

АНАЛІЗ МЕТОДІВ КОНТРОЛЮ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА РІЗНИХ ЕТАПАХ ЖИТТЄВОГО ЦИКЛУ ІНФОРМАЦІЙНИХ СИСТЕМ

Рожко М. О.

Науковий керівник – к.т.н., доц. Міхнова А. В.

Харківський національний університет радіоелектроніки, кафедра ІУС,
м. Харків, Україна

e-mail: maksym.rozhko@nure.ua

This study delves into the software development life cycle, highlighting its stages and quality control measures. Planning involves requirement testing, design incorporates design review, and implementation relies on code review. Testing ensures compliance with requirements, deployment focuses on environment testing, and maintenance utilizes all quality control methods. A formalized description aids effective implementation and improvement of software product quality.

Основою будь-якої інформаційної системи є програмне забезпечення, що являє собою певну сукупність програмних продуктів, які використовуються для обробки інформації з метою автоматизації функціональних задач за допомогою інформаційної системи.

Загальноживаною практикою є представлення процесу розробки програмного забезпечення у вигляді життєвого циклу та його поділ на наступні етапи: планування, проектування, впровадження, тестування, розгортання, обслуговування [1].

Кожний з цих етапів має передбачати контроль якості, що забезпечується використанням тих чи інших методів. Розглянемо етапи життєвого циклу розробки програмного забезпечення та зіставимо їх з методами контролю якості, які у них використовуються.

На етапі планування команда розробників збирає вимоги від кількох зацікавлених сторін, таких як клієнти, внутрішні та зовнішні експерти та менеджери, щоб створити документ специфікації вимог до програмного забезпечення [1].

Для контролю якості на етапі планування використовуються такі методи: тестування вимог, перевірка вимог зацікавленими сторонами (Validation Workshops).

Тестування вимог передбачає перевірку вимог на відповідність властивостям якісних вимог:

- атомарність (вимогу можна вважати атомарною, якщо її не можливо розбити на окремі вимоги без втрати завершеності);
- несуперечність, послідовність (вимога не має містити внутрішні протиріччя та протиріччя іншим вимогам);

– недвозначність (вимога має бути описана без використання жаргону та неоднозначних аббревіатур, повинна допускати лише однозначне розуміння);

– здійсненність (вимога має передбачати реалістичність її здійснення у рамках бюджету та термінів розробки програмного забезпечення).

Основна ціль тестування вимог – знайти вимоги, які не відповідають властивостям якісних вимог, повідомити про них, та проконтролювати їх виправлення.

Перевірка вимог зацікавленими сторонами (Validation Workshops) – метод контролю якості, який полягає у організації семінарів або зустрічей із зацікавленими сторонами для перевірки вимог на відповідність їхнім потребам й очікуванням, гарантуючи, що програмне забезпечення відповідатиме призначеній меті [2].

Основна ціль перевірки вимог зацікавленими сторонами – перевірка та підтвердження відповідності вимог програмного продукту очікуванням та потребам зацікавлених сторін.

На етапі проектування інженери-програмісти аналізують вимоги та визначають найкращі рішення для створення програмного забезпечення.

На етапі проектування використовуються такі методи контролю якості, як огляд дизайну (Design Review) та оцінка прототипу (Prototype Evaluation).

Огляд дизайну – це процес проведення систематичних переглядів проектний документів групою експертів для виявлення та виправлення помилок, невідповідностей і потенційних недоліків дизайну [3].

Основна ціль огляду дизайну – виявити та виправити помилки дизайну.

Оцінка прототипу – створення прототипу системи, щоб дати можливість зацікавленим сторонам спробувати у використанні спрощену версію програмного забезпечення, яке буде розроблятися [4].

Основна ціль оцінки прототипу – виявити проблеми дизайну на ранніх стадіях процесу.

На етапі впровадження команда розробників кодує програмне забезпечення. Вони аналізують вимоги, роблять їх декомпозицію на невеликі завдання, які можна виконувати щодня для досягнення кінцевого результату.

На етапі впровадження найбільш поширений метод контролю якості – це код ревью (code review).

Ціль код ревью – виявити алгоритмічні та логічні помилки в коді.

На етапі тестування для контролю якості програмного забезпечення, використовується тестування програмного забезпечення, яке є процесом перевірки готової частини програмного забезпечення на відповідність відповідним вимогам.

Ціль тестування програмного забезпечення – знайти невідповідності між тим, як працює готова частина програмного забезпечення, та вимогами, які до неї ставляться, виправити невідповідності, перевірити виправлення.

На етапі розгортання використовується такі методи контролю якості: регресійне тестування (Regression Testing), тестування середовища (Environment Validation).

Регресійне тестування – це перевірка кожної частини програмного забезпечення. Ціль регресійного тестування – переконатися, що кожна частина програмного забезпечення працює відповідно до вимог.

Тестування середовища – це перевірка того, що виробниче середовище правильно налаштоване.

Основна ціль тестування середовища – виявити та виправити помилки у налаштуванні середовища.

Етап обслуговування може містити створення нових функцій у програмному забезпеченні, виправлення помилок, тестування виправлень, розгортання виправлень або нових функцій. Це означає, що на цьому етапі необхідно використовувати усі до цього перелічені методи контролю якості: перевірки вимог зацікавленими сторонами, тестування вимог, дизайн рев'ю, оцінка прототипу, код рев'ю, тестування програмного забезпечення, регресійне тестування, тестування середовища.

Отже, проведений аналіз дозволяє зробити висновок, що на кожному етапі життєвого циклу розробки програмного забезпечення можуть бути застосовані певні методи контролю якості. Кожний з цих методів включає в себе певний процес, який займає час та потребує участі різних членів команди.

Прийняття рішення про використання того чи іншого методу контролю якості необхідно робити з урахуванням специфіки проєкту та його обмежень: часу, ресурсів та фінансів.

Список використаних джерел:

1. AWS Amazon What is? : What is SDLC?. URL: https://aws.amazon.com/what-is/sdlc/?nc1=h_ls (дата звернення: 14.02.2024).
2. Linkedin: What are the key deliverables and outcomes of a successful validation workshop? URL: <https://www.linkedin.com/advice/1/what-key-deliverables-outcomes-successful-validation> (дата звернення: 20.02.2024).
3. Chisel glossary : What Is a Design Review? Definition and Process. URL: <https://chisellabs.com/glossary/what-is-a-design-review/> (дата звернення: 15.02.2024).
4. Geeks Gor Geeks : Prototyping Model – Software Engineering. URL: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/> (дата звернення: 15.02.2024).