рядок	Бази даних	Еквівалент	Використовується у SQL
129	Column	Стовпець	Бази даних	Еквівалент	Використовується у SQL
130	Index	Індекс	Бази даних	Еквівалент	Використовується у SQL
131	Primary key	Первинний ключ	Бази даних	Калька	Використовується у SQL
132	Foreign key	Зовнішній ключ	Бази даних	Калька	Використовується у SQL
133	Query	Запит	Бази даних	Еквівалент	Усталений термін
134	Stored procedure	Збережена процедура	Бази даних	Калька	Використовується у SQL

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
135	Trigger	Тригер	Бази даних	Транскрипція	Використовується у SQL
136	Transaction	Транзакція	Бази даних	Транскрипція	Усталений термін
137	Rollback	Відкат	Бази даних	Адаптація	Використовується у SQL
138	Commit	Фіксація	Бази даних	Адаптація	Використовується у SQL
139	Deadlock	Взаємне блокування	Бази даних	Описовий переклад	Використовується у SQL
140	Scalability	Масштабованість	Системи	Калька	Усталений термін
141	Availability	Доступність	Системи	Еквівалент	Використовується у SLA
142	Reliability	Надійність	Системи	Еквівалент	Використовується у SLA
143	Performance	Продуктивність	Системи	Еквівалент	Використовується у тестах
144	Benchmark	Бенчмарк / Еталон	Системи	Транскрипція / Адаптація	Використовується у тестуванні
145	Deployment	Розгортання	Системи	Еквівалент	Використовується у DevOps
146	Release	Реліз / Випуск	Системи	Транскрипція / Адаптація	Використовується у документації
147	Build	Збірка	Системи	Адаптація	Використовується у CI/CD
148	Patch management	Управління патчами	Системи	Калька	Використовується у безпеці
149	Update	Оновлення	Системи	Еквівалент	Усталений термін
150	Upgrade	Модернізація / Оновлення	Системи	Адаптація	Використовується у документації
151	Homepage	Домашня сторінка	Веб	Еквівалент	Усталений термін
152	Landing page	Цільова сторінка	Веб	Адаптація	Використовується у маркетингу
153	Header	Заголовок	Веб	Еквівалент	Частина сторінки
154	Footer	Нижній колонтитул	Веб	Еквівалент	Частина сторінки
155	Navigation bar	Панель навігації	Веб	Калька	Усталений варіант
156	Responsive design	Адаптивний дизайн	Веб	Калька	Використовується у веб-розробці
157	Framework	Фреймворк	Програмування	Транскрипція	Усталений термін
158	Library	Бібліотека	Програмування	Еквівалент	Використовується у коді
159	Module	Модуль	Програмування	Еквівалент	Усталений термін
160	Plugin	Плагін	Програмування	Транскрипція	Використовується у CMS

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
161	Extension	Розширення	Програмування	Еквівалент	Використовується у браузерях
162	Cookie	Кука / Файл cookie	Веб	Транскрипція / Адаптація	Варіативність
163	Session	Сесія	Веб	Еквівалент	Використовується у веб-автентифікації
164	Cache memory	Кеш-пам'ять	Системи	Калька	Усталений термін
165	Virtual reality (VR)	Віртуальна реальність	Неологізм	Калька	Використовується у сучасних технологіях
166	Augmented reality (AR)	Доповнена реальність	Неологізм	Калька	Використовується у мобільних додатках
167	Mixed reality (MR)	Змішана реальність	Неологізм	Калька	Використовується у сучасних технологіях
168	Blockchain	Блокчейн	Неологізм	Транскрипція	Використовується у фінтех
169	Smart contract	Смарт-контракт	Неологізм	Калька	Використовується у блокчейні
170	Cryptocurrency	Криптовалюта	Неологізм	Калька	Усталений термін
171	Bitcoin	Біткоїн	Неологізм	Транскрипція	Назва валюти
172	Ethereum	Ефіріум	Неологізм	Транскрипція	Назва платформи
173	Tokenization	Токенізація	Безпека	Транскрипція	Використовується у фінтех
174	Data breach	Витік даних	Безпека	Адаптація	Усталений термін
175	Insider threat	Внутрішня загроза	Безпека	Калька	Використовується у кіберзахисті
176	Zero-day exploit	Уразливість нульового дня	Безпека	Калька	Використовується у безпеці
177	Patch update	Оновлення патча	Системи	Калька	Використовується у документації
178	Hotfix	Гаряче виправлення	Системи	Калька	Використовується у підтримці
179	Release notes	Примітки до релізу	Системи	Калька	Документація
180	Changelog	Журнал змін	Системи	Адаптація	Використовується у документації
181	Unit test	Модульний тест	Тестування	Калька	Використовується у QA
182	Integration test	Інтеграційний тест	Тестування	Калька	Використовується у QA
183	Regression test	Регресійний тест	Тестування	Калька	Використовується у QA

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
184	Test case	Тест-кейс	Тестування	Транскрипція	Використовується у QA
185	Bug report	Звіт про помилку	Тестування	Адаптація	Використовується у QA
186	Issue tracker	Система відстеження проблем	Тестування	Описовий переклад	Використовується у Jira
187	Agile	Agile / Гнучка методологія	Методологія	Латиниця + адаптація	Використовується у менеджменті
188	Scrum	Scrum / Скрам	Методологія	Латиниця + транскрипція	Методологія управління
189	Kanban	Канбан	Методологія	Транскрипція	Методологія управління
190	Sprint	Спринт	Методологія	Транскрипція	Використовується у Agile
191	Backlog	Беклог	Методологія	Транскрипція	Використовується у Agile
192	Roadmap	Дорожня карта	Методологія	Адаптація	Використовується у менеджменті
193	Milestone	Віха	Методологія	Еквівалент	Використовується у проєктах
194	Task	Завдання	Методологія	Еквівалент	Використовується у менеджменті
195	Deadline	Кінцевий термін	Методологія	Адаптація	Використовується у менеджменті
196	Repository clone	Клон репозиторію	Програмування	Калька	Використовується у Git
197	Fork	Форк / Відгалуження	Програмування	Транскрипція / Адаптація	Використовується у GitHub
198	Pull	Завантаження	Програмування	Адаптація	Використовується у Git
199	Push	Відправлення	Програмування	Адаптація	Використовується у Git
200	Merge conflict	Конфлікт злиття	Програмування	Калька	Використовується у Git
201	Notification	Сповіщення	UI	Еквівалент	Використовується у мобільних додатках
202	Alert	Попередження	UI	Еквівалент	Використовується у системних повідомленнях
203	Badge	Значок	UI	Еквівалент	Використовується у мобільних інтерфейсах
204	Widget	Віджет	UI	Транскрипція	Усталений термін
205	Shortcut key	Гаряча клавіша	UI	Адаптація	Використовується у ОС

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
206	Hotkey	Гаряча клавіша	UI	Адаптація	Синонім до Shortcut key
207	Toolbar	Панель інструментів	UI	Калька	Використовується у програмах
208	Status bar	Рядок стану	UI	Калька	Використовується у мобільних додатках
209	Progress bar	Індикатор виконання	UI	Адаптація	Використовується у програмах
210	Loading screen	Екран завантаження	UI	Калька	Використовується у додатках
211	Splash screen	Вітальний екран	UI	Адаптація	Використовується у мобільних додатках
212	Gesture	Жест	Мобільні технології	Еквівалент	Використовується у сенсорних екранах
213	Swipe	Свайп / Проведення	Мобільні технології	Транскрипція / Адаптація	Використовується у мобільних додатках
214	Tap	Торкання	Мобільні технології	Еквівалент	Використовується у сенсорних екранах
215	Pinch	Стиснення	Мобільні технології	Еквівалент	Використовується у сенсорних екранах
216	Zoom in	Збільшити	Мобільні технології	Еквівалент	Використовується у мобільних додатках
217	Zoom out	Зменшити	Мобільні технології	Еквівалент	Використовується у мобільних додатках
218	Emulator	Емулятор	Системи	Транскрипція	Використовується у тестуванні
219	Simulator	Симулятор	Системи	Транскрипція	Використовується у тестуванні
220	Debug mode	Режим налагодження	Системи	Калька	Використовується у програмуванні
221	Safe mode	Безпечний режим	Системи	Калька	Використовується у ОС
222	Recovery mode	Режим відновлення	Системи	Калька	Використовується у ОС
223	Bootloader	Завантажувач	Системи	Адаптація	Використовується у мобільних ОС
224	Firmware	Мікропрограма	Системи	Адаптація	Використовується у пристроях
225	Update package	Пакет оновлення	Системи	Калька	Використовується у ОС

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
226	Release candidate	Кандидат у реліз	Системи	Калька	Використовується у тестуванні
227	Beta version	Бета-версія	Системи	Транскрипція	Використовується у тестуванні

228	Alpha version	Альфа-версія	Системи	Транскрипція	Використовується у тестуванні
229	Stable version	Стабільна версія	Системи	Калька	Використовується у релізах
230	Nightly build	Нічна збірка	Системи	Калька	Використовується у тестуванні
231	Continuous deployment	Безперервне розгортання	DevOps	Калька	Використовується у CI/CD
232	Pipeline	Конвеєр	DevOps	Адаптація	Використовується у CI/CD
233	Artifact	Артефакт	DevOps	Транскрипція	Використовується у CI/CD
234	Log file	Файл журналу	Системи	Калька	Використовується у документації
235	Error log	Журнал помилок	Системи	Калька	Використовується у документації
236	Access log	Журнал доступу	Системи	Калька	Використовується у документації
237	Performance test	Тест продуктивності	Тестування	Калька	Використовується у QA
238	Load test	Тест навантаження	Тестування	Калька	Використовується у QA
239	Stress test	Стрес-тест	Тестування	Транскрипція	Використовується у QA
240	Usability test	Тест зручності використання	Тестування	Описовий переклад	Використовується у UX
241	Accessibility	Доступність	UX	Еквівалент	Використовується у веб-розробці
242	Localization	Локалізація	UX	Транскрипція	Використовується у перекладі
243	Internationalization	Інтернаціоналізація	UX	Транскрипція	Використовується у перекладі
244	Responsive layout	Адаптивне компонування	Веб	Калька	Використовується у веб-розробці
245	Grid system	Сіткова система	Веб	Калька	Використовується у CSS
246	CSS	CSS (каскадні таблиці стилів)	Веб	Латиниця + пояснення	Усталений термін
247	HTML	HTML (мова гіпертекстової розмітки)	Веб	Латиниця + пояснення	Усталений термін
248	JavaScript	JavaScript	Програмування	Латиниця	Назва мови

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
249	TypeScript	TypeScript	Програмування	Латиниця	Назва мови
250	Python	Python	Програмування	Латиниця	Назва мови

251	Push notification	Push-сповіщення	Мобільні технології	Латиниця + адаптація	Використовується у мобільних додатках
252	In-app purchase	Покупка в додатку	Мобільні технології	Адаптація	Використовується у мобільних сервісах
253	Subscription	Підписка	Мобільні технології	Еквівалент	Використовується у сервісах
254	Free trial	Безкоштовний пробний період	Мобільні технології	Адаптація	Використовується у маркетингу
255	User profile	Профіль користувача	UI	Калька	Використовується у додатках
256	Avatar	Аватар	UI	Транскрипція	Усталений термін
257	Dark mode	Темний режим	UI	Калька	Використовується у мобільних додатках
258	Light mode	Світлий режим	UI	Калька	Використовується у мобільних додатках
259	Accessibility settings	Налаштування доступності	UI	Калька	Використовується у ОС
260	Screen reader	Читаць екрана	UX	Адаптація	Використовується у доступності
261	Voice assistant	Голосовий помічник	UX	Калька	Використовується у мобільних ОС
262	Chatbot	Чат-бот	UX	Транскрипція	Використовується у сервісах
263	Cloud storage	Хмарне сховище	Хмари	Калька	Використовується у сервісах
264	File sharing	Спільний доступ до файлів	Хмари	Описовий переклад	Використовується у сервісах
265	Collaboration tool	Інструмент для співпраці	Хмари	Описовий переклад	Використовується у сервісах
266	SaaS	SaaS (програмне забезпечення як послуга)	Хмари	Латиниця + пояснення	Усталений термін
267	PaaS	PaaS (платформа як послуга)	Хмари	Латиниця + пояснення	Використовується у сервісах
268	IaaS	IaaS (інфраструктура як послуга)	Хмари	Латиниця + пояснення	Використовується у сервісах
269	Multi-factor authentication	Багатофакторна автентифікація	Безпека	Калька	Використовується у безпеці

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
270	Password manager	Менеджер паролів	Безпека	Калька	Використовується у безпеці
271	Brute force attack	Атака перебором	Безпека	Адаптація	Використовується у безпеці
272	Keylogger	Кейлогер	Безпека	Транскрипція	Використовується у кіберзлочинах

273	Rootkit	Руткіт	Безпека	Транскрипція	Використовується у кіберзлочинах
274	Trojan horse	Троян	Безпека	Адаптація	Усталений термін
275	Worm	Черв'як	Безпека	Адаптація	Використовується у кіберзлочинах
276	Spyware	Шпигунське ПЗ	Безпека	Адаптація	Використовується у безпеці
277	Adware	Рекламне ПЗ	Безпека	Адаптація	Використовується у безпеці
278	Data encryption	Шифрування даних	Безпека	Калька	Використовується у криптографії
279	Decryption	Розшифрування	Безпека	Еквівалент	Використовується у криптографії
280	Public cloud	Публічна хмара	Хмари	Калька	Використовується у сервісах
281	Private cloud	Приватна хмара	Хмари	Калька	Використовується у сервісах
282	Hybrid cloud	Гібридна хмара	Хмари	Калька	Використовується у сервісах
283	On-premises	Локальне розгортання	Хмари	Адаптація	Використовується у сервісах
284	Scalability test	Тест масштабованості	Тестування	Калька	Використовується у QA
285	Compatibility test	Тест сумісності	Тестування	Калька	Використовується у QA
286	Smoke test	Димовий тест	Тестування	Калька	Використовується у QA
287	Sanity test	Тест адекватності	Тестування	Калька	Використовується у QA
288	End-to-end test	Кінцевий тест	Тестування	Калька	Використовується у QA
289	Acceptance test	Тест прийняття	Тестування	Калька	Використовується у QA
290	Quality assurance	Забезпечення якості	Тестування	Калька	Усталений термін
291	Continuous testing	Безперервне тестування	Тестування	Калька	Використовується у DevOps
292	Mobile app	Мобільний додаток	Мобільні технології	Калька	Усталений термін
293	Web app	Веб-додаток	Веб	Калька	Використовується у веб-розробці

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
294	Native app	Нативний додаток	Мобільні технології	Калька	Використовується у мобільних ОС
295	Hybrid app	Гібридний додаток	Мобільні технології	Калька	Використовується у мобільних ОС

296	Progressive web app (PWA)	Прогресивний веб-додаток	Веб	Калька	Використовується у сучасних технологіях
297	Cross-platform	Кросплатформний	Мобільні технології	Транскрипція	Використовується у розробці
298	Open source	Відкритий код	Програмування	Адаптація	Усталений термін
299	Proprietary software	Пропрієтарне ПЗ	Програмування	Транскрипція	Використовується у ліцензуванні
300	License agreement	Ліцензійна угода	Програмування	Калька	Використовується у документації
301	Domain name	Доменне ім'я	Веб	Калька	Використовується у DNS
302	Hosting	Хостинг	Веб	Транскрипція	Усталений термін
303	Server	Сервер	Веб	Транскрипція	Використовується у мережах
304	Client	Клієнт	Веб	Транскрипція	Використовується у мережах
305	Proxy server	Проксі-сервер	Веб	Транскрипція	Використовується у мережах
306	Load balancer	Балансувальник навантаження	Веб	Адаптація	Використовується у мережах
307	Content delivery network (CDN)	Мережа доставки контенту	Веб	Калька	Використовується у веб-сервісах
308	Firewall rules	Правила брандмауера	Безпека	Калька	Використовується у мережах
309	Intrusion prevention system	Система запобігання вторгненням	Безпека	Калька	Використовується у кіберзахисті
310	Vulnerability	Уразливість	Безпека	Еквівалент	Використовується у безпеці
311	Exploit	Експлойт	Безпека	Транскрипція	Використовується у кіберзлочинах
312	Patch file	Файл патча	Системи	Калька	Використовується у документації
313	Update center	Центр оновлень	Системи	Калька	Використовується у ОС
314	Control panel	Панель керування	Системи	Калька	Використовується у ОС
315	Task manager	Диспетчер завдань	Системи	Адаптація	Використовується у Windows

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
316	Registry	Реєстр	Системи	Еквівалент	Використовується у Windows
317	Command prompt	Командний рядок	Системи	Адаптація	Використовується у Windows

318	Shell	Оболонка	Системи	Еквівалент	Використовується у Linux
319	Bash	Bash	Системи	Латиниця	Назва оболонки
320	PowerShell	PowerShell	Системи	Латиниця	Назва оболонки
321	Script engine	Двигун скриптів	Програмування	Калька	Використовується у веб-розробці
322	Compiler	Компілятор	Програмування	Транскрипція	Усталений термін
323	Interpreter	Інтерпретатор	Програмування	Транскрипція	Використовується у мовах програмування
324	Runtime environment	Середовище виконання	Програмування	Калька	Використовується у Java
325	Garbage collector	Збирач сміття	Програмування	Калька	Використовується у JVM
326	Thread pool	Пул потоків	Програмування	Калька	Використовується у багатозадачності
327	Event handler	Обробник подій	Програмування	Калька	Використовується у UI
328	Callback	Зворотний виклик	Програмування	Калька	Використовується у JS
329	Promise	Обіцянка	Програмування	Калька	Використовується у JS
330	Async/await	Асинхронність/очікування	Програмування	Калька	Використовується у JS
331	Responsive grid	Адаптивна сітка	Веб	Калька	Використовується у CSS
332	Flexbox	Flexbox	Веб	Латиниця	Використовується у CSS
333	Media query	Медіа-запит	Веб	Калька	Використовується у CSS
334	SEO	SEO (оптимізація для пошукових систем)	Веб	Латиниця + пояснення	Використовується у маркетингу
335	Analytics	Аналітика	Веб	Адаптація	Використовується у маркетингу
336	Dashboard metrics	Метрики панелі керування	Веб	Калька	Використовується у BI
337	Data visualization	Візуалізація даних	Веб	Калька	Використовується у BI
338	Heatmap	Теплова карта	Веб	Калька	Використовується у UX
339	Funnel	Воронка	Веб	Адаптація	Використовується у маркетингу

Продовження таблиці 1

№	Англійський термін	Український переклад	Категорія	Стратегія перекладу	Коментар
340	Conversion rate	Коефіцієнт конверсії	Веб	Калька	Використовується у маркетингу

341	Bounce rate	Показник відмов	Веб	Адаптація	Використовується у маркетингу
342	Click-through rate (CTR)	Коефіцієнт кліків	Веб	Калька	Використовується у маркетингу
343	User journey	Шлях користувача	UX	Калька	Використовується у UX
344	Wireframe	Вайрфрейм / Каркас	UX	Транскрипція / Адаптація	Використовується у дизайні
345	Prototype	Прототип	UX	Еквівалент	Використовується у дизайні
346	Mockup	Макет	UX	Адаптація	Використовується у дизайні
347	Usability	Зручність використання	UX	Адаптація	Використовується у UX
348	Accessibility audit	Аудит доступності	UX	Калька	Використовується у UX
349	Responsive typography	Адаптивна типографіка	Веб	Калька	Використовується у дизайні
350	Font rendering	Відображення шрифтів	Веб	Калька	Використовується у браузерах