

*Кушнірова Т. В.,*

*доктор філологічних наук,*

*професор кафедри германської філології та перекладу*

*Національного університету «Полтавська політехніка імені Юрія Кондратюка»*

*orcid.org/0000-0003-3489-2767*

*Павельєва А. К.,*

*кандидат філологічних наук,*

*доцент кафедри германської філології та перекладу*

*Національного університету «Полтавська політехніка імені Юрія Кондратюка»*

*orcid.org/0000-0002-2306-1928*

*Пінчук Є. В.,*

*здобувач другого (магістерського) рівня вищої освіти*

*Національного університету «Полтавська політехніка імені Юрія Кондратюка»*

*orcid.org/0009-0004-0609-5503*

## СТРАТЕГІЇ ЗАБЕЗПЕЧЕННЯ ТЕРМІНОЛОГІЧНОЇ КОНСИСТЕНТНОСТІ ПРИ ПЕРЕКЛАДІ OPEN-SOURCE ДОКУМЕНТАЦІЇ (НА ПРИКЛАДІ GODOT ENGINE)

**Анотація.** У статті досліджуються лінгвістичні та технічні аспекти перекладу англійської документації до проєктів із відкритим вихідним кодом (open-source) українською мовою. На прикладі ігрового рушія Godot Engine проаналізовано специфіку функціонування терміносистеми в умовах краудсорсингової моделі локалізації. Встановлено, що динамічний характер технічних текстів та залучення великої кількості волонтерів до перекладацького процесу зумовлюють ризики термінологічної ентропії та розсинхронізації із супровідною документацією. У роботі ґрунтовно розглянуто концепцію «Docs as Code», що передбачає долучення перекладацької діяльності в інженерний цикл розробки програмного забезпечення. Наявність значної кількості «фальшивих друзів перекладача» у вихідному тексті такого типу, створеному розробниками, вимагає від локалізатора не просто пасивного відтворення, а активної когнітивної верифікації кожного сегмента через звернення до вихідного коду програми. Виявлено типові термінологічні відхилення та обґрунтовано необхідність застосування комплексного підходу до уніфікації фахової лексики через впровадження нормативних глосаріїв та систем попередньої модерації контенту. У роботі ґрунтовно доведено, що для системної нейтралізації термінологічної ентропії необхідно впроваджувати прескриптивні глосарії, які мають статус обов'язкових для всіх учасників проєкту. Такий глосарій має бути не просто списком слів, а інструментом управління знаннями, що фіксує контекст використання кожного терміна. Крім того, важливою є розробка стилістичного довідника (Style Guide). Орієнтуючись на стандарти, закладені у Microsoft Manual of Style, такий довідник дозволить уніфікувати звернення до користувача та правила оформлення елементів інтерфейсу, що є базовою вимогою для професійної технічної комунікації. Констатовано, що лише через чітку регламентацію всіх етапів локалізаційного циклу можна забезпечити високу якість фінального продукту в умовах стихійного краудсорсингу. Запропоновано компетентнісний профіль перекладача-«-

гібридного спеціаліста», здатного ефективно працювати з інструментами автоматизації та системами контролю версій (Git) у межах сучасних локалізаційних платформ.

**Ключові слова:** технічний переклад, локалізація, open-source документація, Godot Engine, термінологічна консистентність, краудсорсинг, Docs as Code.

**Постановка проблеми.** Стрімка цифровізація суспільства та глобальний характер ІТ-індустрії зумовили трансформацію підходів до створення та адаптації технічного контенту [1, с. 42]. Особливе місце в сучасній екосистемі програмного забезпечення посідають проєкти з відкритим вихідним кодом (open-source), які базуються на принципах вільного доступу до знань та колективної розробки. Якість локалізації таких продуктів, і насамперед їхньої документації, виступає вирішальним фактором успішної взаємодії між розробниками та кінцевими користувачами [2, с. 14]. Проте специфіка формування open-source документації, де авторами та перекладачами виступають сотні волонтерів із різним рівнем фахової підготовки, створює складні виклики для збереження термінологічної єдності [3, с. 18]. Відсутність базового централізованого контролю та динамічне оновлення текстів у режимі реального часу, як це відбувається з Godot [4], призводять до появи термінологічних дублетів та системних помилок, що безпосередньо впливає на функціональність та зрозумілість програмного продукту для українського реципієнта.

**Аналіз останніх досліджень і публікацій.** Проблема перекладу технічних текстів та забезпечення термінологічної єдності перебуває у центрі уваги багатьох вітчизняних та зарубіжних науковців. Теоретичні засади технічного перекладу ґрунтовно розроблені у працях В. Карабана, який акцентує увагу на логічності, об'єктивності та однозначності викладу як фундаментальних рисах фахового тексту [5, с. 24]. Питання автоматизації перекладацького процесу та використання

інструментів CAT (Computer-Aided Translation) як засобу підвищення продуктивності детально досліджувалися Л. Боукер [6, с. 65]. Окремим напрямом є вивчення перекладацького краудсорсингу, який М.О'Хаган трактує як трансформацію перекладу із суто професійної послуги на соціальну активність спільноти, що має значні наслідки для цифрової епохи [3, с. 15]. М. Хіменес-Креспо наголошує на важливості розширення меж перекладознавства через вивчення онлайн-колаборацій та впровадження систем модерації для контролю якості краудсорсингових проєктів [7, с. 112]. Попри значну кількість розробок, аспекти локалізації документації ігрових рушіїв у межах стандартів технічної документації та моделі «Docs as Code» [8, с. 45] у вітчизняному перекладознавстві залишаються висвітленими лише частково.

**Виділення невирішених раніше частин загальної проблеми.** Хоча загальні принципи технічного перекладу є ґрунтовно дослідженими, специфіка «Developer English» (мови розробників) та її вплив на українську локалізацію в умовах динамічного оновлення контенту потребують додаткового аналізу. Залишається малодослідженим питання гармонізації зусиль волонтерів-програмістів та професійних лінгвістів для створення цілісної терміносистеми в проєктах із відкритим кодом. Зокрема, це стосується соціально-лінгвістичних аспектів взаємодії у волонтерських спільнотах та специфічних викликів, що виникають при застосуванні краудсорсингових моделей, які часто призводять до неоднорідності тексту [9, с. 34]. Окрім того, відкритим залишається питання технічної інтеграції перекладу в середовище ігрового рушія, де документація є невід'ємною частиною програмного коду.

**Мета статті** полягає у здійсненні системного аналізу проблеми узгодженості термінології в українській локалізації Godot Engine та обґрунтуванні методики її стандартизації з урахуванням концепції «Docs as Code» та специфіки волонтерської праці.

**Виклад основного матеріалу дослідження.** У сучасному цифровому середовищі документація до open-source проєктів (таких як Godot Engine) становить унікальну категорію текстів, що виконує роль комунікаційного містка між розробниками та користувачами. На відміну від комерційного ПЗ, де наповнення створюється професійними технічними кодерами за суворими регламентами, open-source документація формується за принципом спільноти (community-driven).

Ключовою особливістю такого об'єкта перекладу є впровадження концепції «Docs as Code» (документація як код). Вона передбачає, що тексти створюються за тими ж принципами, що й програмний код: зберігаються у репозиторіях систем контролю версій (Git), використовують полегшені формати розмітки (Markdown, reStructuredText) та проходять через автоматизовані системи збірки. Відповідно до сучасних стандартів підготовки технічної документації, такий підхід забезпечує безперервну інтеграцію оновлень у реальному часі, проте він значно підвищує поріг входу для лінгвіста, який фактично стає частиною технічного циклу розробки [8, с. 142]. Це вимагає від перекладача не лише філологічної підготовки, а й навичок технічного редагування, оскільки будь-яка синтаксична помилка у службових тегах може призвести до збою компіляції всієї сторінки документації.

Процес локалізації Godot Engine базується на використанні спеціалізованих хмарних платформ для безперервного

перекладу, зокрема Weblate, що інтегрована безпосередньо з репозиторіями вихідного коду [10]. Це дозволяє реалізувати модель перекладацького краудсорсингу, залучаючи сотні волонтерів до роботи над проєктом одночасно. Як зазначає М. Хіменес-Креспо, онлайн-колаборація у такий спосіб значно прискорює адаптацію продукту, але водночас розмиває традиційні межі професійної відповідальності перекладача [7, с. 56]. Головним ризиком такої моделі є «контекстуальна сліпота» [3, с. 24], оскільки волонтери працюють з окремими сегментами тексту (strings), часто не бачачи загальної картини, що призводить до стилістичного розриву та термінологічної плутанини.

**Структурно-семантичні особливості документації Godot Engine.** Аналіз оригінальної документації ігрового рушія виявляє її гібридну природу: вона є синтезом лінійних оповідних текстів (навчальних посібників) та нелінійних, автоматично згенерованих фрагментів опису програмного інтерфейсу (Class Reference). Така структура детермінована функціональністю продукту: кожен сегмент має бути автономним, щоб користувач міг отримати вичерпну інформацію без читання всього масиву тексту. Це призводить до високої рекурсивності та повторюваності термінологічних конструкцій, що вимагає від перекладача особливої пильності для збереження єдності найменувань у різних розділах.

Важливим визначником структури є використання мови розмітки reStructuredText (reST). Документація насичена службовими символами: подвійними зворотними апострофами для виділення коду, специфічними директивами для перехресних посилань (наприклад, :ref:) та індикаторами типів даних. Будь-яка некоректна модифікація цих елементів у процесі перекладу призводить до критичних помилок компіляції сторінки, що фактично робить лінгвіста відповідальним за технічну валідність продукту.

**Феномен краудсорсингу в перекладі, та як це впливає на термінологічну єдність.** Сучасна практика локалізації програмного забезпечення характеризується фундаментальною зміною парадигм: переходом від традиційної лінійної моделі ТЕР (*Translation, Editing, Proofreading*), де залучені професійні лінгвісти, до моделі колаборативного перекладу, відомої як краудсорсинг [11, с. 18]. У контексті таких open-source проєктів, як Godot Engine, краудсорсинг стає безальтернативним методом локалізації через значні обсяги текстового матеріалу та відсутність бюджету на оплату праці професійних перекладацьких агенцій.

Специфікою перекладацького краудсорсингу є радикальна зміна статусу виконавця. У цій моделі перекладачем виступає не лише дипломований фахівець, а так званий «прозюмер» (*prosumer* – від *producer* + *consumer*) – активний користувач продукту, який одночасно споживає його і бере участь у його вдосконаленні [1, с. 94]. Це має суттєву перевагу: такий перекладач володіє притаманним знанням предметної області, що є критично важливим для адекватної інтерпретації технічних функцій. Як зазначає М. О'Хаган, користувач рушія, який щодня працює із системою рендерингу, часто розуміє технічні нюанси краще за філолога загального профілю [3, с. 14]. Однак відсутність фахової лінгвістичної підготовки у більшості волонтерів (яка, за деякими даними, сягає 68%) створює системний ризик зниження якості тексту [9, с. 22]. Технологічно процес краудсорсингового перекладу базується на деконструкції тексту. На

спеціалізованих вебплатформах, таких як Weblate, вихідний матеріал розбивається на мікрозавдання – окремі «стрічки» (*strings*). Це дозволяє залучити сотні учасників одночасно, але таке роздроблення тексту часто руйнує його смислову цілісність. Виникає явище «контекстуальної сліпоты», коли перекладач бачить ізольований сегмент, не розуміючи його місця в загальній логічній структурі розділу.

З погляду термінологічної консистентності головним викликом є «термінологічна ентропія» [12, с. 27]. У відкритій системі без жорсткої централізованої регуляції кількість варіантів перекладу одного терміна зростає пропорційно до кількості волонтерів. Кожен учасник приносить у проєкт власний ідіолект, що за відсутності примусових механізмів уніфікації призводить до термінологічної розупорядкованості. Як наслідок, виникає ефект «клапикової ковдри» [7, с. 22], де один і той самий технічний концепт у межах однієї сторінки документації може мати кілька різних назв, що суттєво знижує когнітивну якість фінального тексту для реципієнта. Одним із ключових викликів локалізації є якість вихідного тексту, написаного розробниками, для яких англійська мова часто не є рідною. Цей феномен, відомий як «Developer English», характеризується термінологічною нестабільністю та змішуванням стилістичних реєстрів. Його характерними рисами є:

**Стилістична гетерогенність:** поєднання вузькоспеціалізованого жаргону з розмовними конструкціями та ідіомами, що ускладнює вибір відповідного стилістичного реєстра в українській мові [13, с. 11].

**Термінологічна нестабільність:** автори-розробники часто використовують синоніми для позначення одного й того самого технічного концепту (наприклад, одночасне вживання *Asset*, *Resource* та *Item*), що провокує плутанину при автоматизованому перекладі [14, с. 269].

**Синтаксична надмірність або еліптичність:** конструкції, зрозумілі програмісту на рівні коду, можуть бути граматично некоректними або двозначними для лінгвіста [15, с. 1269].

**Семантичний аналіз** виявляє значну кількість випадків полісемії та міжмовної інтерференції. Наприклад, прикметник *original* у словосполученні *original node* часто помилково відтворюється як «оригінальний вузол», хоча нормативним технічним відповідником є «вихідний» або «первинний». Подібна ситуація спостерігається з лексемою *regular* (у контексті *regular polygon* це «правильний багатокутник», а не «регулярний») та словом *invalid* (яке означає «некоректний» або «недійсний», а не «інвалід») [4]. Наявність таких «фальшивих друзів перекладача» у вихідному тексті, створеному розробниками, вимагає від локалізатора не просто пасивного відтворення, а активної когнітивної верифікації кожного сегмента через звернення до вихідного коду програми.

Таким чином, «Developer English» виступає фактором, що зумовлює термінологічну ентропію ще на етапі створення оригіналу, що суттєво ускладнює завдання забезпечення консистентності української локалізації.

**Лексико-граматичні трансформації та способи перекладу терміносистеми Godot Engine.** Вибір адекватної стратегії перекладу технічних термінів є ключовим для забезпечення зрозумілості документації та запобігання когнітивному перевантаженню користувача. На основі аналізу корпусу текстів Godot Engine було виявлено використання основних способів перекладу [11, с. 82]: прямих еквівалентів, транскодування

(транслітерації), калькування та описового перекладу (експлікації).

Транскодування та транслітерація найчастіше застосовуються для новітніх термінів, які ще не мають сталих відповідників в українській мові або стали частиною міжнародного професійного жаргону. Наприклад, термін *asset* часто відтворюється як «асет» (поруч із варіантом «ресурс»), а *shader* – як «шейдер». Згідно з В. Карабаном, такий підхід забезпечує лаконічність тексту та полегшує комунікацію між фахівцями, проте надмірне захоплення транслітерацією може зробити текст надто перевантаженим англіцизмами, що ускладнює сприйняття для початківців [5, с. 118]. Калькування ефективно використовується для термінів зі зрозумілою внутрішньою структурою. Прикладом є переклад *viewport* як «вікно перегляду» або «область перегляду», що дозволяє зберегти семантичну прозорість терміна для українського користувача, орієнтуючи його на функціональне призначення об'єкта.

Особливий інтерес становлять випадки неологізації та морфологічної адаптації, де виникає конфлікт між «академічною нормою» та «практичною доцільністю». Яскравим прикладом є термін *instancing* – унікальна механіка Godot, що дозволяє використовувати сцени як шаблони. У процесі аналізу виявлено дві протилежні стратегії:

**Описовий переклад (Експлікація):** «створення екземпляра сцени». Варіант стилістично бездоганий, але занадто громіздкий для лаконічних технічних інструкцій.

**Калькування з адаптацією: «інстанціювання».** Хоча цей неологізм відсутній у загальних словниках, він ідеально відповідає програмному методу `«.instance()»`, є лаконічним і вже закріпився у професійному середовищі розробників.

Дослідження показало, що головною проблемою є не сам вибір стратегії, а їх хаотичне чергування в межах одного тексту, що безпосередньо веде до термінологічної деградації проєкту [16, с. 215].

Грамматичні трансформації при перекладі документації зумовлені суттєвими відмінностями у будові мов. Специфіка «Developer English» характеризується високою продуктивністю конверсії (вживання іменників у ролі дієслів), що вимагає від перекладача повної структурної перебудови речення при переході до флективної української мови. Крім того, критичним викликом є робота зі змінними підстановки (*placeholders*), такими як «%s». Як зазначає Дж. Ротур'є, перекладач повинен узгодити навколишній текст так, щоб при підстановці будь-якого програмного значення (числа, назви об'єкта тощо) речення залишалося граматично коректним, що часто потребує використання нейтральних конструкцій або зміни порядку слів [17, с. 124].

**Практичні кейси та верифікація стратегій локалізації.** Проведене дослідження підтверджує, що ефективність стратегій забезпечення термінологічної консистентності значною мірою залежить від гнучкості перекладацького підходу до морфологічної адаптації запозичень. Такий підхід дозволяє зберегти лаконічність технічної інструкції та забезпечити пряму відповідність між терміном у документації та методом у програмному коді. Проте аналіз також виявив випадки надмірного калькування, які негативно впливають на сприйняття тексту. Наприклад, спроби буквального перекладу терміна *Scene Tree* як «Сценарне дерево» (що помилково асоціюється зі сценарієм/скриптом) замість контекстуально адекватного «Дерево сцени»

свідчать про ризик втрати логічних зв'язків усередині терміносистеми через ігнорування синтаксичних норм української мови [11, с. 95]. Дослідження практичних кейсів перекладу підсистем рендерингу та скриптування виявило, що найбільша кількість термінологічних девіацій виникає у сегментах, де англійська термінологія базується на метафоричному перекладі загальноживаних слів. У таких випадках «гібридний спеціаліст» має надавати перевагу функціональному еквіваленту, який чітко вказує на роль об'єкта в ігровому русії, навіть якщо це потребує відходу від словникового значення слова.

Таким чином, баланс між мовною чистотою та технічною доцільністю є динамічною величиною, яка має постійно коригуватися через зворотний зв'язок від професійної спільноти розробників, наприклад через платформи GitHub або Discord. Саме така верифікація теоретичних моделей через практичне тестування в реальних проєктах дозволяє сформувати життєздатну терміносистему, здатну адекватно обслуговувати потреби сучасної IT-галузі в Україні. Для системної нейтралізації термінологічної ентропії необхідно впроваджувати прескриптивні глосарії, які мають статус обов'язкових для всіх учасників проєкту. Крім того, важливою є розробка стилістичного довідника (*Style Guide*). Орієнтуючись на стандарти, закладені у *Microsoft Manual of Style*, такий довідник дозволить уніфікувати звернення до користувача та правила оформлення елементів інтерфейсу, що є базовою вимогою для професійної технічної комунікації. Важливо наголосити, що лише через чітку регламентацію всіх етапів локалізаційного циклу можна забезпечити високу якість фінального продукту в умовах стиійного краудсорсингу.

#### Література:

- Cronin M. Translation in the Digital Age. London/New York : Routledge, 2013. 172 p.
- Esselink B. A Practical Guide to Localization. Amsterdam/Philadelphia : John Benjamins Publishing Company, 2000. 488 p.
- O'Hagan M. Community Translation: Translation as a social activity and its possible consequences in the digital age. *Spanish Journal of Applied Linguistics*. 2011. Vol. 24. P. 11–34.
- Godot Engine Documentation. URL: <https://docs.godotengine.org/>
- Карабан В. І. Переклад англійської наукової і технічної літератури. Граматичні труднощі, лексичні, термінологічні та жанрово-стилістичні проблеми. Вінниця : Нова книга, 2004. 576 с.
- Bowker L. Computer-aided translation technology: a practical introduction. Ottawa : University of Ottawa Press, 2002. 211 p.
- Jiménez-Crespo M. A. Crowdsourcing and Online Collaborative Translations: Expanding the Limits of Translation Studies. Amsterdam : John Benjamins, 2017. 226 p.
- Schmitt P. A. Handbuch Technical Documentation. Berlin : Springer, 2011. 534 p.
- Дебела А. С. Особливості краудсорсингу в перекладі. Київ, 2021. 76 с.
- Weblate: Open source continuous localization. URL: <https://weblate.org/>
- Корунець І. В. Теорія і практика перекладу (аспектний переклад). Вінниця : Нова Книга, 2003. 448 с.
- Рудишин, С. Психолого-педагогічні особливості наукового екологічного дискурсу. *Український педагогічний журнал*. 2023. (1), 25–42.
- Gotti M. Investigating Specialized Discourse. Bern : Peter Lang, 2011. 238p.
- Yunker J. Beyond Borders: Web Globalization Strategies. Indianapolis : New Riders, 2002. 576 p.
- Maalej W., Robillard M. P. Patterns of Knowledge in API Reference Documentation. *IEEE Transactions on Software Engineering*. 2013. Vol. 39. No. 9. P. 1264–1282.
- Wright S. E., Budin G. (Eds.). *The Handbook of Terminology Management*. Amsterdam/Philadelphia : John Benjamins, 2001. 915 p.
- Roturier J. *Localizing Apps: A practical guide for translators and translation students*. London : Routledge, 2015. 214 p.

#### Kushnirova T., Pavelieva A., Pinchuk Ye. Strategies for ensuring terminological consistency in the translation of open-source documentation (A case study of Godot Engine)

**Summary.** The article investigates the linguistic and technical aspects of translating English-language open-source documentation into Ukrainian. Using the Godot Engine game engine as a case study, the specificity of the terminological system's functioning within the crowdsourced localization model is analyzed. It is established that the dynamic nature of technical texts and the involvement of a large number of volunteers in the translation process lead to risks of terminological entropy and desynchronization with the accompanying documentation. The presence of a significant number of “false friends of the translator” in source texts of this type, created by developers, requires the localizer to perform active cognitive verification of each segment by referring to the program's source code rather than engaging in passive reproduction. Typical terminological deviations are identified, and the necessity of applying a comprehensive approach to the unification of professional vocabulary through the implementation of regulatory glossaries and content pre-moderation systems is substantiated. The study proves that for the systemic neutralization of terminological entropy, it is necessary to implement prescriptive glossaries that have mandatory status for all project participants. Such a glossary should be not merely a list of words but a knowledge management tool that captures the context of each term's use. Furthermore, the development of a Style Guide is essential. Based on the standards established in the Microsoft Manual of Style, such a guide will allow for the unification of user address forms and formatting rules for interface elements, which is a fundamental requirement for professional technical communication. It is stated that only through the clear regulation of all stages of the localization cycle can the high quality of the final product be ensured in conditions of spontaneous crowdsourcing. The competency profile of a “hybrid specialist” translator is proposed, who is capable of working effectively with automation tools and version control systems (Git) within modern localization platforms.

**Key words:** technical translation, localization, open-source documentation, Godot Engine, terminological consistency, crowdsourcing, Docs as Code, hybrid specialist.

Дата першого надходження статті до видання: 23.02.2026  
Дата прийняття статті до друку після рецензування: 19.03.2026  
Дата публікації (оприлюднення) статті: 29.04.2026