

РОЛЬ ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ У ПІДТРИМЦІ ТА РОЗВИТКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

Досліджено важливість технічної документації у процесі підтримки та подальшого розвитку програмного забезпечення. Технічна документація є критично важливим інструментом для забезпечення ефективної співпраці між розробниками, тестувальниками та кінцевими користувачами. Розглянуто ключові проблеми, пов'язані з її створенням та використанням, а також запропоновано підходи до оптимізації цього процесу.

Ключові слова: технічна документація, підтримка ПЗ, розвиток ПЗ, якість документації.

Abstract

The importance of technical documentation in the process of software support and further development is investigated. Technical documentation is a critical tool for ensuring effective cooperation between developers, testers and end users. The key problems associated with its creation and use are considered, and approaches to optimising this process are proposed.

Keywords: technical documentation, software support, software development, quality of documentation.

Вступ

Програмне забезпечення потребує постійної підтримки та модернізації для відповідності сучасним вимогам. У цьому процесі технічна документація відіграє ключову роль, слугуючи сполучною ланкою між розробниками, адміністраторами та іншими зацікавленими сторонами. Проте недоліки документації або її відсутність часто стають причиною затримок у проєктах, технічних помилок або надмірного витрачання ресурсів. Основною проблемою є забезпечення актуальності, доступності та зрозумілості документації для всіх користувачів.

Проблематика та роль технічної документації

Технічна документація забезпечує зрозумілість функціональності програмного забезпечення для всіх учасників проєкту. Вона є незамінним інструментом для навчання нових членів команди, виявлення помилок та внесення змін до існуючого коду. Необхідно зауважити, що проблеми з документацією є однією із критичних проблем, що є причиною вихідних проблем. Часто створення документації відкладається до завершення розробки, що впливає на її якість або актуальність у майбутньому. Відсутність стандартизації у структурі документації призводить до плутанини або невідповідності. Невідповідність актуальних версій програмного забезпечення і його документації ускладнює підтримку.

Технічна документація відіграє ключову роль у процесі розробки, підтримки та розвитку програмного забезпечення. Вона забезпечує ефективну комунікацію між розробниками, адміністраторами та кінцевими користувачами, а також сприяє надійності продукту.

Необхідно дослідити основні аспекти ролі технічної документації, з якої ми зможемо зробити висновок щодо необхідності ролі існування технічної документації. Документація з вимогами до продукту та архітектури допомагає командам розробників зрозуміти функціональні та нефункціональні вимоги до програмного забезпечення. Документація вихідного коду дорожні карти продукту та тестові документ дозволяють ефективно координувати роботу над кодом і забезпечувати його якість через тестування [1]. Документація для кінцевих користувачів і дизайн інтерфейсу користувача полегшує взаємодію з програмним забезпеченням і знижує кількість помилок під час експлуатації. Документація для системних адміністраторів дозволяє налаштовувати, підтримувати та оптимізувати роботу програмного забезпечення. Планування, оцінка, розклад, звіти та документація за стандартами допомагають керувати процесами розробки та контролювати відповідність вимогам. На рисунку 1 представлено структуру проєктної

документації, які поділяється на три основні категорії: документація продукту, документація користувача та документація про бізнес-процеси.



Рисунок 1 – Структура проєктної документації

Описуючи рисунок, можна виділити та описати три основні категорії:

1. Документація продукту: містить систему документації, яка включає вимоги до продукту, архітектуру, дорожні карти, вихідний код, дизайн інтерфейсу, тестові завдання та технічну підтримку.
 2. Документація користувача: складається з документації для кінцевих користувачів та документації системного адміністратора.
 3. Документація про бізнес-процеси: охоплює базові речі, щоб бути актуальними.
- Ця структура забезпечує всебічне охоплення процесів розробки, підтримки та використання програмного забезпечення, що сприяє його ефективному функціонуванню та розвитку.

Оптимізація створення та використання документації

Автоматизовані інструменти значно спрощують процес створення і підтримки документації [2]. Вони автоматично генерують потрібні матеріали на основі коду або інших джерел, що зменшує людські помилки і заощаджує час. Серед них можна виділити наступні:

1. Javadoc: Використовується для Java-програм. Він автоматично створює HTML-документацію з коментарів у коді. Достатньо просто дотримуватись певних стандартів оформлення коментарів.
2. Sphinx: Потужний інструмент для створення документації в Python. Він дозволяє створювати як HTML, так і PDF-документи. Sphinx підтримує розширення, які дозволяють інтегрувати діаграми, формули, таблиці тощо.
3. Doxygen: Популярний для мов програмування на кшталт C, C++ і Python. Він забезпечує генерування документації у форматах HTML, CHM, PDF. Doxygen підтримує UML-діаграми та інші графічні представлення.

Переваги використання таких інструментів можна виділити такі:

- автоматизація зменшує ручну роботу;
- легкість оновлення документації разом зі змінами в коді;
- підтримка декількох форматів вихідних файлів.

Регулярне оновлення документації під час внесення змін до програмного забезпечення є ключовим моментом для забезпечення її актуальності та ефективності, тому потрібно виділити основні аспекти:

- документація повинна змінюватись паралельно з кодом, що мінімізує розбіжності між ними;
- у циклі розробки програмного забезпечення (SDLC) слід інтегрувати створення та оновлення документації як частину pipeline;

- регулярний перегляд і оновлення існуючих матеріалів дозволяють уникати застарілої інформації.

У результаті можна отримати зниження ризиків через застарілі матеріали та забезпечити розробників актуальною інформацією для підтримки системи.

Створення шаблонів та гайдлайнів для підтримання єдності стилю документації необхідно для уніфікації документації, аби зробити її більш структурованою та зрозумілою для користувачів.

Використання стандартних шаблонів для різних типів документації (API-інструкції, посібники користувача, технічні специфікації) забезпечує стабільність і впізнаваність структури.

Чітко визначені принципи написання тексту, включно з вимогами до стилю, формату та мовних конструкцій, роблять документацію більш читабельною. Інструменти для управління шаблонами дозволяють створювати та ділитися єдиними корпоративними шаблонами. Зібравши до купи усі пункти, створить професійний вигляд документів, покращить взаємодію команди через єдиний стиль та полегшить навчання нових працівників.

У підсумку, оптимізація документації через автоматизації, регулярне оновлення та уніфікацію за допомогою шаблонів і гайдлайнів сприяє підвищенню її якості, актуальності та доступності. Це, своєю чергою, полегшує підтримку та розвиток програмного забезпечення, зменшує витрати часу та ресурсів, а також зменшує кількість помилок у процесі роботи.

Висновок

Технічна документація є невід'ємною складовою розробки програмного забезпечення, яка впливає на його життєвий цикл. Вона сприяє ефективності командної роботи, економії ресурсів та підвищенню якості продукту. Недостатня увага до створення та оновлення документації може призвести до значних проблем у майбутньому. Отже, для забезпечення довговічності та стабільності програмного забезпечення необхідно впроваджувати сучасні підходи до документації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Коновалов В.С., Радоуцький К.Є. Сучасні принципи і методи проектування програмного забезпечення: Конспект лекцій. – Харків: УкрДАЗТ, 2015. – Ч. 2. – 109 с.
2. Петрик М., Мудрик І., Петрик О., Ю. Стоянов. Сучасні технології ООП-проекування та автоматичного генерування програмного коду: навчальний посібник, Тернопіль: ТНТУ імені Івана Пулюя, 2018. 48 с.

Панасюк Володимир Віталійович – студент групи ІПІ-24м, факультет інформаційних технологій і комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: freezerbon@gmail.com

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Panasiuk Volodymyr Vitaliyovych – student of group 1PI-24m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Ukraine.

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua