

Документування як інструмент підвищення ефективності управління знаннями в ІТ-командах за умов віддаленої роботи

Олександр Боголій

*аспірант кафедри управлінських технологій,
ВНЗ «Університет економіки та права «КРОК», м. Київ, Україна,
e-mail: boholiom@krok.edu.ua,
ORCID: 0000-0003-0253-667X*

Документація програмного забезпечення є однією з ключових складових процесу розробки та супроводу сучасних ІТ-продуктів. Якісне документування забезпечує збереження знань, узгодженість технічних рішень, ефективність взаємодії між членами команди та прискорює процес внесення змін у складні програмні системи.

Актуальні міжнародні стандарти, зокрема ISO/IEC 6592:20000 та відповідний ДСТУ 4302:2004, визначають підходи до формування, структури та вимог до документації програмних систем. У сучасній ІТ-практиці під терміном «ІТ-документація» розуміють будь-яку зафіксовану інформацію, яка описує архітектуру, функціональність, інфраструктуру, процеси розробки чи операційної діяльності. Вона є важливим інструментом для розробників, тестувальників, менеджерів, DevOps-фахівців та інших зацікавлених сторін, які залучені до постачання та підтримки програмного продукту.

Значення документації зростає в умовах віддаленої та розподіленої роботи, де обмежена синхронна комунікація між учасниками проєкту та ускладнений неформальний обмін інформацією [1]. У таких командах саме документація виконує роль центрального сховища знань, компенсуючи нестачу особистих зустрічей і спонтанних обговорень [2]. Усі ці фактори підвищують необхідність у стандартизованій та доступній документації. Наявність чітко структурованих документів знижує залежність від синхронних зустрічей, забезпечує швидшу адаптацію нових фахівців та сприяє мінімізації ризиків втрати знань у випадку ротації персоналу.

Використання шаблонів на кшталт TDD [3], RFC або ADR-документів[4] допомагає зменшити час підготовки матеріалів, спростити узгодження рішень та полегшити доступ до накопичених знань для нових учасників команди [5]. Такий підхід формує основу для сталої якості документації, яка надалі може бути інтегрована в ширші системи управління знаннями.

Крім того, логічно розвивається застосування інструментів штучного інтелекту, які можуть значно підсилити як якість, так і швидкість документування. Автоматичне генерування описів коду та створення короткої виклад технічних рішень допомагає зменшити навантаження на розробників і аналітиків, мінімізує дублювання зусиль і скорочує час, необхідний для створення або оновлення документації [6]. У результаті процес документування стає більш динамічним і адаптивним.

Таким чином, стандартизована документація програмного забезпечення у

поєднанні з інструментами штучного інтелекту забезпечує ефективне збереження знань, узгодження рішень і покращує взаємодію в ІТ-командах, особливо за віддаленої роботи.

Ключові слова: технічний дизайн, документування, віддалена робота команд, розробка ПЗ, штучний інтелект.

Список використаних джерел

1. R. Reunamäki i C. F. Fey, «Remote agile: Problems, solutions, and pitfalls to avoid», *Bus. Horiz.*, 2022, [Online]. Доступний у: <https://research.aalto.fi/en/publications/remote-agile-problems-solutions-and-pitfalls-to-avoid>
2. C. J. Stettina i E. Kroon, «Is there an agile handover? An empirical study of documentation and project handover practices across agile software teams», в *2013 International Conference on Engineering, Technology and Innovation (ICE) & IEEE International Technology Management Conference*, Чер 2013, с. 1–12. doi: 10.1109/ITMC.2013.7352703.
3. О. Боголій, «ДОКУМЕНТАЦІЯ ТЕХНІЧНОГО ДИЗАЙНУ ПРОГРАМНОГО РІШЕННЯ», 2024. [Online]. Доступний у: <https://conf.krok.edu.ua/ММО/ММО-2024/paper/view/2025>
4. «Architectural Decision Records (ADRs)», *Architectural Decision Records*. Дата звернення: 01, Грудень 2025. [Online]. Доступний у: <https://adr.github.io/>
5. Mage, «How to write technical design docs», *DEV Community*. Листопад 2021. Дата звернення: 24, Червень 2023. [Online]. Доступний у: https://dev.to/mage_ai/how-to-write-technical-design-docs-c02
6. S. S. Dvivedi, V. Vijay, S. L. R. Pujari, S. Lodh, i D. Kumar, «A Comparative Analysis of Large Language Models for Code Documentation Generation», 27, Квітень 2024, arXiv: arXiv:2312.10349. doi: 10.48550/arXiv.2312.10349.hh