

СЕКЦІЯ 4. ПРОБЛЕМАТИКА ІНФОРМАЦІЙНОГО МЕНЕДЖМЕНТУ, ГНУЧКИХ МЕТОДОЛОГІЙ УПРАВЛІННЯ ТА ЕКОНОМІКО-МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

Документація технічного дизайну програмного рішення

Олександр Боголій

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

Останнім часом, розробка програмного забезпечення все частіше відбувається географічно розподіленими командами. Така організація роботи, разом із перевагами, створює також певні складнощі для управлінців. Основні проблеми ефективної розробки програмного забезпечення в розподіленому віддаленому середовищі - це недостатність комунікації, координації, контролю та складний обмін знаннями [1].

Поширеною практикою, серед розробників розподілених команд, що працюють у різних часових поясах, є написання документу з описом технічного дизайну програмного рішення. Такий документ, що зветься в літературі Technical Design Document (TDD), в подальшому використовується для обміну знаннями між членами команди, а також пояснення стратегії впровадження технічного рішення. TDD описує, як вирішити технічну проблему або побудувати функціонал чи його частину [2].

Розробка дизайну технічного рішення зазвичай розпочинається після узгодження функціональних вимог [3]. Написання TDD спонукає інженерів подумати задалегідь про різноманітні проблеми, з якими вони можуть зіткнутися, та знайти рішення для цих проблем, а також проаналізувати можливі “corner-cases” (англ. corner (кут) і case (випадок) - стан системи, що може виникнути за не стандартних умов). Крім цього, TDD також допомагає в ранньому виявленні проблем у запропонованому рішенні, коли ще досить легко внести відповідні зміни.

На рівні організації, такий документ спрощує процедуру погодження технічного рішення, з урахуванням більшості крос-функціональних зауважень, та наповнює бібліотеку технічних рішень, що освоєні в компанії [4].

Більшість організацій зберігають технічну документацію в системах контролю версій, або ж віддають перевагу хмарним офісним пакетам, що дозволяють спільне редагування та мають можливість залишати коментарі та надавати зворотний зв'язок (Google Docs, Dropbox, та інші).

Загалом, написання документу з описом технічного дизайну, включає наступні кроки:

1. Написання чорнового варіанту.
2. Отримання фідбеку від колег. Автор просить учасників команди ознай-

омитись з документом та надати відгуки, які потім аналізуються та готується оновлена версія документу. Крок повторюється кілька разів, поки не отримаємо стабільну версію, де враховані основні зауваження та пропозиції від колег.

3. Отримання фідбеку від основних стейкхолдерів. Автор оновлює документ, враховуючи отримані зауваження та пропозиції. Крок повторюється, поки фінальна версія дизайну не буде затверджена до імплементації.

4. Імплементація технічного рішення. Зазвичай, відбувається у кілька етапів, після кожного з яких, за необхідності, проводиться оновлення технічного дизайну.

5. Ретроспектива. Команда та основні стейкхолдери проводять ретроспективу, де проговорюються проблеми, з якими зіткнулися учасники, та аналізуються рішення, з метою не допущення схожих проблем у майбутньому.

Різні команди розробників мають різні стандарти та конвенції щодо оформлення технічної документації. TDD є неформальним документом, тому організації не дотримуються жорстких вимог до їх вмісту.

В більшості випадків TDD включає наступні секції:

- *Мета* - навіщо потрібно розробляти дане програмне рішення, які проблеми ми ми плануємо вирішити.
- *Вимоги* - що ми очікуємо в результаті успішної імплементації. Тут описуються як функціональні так і нефункціональні вимоги (швидкість обробки запитів, які пристрої повинні підтримуватись тощо).
- *Дизайн імплементації* - детальний опис технічного рішення. Може містити різноманітні діаграми, що описують потоки даних, правила перевірки даних, високорівневу документацію коду тощо.
- *План імплементації* - конкретні задачі з імплементації дизайну.
- *Тестування* - яким чином планується перевіряти чи рішення працює коректно.
- *Експлуатація* - набір інструкцій або операцій необхідних для використання готового рішення.

Корисною практикою є стандартизація структури TDD всередині організації, що сприяє уніфікації процесів розробки в компанії.

Таким чином, написання документу з описом дизайну технічного рішення (Technical Design Document) є поширеною практикою серед розподілених команд, що сприяє ефективному обміну знаннями між членами та налагодженню процесів розробки програмного забезпечення організації.

Ключові слова: технічний дизайн, документація, розподілена розробка ПЗ.

Список використаних джерел

1. Bogolii, O. (2023). Agile Software Development in a Remotely Working Geographically Distributed Team: A Systematic Review. *European Project Management Journal*, 13(1): 23-36. <https://doi.org/10.56889/idnv2224>
2. Mage. (2021, September 14). How to Write Technical Design Docs. *Dev.to*. https://dev.to/mage_ai/how-to-write-technical-design-docs-c02

3. Kasat, Y., & Yadav, J. J. (2015). *The Technical Design Document*. In *Microsoft Dynamics Ax Implementation Guide*. <https://learning.oreilly.com/library/view/microsoft-dynamics-ax/9781785288968/ch07s04.html>
4. Golman, J. (2019, September 12). *Design Docs at Google*. *Industrial Empathy*. Retrieved from <https://www.industrialempathy.com/posts/design-docs-at-google/>