

ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

У тезах розглядається поняття життєвого циклу програмного забезпечення, його основні етапи та моделі. Визначено значення кожного етапу в процесі розробки програмного продукту, включаючи аналіз вимог, проектування, реалізацію, тестування, супровід та виведення з експлуатації. Проаналізовано популярні моделі життєвого циклу, такі як каскадна, спіральна та гнучкі підходи.

Ключові слова: життєвий цикл, програмне забезпечення, моделі розробки, SDLC, етапи.

Abstract

This paper examines the concept of the software development life cycle, its main stages, and models. Each stage in the software development process is discussed, including requirements analysis, design, implementation, testing, maintenance, and decommissioning. Popular SDLC models such as waterfall, spiral, and agile methodologies are also analyzed.

Keywords: software life cycle, software development, SDLC, development models, stages.

Вступ

Життєвий цикл програмного забезпечення (ЖЦПЗ) — це послідовність етапів, які проходить програмний продукт від моменту виникнення ідеї до повного припинення його використання. Ефективне управління життєвим циклом дозволяє забезпечити якість, стабільність та підтримку ПЗ протягом усього його існування.

Результати дослідження

До основних етапів життєвого циклу ПЗ належать:

1. Identify and Gather Information

Цей етап включає в себе детальне вивчення вимог користувачів та бізнес-цілей, для яких розробляється програмне забезпечення. Це також передбачає збирання та аналізування інформації з різних джерел, включаючи інтерв'ю з зацікавленими сторонами, опитування користувачів, аналіз конкурентів і ринку, а також огляд існуючих рішень. На цьому етапі важливо чітко визначити всі функціональні та нефункціональні вимоги, обмеження, технічні потреби та критерії успіху. Також слід враховувати технологічні тренди, законодавчі вимоги, бюджетні обмеження та терміни.

2. Plan

Планування — це стратегічний етап, на якому визначається, як будуть досягнуті цілі проекту. Включає розробку детального плану проекту, що охоплює етапи, ресурси, бюджет, час, ідентифікацію ризиків і визначення основних етапів виконання. Це також включає створення плану управління ризиками, визначення необхідних інструментів і технологій, а також розподіл задач серед команди. Крім того, важливо визначити методологію розробки (Waterfall, Agile тощо) і способи контролю якості.

3. Design

Дизайн — це етап, на якому створюється архітектура системи та її компоненти. Під час цього етапу відбувається трансформація вимог та ідей у конкретні технологічні рішення. Архітектура програмного забезпечення, проектування баз даних, інтерфейсу користувача, а також розробка прототипів і моделей системи — усе це є частиною етапу дизайну. Важливо врахувати масштабованість, безпеку, інтеграцію з іншими системами, ефективність і простоту підтримки системи в майбутньому. Дизайн визначає фундамент для подальшого кодування, тому він має бути ретельно продуманий.

4. Implement

Реалізація або розробка програмного забезпечення — це етап, коли команда починає писати код на основі розробленого дизайну. Важливо, щоб програмісти суворо дотримувались специфікацій, визначених на етапі планування і дизайну. Тут також реалізуються алгоритми, структури даних, API та інтеграції з іншими системами. Це може включати в себе розробку серверної та клієнтської частин,

створення бази даних, налаштування середовища для тестування та інші операції, необхідні для функціонування ПЗ. Кожен розробник повинен співпрацювати з командою для забезпечення правильності та ефективності реалізації.

5. Test

Тестування — критичний етап, що дозволяє переконатися в тому, що програмне забезпечення відповідає всім вимогам і не має дефектів. Це може включати різноманітні види тестів: юніт-тести, інтеграційні, системні, приймальні тести, тестування безпеки та продуктивності. Під час тестування також важливо перевірити взаємодію системи з іншими програмами та її стабільність при високих навантаженнях. Автоматизоване тестування та CI/CD можуть бути корисними на цьому етапі для прискорення процесу та забезпечення безперервної перевірки якості.

6. Deploy

Етап розгортання включає в себе переміщення розробленого і протестованого програмного забезпечення з середовища розробки в реальне або тестове середовище для використання кінцевими користувачами. Це може бути локальний сервер, хмарна платформа чи інше середовище. На цьому етапі можуть проводитися налаштування, оптимізація, передача даних, і переналаштування системи на вимоги конкретного середовища. Важливо продумати процеси резервного копіювання та відновлення, щоб запобігти втраті даних під час перенесення. Крім того, можна реалізувати безперервне розгортання для забезпечення безперервного оновлення та підтримки актуальності системи.

7. Maintain

Підтримка включає в себе всі операції, які необхідні для того, щоб програмне забезпечення функціонувало належним чином протягом всього його життєвого циклу. Це включає виправлення багів, випуск оновлень і патчів для забезпечення безпеки, а також удосконалення функціональності на основі відгуків користувачів. Підтримка також може включати оптимізацію системи для збереження її ефективності і масштабованості. Це постійний процес, що вимагає моніторингу, тестування та управління конфігураціями для забезпечення безперервної роботи і зручності користувачів.

На рисунку 1 зображено схему життєвого циклу ПЗ.

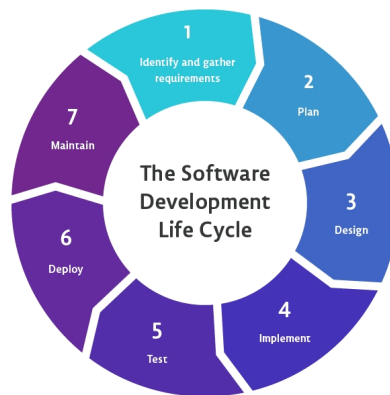


Рисунок 1 – Схема життєвого циклу ПЗ

Кожен етап життєвого циклу програмного забезпечення важливий і тісно пов'язаний з іншими, забезпечуючи успішне виконання проекту від початку до кінця.

Висновки

Життєвий цикл програмного забезпечення є фундаментальним поняттям у галузі ІТ. Правильна організація етапів розробки дозволяє створювати надійні та ефективні програмні продукти. Вибір моделі ЖЦПЗ залежить від потреб проекту, команди та очікуваних результатів. Гнучкі підходи останнім часом набули великої популярності завдяки своїй адаптивності та клієнтоорієнтованості.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sommerville I. Software Engineering. 10th ed. Pearson, 2015.

2. Pressman R. S. Software Engineering: A Practitioner's Approach. 8th ed. McGraw-Hill, 2014.
3. Agile Manifesto: <https://agilemanifesto.org/>
4. IEEE Std 1074-2006 – IEEE Standard for Developing a Software Project Life Cycle Process

Артем Юрійович Білінський – студент ЗПІ-23Б факультету інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: bilinskiyartem110407@gmail.com;

Artem Y. Bilinskyi – Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: bilinskiyartem110407@gmail.com;