

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА**

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до проведення практичних занять та виконання самостійної роботи
з навчальної дисципліни

«ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Частина 1

*(для здобувачів 3 курсу першого (бакалаврського) рівня
вищої освіти денної форми навчання
зі спеціальності 126, F6 – Інформаційні системи і технології)*

**Харків
ХНУМГ ім. О. М. Бекетова
2025**

Методичні рекомендації до проведення практичних занять та виконання самостійної роботи з навчальної дисципліни «Якість програмного забезпечення». Частина 1 (для здобувачів 3 курсу першого (бакалаврського) рівня вищої освіти денної форми навчання зі спеціальності 126, F6 – Інформаційні системи і технології) / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова ; уклад. М. В. Булаєнко. – Харків : ХНУМГ ім. О. М. Бекетова, 2025. – 49 с.

Укладач канд. техн. наук, доц. М. В. Булаєнко

Рецензент

О. Б. Костенко, кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та інформаційних технологій Харківського національного університету міського господарства імені О. М. Бекетова

Рекомендовано кафедрою комп'ютерних наук та інформаційних технологій, протокол № 17 від 30.04.2025

ЗМІСТ

ВСТУП	4
Практична робота 1 Аналіз якості програмних продуктів. Сертифікація програмного забезпечення. Стандарти якості	5
Тест 1.....	23
Практична робота 2 Графічні моделі якості. Методи оцінки значень показників якості.....	25
Тест 2.....	35
Практична робота 3 Метрики якості програмного забезпечення.....	38
Тест 3.....	45
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....	48

ВСТУП

Дисципліна «Якість програмного забезпечення» є важливою складовою підготовки фахівців у галузі комп'ютерних наук та інформаційних технологій. Метою її вивчення є формування у студентів знань, умінь і навичок, необхідних для забезпечення якості програмних продуктів на всіх етапах життєвого циклу програмного забезпечення – від вимог і проєктування до тестування, експлуатації та супроводу.

Практичні роботи спрямовані на закріплення теоретичного матеріалу, розвиток аналітичного мислення та оволодіння сучасними підходами до оцінки й забезпечення якості програмного забезпечення. У процесі виконання завдань студенти ознайомлюються зі стандартами, що регламентують якість ПЗ, метриками якості, методами розробки вимог до якості, методиками тестування та принципами управління дефектами.

Методичні рекомендації містять опис тем практичних робіт, порядок їхнього виконання, вимоги до оформлення результатів та критерії оцінювання. Вони сприяють розвитку професійної відповідальності, уважності до деталей, системного підходу до аналізу та вдосконалення програмних продуктів.

Матеріали методичних рекомендацій розраховані на самостійну та аудиторну роботу студентів денної форми навчання.

Практична робота № 1

Аналіз якості програмних продуктів

Сертифікація програмного забезпечення. Стандарти якості

Задача 1

Порівняйте стандарти ISO/IEC 9126 та ISO/IEC 25010. Опишіть їхні основні характеристики та відмінності.

Розв'язок

ISO – International Organization for Standardization (Міжнародна організація зі стандартизації).

IEC – International Electrotechnical Commission (Міжнародна електротехнічна комісія).

Повна назва стандарту:

ISO/IEC 9126: Software engineering – Product quality
(«Інженерія програмного забезпечення – Якість продукту»).

Цей стандарт визначав модель якості ПЗ та методи оцінки його характеристик. Він був прийнятий у 1991 році та оновлений у 2001 році.

ISO/IEC 9126 був першим стандартом для оцінки якості ПЗ. Він складався з чотирьох частин:

1. Модель якості.
2. Зовнішні показники якості.
3. Внутрішні показники якості.
4. Якість в експлуатації.

Стандарт пропонував шість основних характеристик якості ПЗ:

1. Функціональність (Functionality).
2. Надійність (Reliability).
3. Зручність використання (Usability).
4. Ефективність (Efficiency).
5. Супроводжуваність (Maintainability).
6. Переносимість (Portability).

ISO/IEC 25010 був офіційно прийнятий у 2011 році як заміна стандарту ISO/IEC 9126, він розширив модель якості, включивши нові характеристики.

Його повна назва:

ISO/IEC 25010:2011 – Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – System and software quality models.

Цей стандарт визначає модель якості програмного забезпечення та систем, що містить такі характеристики:

1. Функціональна придатність.
2. Продуктивність.
3. Сумісність.
4. Зручність використання.
5. Надійність.
6. Захищеність.
7. Супроводжуваність.
8. Переносимість.

Відмінності.

ISO/IEC 25010 є оновленою версією з більш деталізованими характеристиками якості. Додано нові аспекти, зокрема сумісність та захищеність.

Задача 2

Проаналізуйте етапи сертифікації програмного забезпечення за стандартом ISO/IEC 27001. Опишіть ключові вимоги та процеси.

Розв'язок

ISO/IEC 27001 – це міжнародний стандарт для управління інформаційною безпекою, тоді як ISO/IEC 25010 та ISO/IEC 9126 орієнтовані на якість програмного забезпечення.

Характеристика	ISO/IEC 27001 (Інформаційна безпека)	ISO/IEC 25010 (Якість ПЗ)	ISO/IEC 9126 (Якість ПЗ, застарілий)
Призначення	Управління інформаційною безпекою	Оцінка якості ПЗ	Оцінка якості ПЗ (застарілий)
Основні характеристики	Захист даних, управління ризиками, контроль доступу	Функціональність, продуктивність, безпека, надійність тощо	Функціональність, ефективність, зручність використання, переносимість тощо
Область застосування	Організації та ІТ-системи	Програмне забезпечення та системи	Програмне забезпечення
Рік прийняття	2005 (оновлений у 2013)	2011	1991 (оновлений у 2001, замінений у 2011)
Фокус на безпеку	Основний аспект	Один з аспектів	Один з аспектів
Вимоги до сертифікації	Вимагає проходження аудиту	Немає вимог до сертифікації	Немає вимог до сертифікації

Рисунок 1 – Основні відмінності ISO/IEC 27001 від ISO/IEC 25010 та 9126

Етапи сертифікації програмного забезпечення за стандартом ISO/IEC 27001:

1. **Попередній аналіз** – визначення поточного стану інформаційної безпеки.
2. **Планування** – розробка політик безпеки, визначення ризиків.
3. **Реалізація заходів** – впровадження заходів для мінімізації ризиків.
4. **Внутрішній аудит** – оцінка відповідності організації вимогам стандарту.
5. **Зовнішній аудит** – незалежний сертифікаційний аудит.
6. **Отримання сертифіката** – видача сертифіката на відповідність ISO/IEC 27001.

Задача 3

Дослідження сертифікації за CMMI (Capability Maturity Model Integration). Які рівні зрілості передбачає цей підхід?

Розв'язок

CMMI – це модель зрілості можливостей (Capability Maturity Model Integration), яка використовується для оцінки та покращення процесів розробки програмного забезпечення, управління проєктами та інших бізнес-процесів.

CMMI був вперше випущений у 2000 році Інститутом інженерії програмних систем (SEI – Software Engineering Institute, США).

Основні версії CMMI:

CMMI v1.0 – 2000 рік.

CMMI v1.1 – 2002 рік.

CMMI v1.2 – 2006 рік.

CMMI v1.3 – 2010 рік.

CMMI v2.0 – 2018 рік (найновіша версія, випущена CMMI Institute).

Основна мета CMMI – допомогти організаціям:

1. Визначити рівень зрілості процесів.
2. Оптимізувати управління проєктами.
3. Покращити якість розробки ПЗ.
4. Знизити ризики та дефекти у продуктах.

CMMI (Capability Maturity Model Integration) має 5 рівнів зрілості:

1. **Початковий (Initial)** – хаотичний процес без чітких правил.
2. **Повторюваний (Managed)** – базові процеси впроваджені.
3. **Визначений (Defined)** – стандартизовані процеси.
4. **Керований кількісно (Quantitatively Managed)** – процеси вимірюються.
5. **Оптимізований (Optimizing)** – безперервне вдосконалення процесів.

Задача 4

Розгляньте процес сертифікації за стандартом ISO/IEC 15504 (SPICE). Як він використовується для оцінки якості процесів розробки ПЗ?

Розв'язок

ISO/IEC 15504 (SPICE) – стандарт оцінки процесів розробки ПЗ, також відомий як SPICE (Software Process Improvement and Capability Determination – «Покращення процесів розробки ПЗ та визначення їхньої здатності»), є міжнародним стандартом для оцінки зрілості та якості процесів розробки програмного забезпечення.

Перша версія ISO/IEC 15504 була прийнята у 1998 році.

Остаточне доопрацювання стандарту завершено у 2004 році.

У 2015 році SPICE став основою для розробки нового стандарту ISO/IEC 330xx, який поступово замінює ISO/IEC 15504.

Основна мета ISO/IEC 15504 (SPICE):

1. Визначити рівень зрілості процесів у компанії.
2. Покращити процеси розробки та супроводу ПЗ.
3. Підвищити ефективність управління якістю ПЗ.
4. Використовується для аудиту та сертифікації процесів.

Рівні зрілості в ISO/IEC 15504:

SPICE оцінює процеси за **6-ма рівнями зрілості (0–5)**.

Рівень	Назва	Опис
0	Не виконано (Incomplete)	Процес не виконується або виконується частково.
1	Виконано (Performed)	Процес виконується, але не контролюється.
2	Керовано (Managed)	Процес контрольований, має базові механізми управління.
3	Визначено (Defined)	Процес стандартизований у межах організації.
4	Прогнозовано (Predictable)	Процес вимірюється та керується на основі метрик.
5	Оптимізовано (Optimizing)	Процес постійно вдосконалюється для підвищення ефективності.

Рисунок 2 – Рівні зрілості процесів розробки програмного забезпечення

Сфери застосування ISO/IEC 15504:

1. Розробка програмного забезпечення.
2. Оцінка постачальників ІТ-рішень.
3. Покращення процесів у компаніях.
4. Підготовка до сертифікації за іншими стандартами (ISO 9001, CMMI тощо).

Основні відмінності ISO/IEC 15504 (SPICE) від CMMI наведено на рисунку 3.

Критерій	ISO/IEC 15504 (SPICE)	CMMI
Структура	Гнучка, дозволяє оцінювати окремі процеси	Жорстка, оцінює рівень зрілості всієї організації
Рівні зрілості	0-5 (6 рівнів)	1-5 (5 рівнів)
Фокус	Покращення процесів розробки ПЗ	Загальне управління якістю процесів
Використання	Глобально, особливо в Європі	Більш популярний у США

Рисунок 3 – Порівняння ISO/IEC 15504 (SPICE) та CMMI

Задача 5

Поясніть зв'язок між сертифікацією та тестуванням ПЗ. Які типи тестування використовуються для перевірки відповідності стандартам?

Розв'язок

Тестування допомагає перевірити відповідність вимогам стандартів (ISO/IEC 25010, 9126, ISO/IEC 27001, ISO/IEC 15504, CMMI).

Основні види тестування, які використовуються для перевірки відповідності стандартам:

1. Функціональне тестування.

Перевіряє відповідність функціональних можливостей ПЗ вимогам стандарту.

Використовується для стандартів ISO/IEC 25010 (функціональна придатність).

Приклади: тестування відповідності бізнес-логіки вимогам, тестування API.

API (Application Programming Interface) – Інтерфейс програмування застосунків. Це набір правил, протоколів і інструментів, що дозволяють програмному забезпеченню взаємодіяти між собою. API визначає, як одна програма може запитувати дані або функціональність іншої програми або сервісу.

2. Нефункціональне тестування.

Перевіряє, наскільки система відповідає нефункціональним вимогам стандартів.

2.1 Тестування продуктивності (Performance Testing).

Вимірює швидкість, масштабованість, споживання ресурсів.

Використовується для відповідності стандартам ISO/IEC 25010 (ефективність продуктивності).

Методи: навантажувальне тестування, стрес-тестування, тестування витривалості.

2.2 Тестування безпеки (Security Testing).

Перевіряє захищеність системи від загроз та відповідність стандартам безпеки.

Використовується для ISO/IEC 27001 (інформаційна безпека).

Методи: пенетраційне тестування, тестування управління доступом, криптографічне тестування.

2.3 Тестування зручності використання (Usability Testing).

Перевіряє, чи відповідає інтерфейс стандартам ергономіки та UX.

Використовується для ISO/IEC 25010 (зручність використання).

Методи: тестування за участю користувачів, когнітивне проходження.

2.4 Тестування надійності (Reliability Testing).

Перевіряє стабільність та відмовостійкість програмного забезпечення.

Використовується для ISO/IEC 25010 (надійність, відмовостійкість).

Методи: тестування відмов, тестування відновлення після збою.

2.5 Тестування сумісності (Compatibility Testing).

Перевіряє, чи працює ПЗ на різних платформах та в різних середовищах.

Використовується для ISO/IEC 25010 (сумісність).

Методи: тестування міжплатформної сумісності, тестування браузерної сумісності.

3. Тестування відповідності (Compliance Testing).

Основний тип тестування для сертифікації відповідно до міжнародних стандартів.

Використовується для стандартів ISO/IEC 15504 (SPICE), ISO 9001, CMMI, ISO/IEC 27001.

Включає перевірку наявності необхідних артефактів, звітів та документів. Виконується аудитором або незалежними тестувальниками.

4. Тестування процесів (Process Testing).

Перевіряє, чи процеси розробки та тестування відповідають ISO/IEC 15504 (SPICE), CMMI.

Включає перевірку використання методологій Agile, DevOps, V-модель для відповідності вимогам стандартів.

Використовується в аудитах для сертифікації компанії.

Задача 6

Оцініть якість програмного продукту за моделлю ISO/IEC 25010. Виберіть програмний продукт та проаналізуйте його за ключовими характеристиками моделі.

Розв'язок

Цей стандарт визначає модель якості програмного забезпечення та систем, що містить такі характеристики:

Характеристика	Опис	Приклад оцінки
Функціональна придатність (Functional Suitability)	Наскільки ПЗ виконує необхідні функції.	Чи всі вимоги замовника реалізовані? Чи коректно працюють основні функції?
Продуктивність та ефективність (Performance Efficiency)	Час відгуку, використання ресурсів.	Чи витримує система навантаження? Час відповіді сервера?
Сумісність (Compatibility)	Робота з іншими системами.	Чи працює ПЗ на різних ОС? Чи є підтримка API?
Зручність використання (Usability)	Наскільки ПЗ зручне для користувачів.	Чи є логічний UI/UX? Чи легко навчитися користуватися?
Надійність (Reliability)	Стійкість до збоїв і відновлення.	Чи працює без аварій? Чи швидко відновлюється?
Безпека (Security)	Захист даних та від доступу третіх осіб.	Чи є шифрування? Чи пройдено пенетраційне тестування?
Супроводжуваність (Maintainability)	Простота внесення змін у код.	Чи добре документований код? Чи є модульність?
Переносимість (Portability)	Можливість роботи на різних платформах.	Чи підтримуються різні ОС та браузери?

Рисунок 4 – Характеристики якості стандарту ISO/IEC 25010

Наприклад, оцінюємо браузер Google Chrome:

Характеристика	Оцінка (1-5)
Функціональність	5
Продуктивність	4
Сумісність	5
Зручність використання	5
Надійність	4
Захищеність	5
Супроводжуваність	5
Переносимість	4

Рисунок 5 – Оцінка характеристик якості браузера Google Chrome

Індивідуальне завдання № 1:

- 1) оберіть БУДЬ-ЯКИЙ програмний продукт та проаналізуйте його за ключовими характеристиками моделі ISO/IEC 25010;
- 2) оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Оцініть якість програмного продукту за моделлю ISO/IEC 25010.

Задача 7

Складіть список документів, необхідних для сертифікації ПЗ за стандартом ISO 9001. Визначте, які документи та процеси мають бути підготовлені.

Розв'язок

ISO 9001 – це міжнародний стандарт системи управління якістю (Quality Management System, QMS), він визначає вимоги до процесів управління якістю в організаціях, які прагнуть забезпечити стабільну якість продукції або послуг і підвищити задоволеність клієнтів.

Стандарт ISO 9001 був вперше прийнятий у 1987 році.

Остання діюча версія – ISO 9001:2015.

Для проходження сертифікації за ISO 9001:2015, організація повинна підготувати **систему документації**, яка підтверджує відповідність стандарту. Документація залежить від специфіки компанії, але є **обов'язкові документи** та **додаткові (рекомендовані) документи**:

1. Обов'язкові документи (згідно з ISO 9001:2015).

1.1 Політика якості.

Документ, що визначає загальні принципи та зобов'язання організації щодо якості.

Приклад «Політики якості»:

Політика якості програмного забезпечення

1. **Мета.** Ця політика визначає основні принципи та зобов'язання щодо забезпечення високої якості програмного забезпечення (ПЗ), яке розробляється, тестується та впроваджується в компанії [Назва компанії].
2. **Сфера застосування.** Ця політика поширюється на всі етапи життєвого циклу програмного забезпечення, включаючи планування, розробку, тестування, впровадження та підтримку.

3. Основні принципи:

Орієнтація на клієнта – відповідність вимогам замовника, очікуванням користувачів та регуляторним вимогам.

Безперервне вдосконалення – використання найкращих практик, методологій та технологій для підвищення якості ПЗ.

Управління ризиками – ідентифікація та мінімізація ризиків, що можуть вплинути на якість та безпеку ПЗ.

Автоматизація процесів – використання сучасних інструментів для тестування та контролю якості.

Компетентність персоналу – постійне навчання та розвиток співробітників для підтримки високого рівня якості розробки.

4. Зобов'язання:

Використовувати міжнародні стандарти (ISO 9001, ISO/IEC 25010, ISO/IEC 27001) для забезпечення відповідності якості ПЗ.

Впроваджувати процеси тестування на всіх етапах життєвого циклу ПЗ.

Забезпечувати прозорість процесів розробки та якості через документацію та аудит.

Вчасно реагувати на дефекти та інциденти, що впливають на якість ПЗ.

5. **Контроль і відповідальність.** Керівництво компанії несе відповідальність за реалізацію політики якості, а кожен співробітник – за дотримання стандартів та процедур контролю якості.

6. **Перегляд політики.** Ця політика підлягає перегляду не рідше одного разу на рік або при значних змінах у процесах розробки ПЗ.

Дата затвердження: [Дата] Відповідальна особа: [ПІБ, посада]

1.2 Цілі у сфері якості.

Визначені, вимірювані та досяжні цілі щодо забезпечення якості ПЗ.

1.3 Процедура управління документацією.

Правила створення, контролю, оновлення та зберігання документів.

1.4 Опис основних процесів компанії.

Включає процеси розробки ПЗ, тестування, управління проєктами, підтримки клієнтів тощо.

1.5 Процедура управління невідповідностями та коригувальними діями.

Регламент усунення проблем у процесах розробки та тестування.

1.6 Реєстр ризиків і можливостей.

Аналіз ризиків, які можуть впливати на якість продукту, та стратегії їхнього усунення.

1.7 Записи аудитів внутрішньої системи якості.

Протоколи перевірки відповідності внутрішнім стандартам.

1.8 Записи аналізу керівництвом.

Документація, що підтверджує регулярний аналіз результатів роботи компанії.

1.9 Записи про навчання персоналу.

Підтвердження навчання працівників щодо стандартів якості.

1.10 Записи про оцінку постачальників і партнерів.

Визначення вимог до зовнішніх постачальників (наприклад, хмарних сервісів, субпідрядників).

2. Додаткові (рекомендовані) документи.

2.1 Структура організації та обов'язки персоналу.

Опис ролей та відповідальності працівників.

2.2 Інструкції з розробки та тестування ПЗ.

Алгоритм дій для забезпечення якості програмного забезпечення.

2.3 Керівництво користувача для клієнтів.

Документ з описом функціональності програмного продукту.

2.4 Записи про інциденти та усунення дефектів.

Лог-файли, звіти про помилки, звіти про усунення збоїв у ПЗ.

2.5 Результати тестування.

Протоколи тестування, звіти про дефекти, чеклісти якості.

2.6 Процедура управління змінами.

Як компанія контролює зміни в ПЗ (оновлення, виправлення помилок тощо).

2.7 Документація щодо зворотного зв'язку з клієнтами.

Записи про скарги та пропозиції користувачів щодо покращення ПЗ.

2.8 Політика інформаційної безпеки.

Важливо, якщо ПЗ працює з чутливими даними або персональною інформацією.

2.9 Сертифікати відповідності інших стандартів.

Наприклад, ISO/IEC 27001 (інформаційна безпека) або ISO/IEC 25010 (якість ПЗ).

3. Етапи підготовки до сертифікації ISO 9001:

1. Аналіз поточних процесів – оцінка відповідності стандарту.
2. Розробка та впровадження необхідних документів.
3. Навчання персоналу щодо вимог ISO 9001.
4. Внутрішній аудит – перевірка перед зовнішньою сертифікацією.
5. Зовнішній аудит – оцінка незалежним сертифікаційним органом.
6. Отримання сертифіката ISO 9001.

Задача 8

Проаналізуйте звіт про аудит сертифікації ПЗ. Визначте основні невідповідності та запропонуйте шляхи їхнього усунення.

Розв'язок

Офіційний сайт ISO (<https://www.iso.org/>) – надає документацію щодо стандартів, включаючи шаблони та приклади для сертифікаційних звітів.

Звіт про аудит сертифікації ISO 9001:2025

Компанія: «ABC Software Solutions Ltd».

Дата аудиту: 3–5 березня 2025 року.

Місце проведення: Офіс компанії «ABC Software Solutions», Київ.

Аудиторська команда:

Борейко Іван Олександрович, головний аудитор;

Шевчук Олена Сергіївна, аудитор з якості.

Мета аудиту: Перевірка відповідності системи управління якістю (QMS) компанії «ABC Software Solutions» вимогам стандарту ISO 9001:2015.

1 Коротка характеристика компанії

«ABC Software Solutions Ltd» спеціалізується на розробці програмного забезпечення для автоматизації бізнес-процесів. Компанія надає послуги з розробки, тестування та підтримки програмного забезпечення для корпоративних клієнтів в Україні та за кордоном.

2 Опис проведеного аудиту

Аудит здійснювався за стандартною процедурою відповідно до вимог ISO 9001:2015. Під час перевірки були оцінені:

1. Політика якості: проаналізовано відповідність політики якості організації вимогам ISO 9001:2015.

2. Цілі у сфері якості: перевірено, чи відповідають цілі організації стратегічним напрямкам та вимогам клієнтів.

3. Процеси та документи: оцінка документації, що стосується розробки та тестування ПЗ, а також процесів управління змінами, ризиками та коригувальними діями.

4. Внутрішні аудити: аналіз звітів про внутрішні аудити якості, виконаних у попередньому році.

5. Навчання та кваліфікація персоналу: оцінка відповідності вимогам підготовки співробітників для виконання завдань в межах системи управління якістю.

3 Основні висновки аудиту

3.1 Сильні сторони системи управління якістю

Політика якості: відповідає вимогам ISO 9001:2015, чітко визначена та надає загальні орієнтири для досягнення високої якості продукції та послуг.

Інтегровані процеси: процеси розробки, тестування та підтримки програмного забезпечення є добре організованими та стандартизованими.

Документація: усі основні документи системи управління якістю, включаючи опис процесів, інструкції та керівництва, є актуальними, зберігаються у відповідних системах та регулярно оновлюються.

Управління ризиками: присутній чіткий механізм аналізу та мінімізації ризиків, зокрема при розробці нових версій ПЗ.

Навчання та кваліфікація персоналу: компанія регулярно проводить тренінги для працівників, особливо для тих, хто працює з критичними етапами розробки програмного забезпечення.

3.2 Виявлені невідповідності та рекомендації

1. Невідповідність у документообігу: у деяких старих документах виявлено неактуальну інформацію щодо деяких розроблених інтерфейсів. Рекомендується провести ревізію старих документів і оновити їх відповідно до нових версій продуктів.

2. Планування коригувальних дій: у випадку деяких виявлених дефектів у продукті не були зафіксовані чіткі строки виконання коригувальних дій. Рекомендується впровадити чітку систему термінових коригувальних дій.

3. Не всі співробітники проходили навчання за стандартами ISO 9001: рекомендується провести додаткове навчання співробітників у відділі тестування щодо вимог до управління якістю.

4 Рекомендації щодо поліпшення

1. Оновлення документації: для покращення чіткої відповідності процесів розробки та тестування програмного забезпечення необхідно провести ревізію старих інструкцій і адаптувати їх до поточних вимог ринку та технологій.

2. Зміцнення внутрішніх комунікацій: варто організувати регулярні зустрічі між департаментами для обговорення проблем, що виникають під час тестування програмного забезпечення.

3. Посилення контролю за коригувальними діями: встановлення чітких термінів виконання коригувальних дій та системи моніторингу для своєчасного усунення виявлених дефектів.

5 Загальний висновок

За результатами аудиту компанія «ABC Software Solutions Ltd» продемонструвала високу відповідність вимогам ISO 9001:2015 щодо **системи управління якістю**. Однак для досягнення повної відповідності стандарту рекомендується вжити заходів щодо покращення деяких аспектів документообігу та коригувальних дій. У разі усунення виявлених невідповідностей компанія зможе отримати сертифікацію за стандартом ISO 9001.

Підпис аудиторів:

Борейко Іван Олександрович

Шевчук Олена Сергіївна

Дата складання звіту: 6 березня 2025 року.

Цей приклад є **загальним шаблоном** звіту про аудит сертифікації ПЗ за стандартом ISO 9001. Реальні звіти можуть містити додаткові деталі залежно від специфіки організації та обсягу аудиту.

Задача 9

Оцініть відповідність реального програмного продукту вимогам стандарту ISO/IEC 27001. Проведіть аналіз захисту даних та управління ризиками.

Розв'язок

Приклад оцінки реального програмного продукту вимогам стандарту ISO/IEC 27001:2013 (Система управління інформаційною безпекою).

Оцінка програмного продукту: «SecureCloudApp»

Компанія: XYZ Technologies.

Дата оцінки: 5–7 березня 2025 року.

Програмний продукт: SecureCloudApp (хмарна платформа для зберігання та обміну даними).

Версія: 3.2.1.

Мета оцінки: Перевірка відповідності програмного продукту вимогам стандарту ISO/IEC 27001:2013.

1 Огляд стандарту ISO/IEC 27001:2013

ISO/IEC 27001:2013 – це міжнародний стандарт, що визначає вимоги до **системи управління інформаційною безпекою (ISMS)**, зокрема до політики, процедур, фізичних та технічних контролів для захисту конфіденційності, цілісності та доступності інформації.

Основні аспекти стандарту включають:

- оцінка та управління ризиками;
- управління доступом;
- шифрування та захист даних;
- моніторинг та аудит;
- управління інцидентами безпеки.

2 Ключові вимоги ISO/IEC 27001 і їх відповідність програмному продукту

2.1 Політика безпеки інформації

Вимога: організація повинна мати формалізовану політику безпеки інформації, яка узгоджена з бізнес-цілями і забезпечує захист інформації на всіх етапах її обробки.

Оцінка: програмний продукт **SecureCloudApp** має офіційну політику безпеки інформації, що включає правила управління доступом до даних, вимоги до шифрування та заходи щодо захисту даних користувачів. Політика регулярно оновлюється відповідно до змін у законодавстві та технологіях.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

2.2 Оцінка та управління ризиками

Вимога: оцінка інформаційних ризиків, включаючи технічні, фізичні та організаційні аспекти, має бути проведена для визначення необхідних заходів захисту.

Оцінка: **SecureCloudApp** проводить регулярні оцінки ризиків, що охоплюють потенційні загрози, такі як атаки на сервери, порушення конфіденційності даних та доступ до критичних систем. Ризики оцінюються за допомогою методу **FMEA** (Failure Mode and Effects Analysis), і на основі результатів визначаються відповідні контролю.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

2.3 Управління доступом

Вимога: повинен бути розроблений і впроваджений контроль доступу до інформаційних систем та даних, що включає автентифікацію та авторизацію користувачів.

Оцінка: програмний продукт **SecureCloudApp** використовує механізм багатофакторної автентифікації (MFA) для всіх користувачів, що мають доступ до чутливої інформації. Доступ до даних обмежений на основі ролей користувачів та визначених прав доступу. Для кожної сесії ведеться журнал доступу.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

2.4 Шифрування даних

Вимога: дані, що зберігаються або передаються через мережу, мають бути захищені за допомогою шифрування, щоб уникнути несанкціонованого доступу або зміни даних.

Оцінка: всі дані, що зберігаються в хмарі та передаються між користувачами, зашифровані з використанням сучасних алгоритмів шифрування, таких як **AES-256** для зберігання та **TLS 1.2/1.3** для передачі даних.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

2.5 Управління інцидентами безпеки

Вимога: в організації мають бути розроблені та впроваджені процедури для виявлення, реєстрації та реагування на інциденти безпеки інформації.

Оцінка: програмний продукт має вбудовану систему для моніторингу безпеки, яка виявляє підозрілі дії в режимі реального часу. Всі інциденти, включаючи спроби несанкціонованого доступу або порушення цілісності даних, автоматично реєструються в системі. Для реагування на інциденти створено команду, яка діє відповідно до затвердженого плану.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

2.6 Контроль фізичного доступу

Вимога: фізичний доступ до серверів та обладнання має бути обмежений та контрольований для запобігання несанкціонованому доступу.

Оцінка: обладнання, на якому розміщується **SecureCloudApp**, знаходиться в захищених дата-центрах, які відповідають міжнародним стандартам з безпеки, з обмеженим доступом за допомогою біометричної автентифікації та карток доступу.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

2.7 Моніторинг та аудит

Вимога: потрібно вести моніторинг і аудити для перевірки ефективності впроваджених заходів безпеки.

Оцінка: для моніторингу безпеки програма використовує інтегровану систему аудиту, яка записує всі спроби доступу та дії користувачів. Аудити здійснюються щоквартально, а результати переглядаються менеджерами безпеки.

Відповідність: повна відповідність вимогам ISO/IEC 27001.

3 Загальний висновок

Програмний продукт **SecureCloudApp** демонструє **повну відповідність вимогам стандарту ISO/IEC 27001:2013**. Компанія «XYZ Technologies» успішно реалізувала всі ключові елементи системи управління інформаційною безпекою, що забезпечує захист даних, їхню цілісність і конфіденційність. Рекомендації щодо покращення не виявлено, оскільки програмний продукт вже відповідає найвищим стандартам безпеки інформації.

Рішення: підтримувати та продовжувати вдосконалення системи безпеки відповідно до вимог ISO/IEC 27001.

Цей приклад є **загальним шаблоном** оцінки відповідності програмного продукту стандарту ISO/IEC 27001. Реальна оцінка може включати додаткові деталі залежно від специфіки програмного продукту та організації.

Наприклад, оцінка CRM-системи за критеріями безпеки:

Кодування даних – виконано.

Контроль доступу – виконано.

Захист від атак – частково виконано.

Аудит безпеки – не виконано.

Рекомендації:

1. Запровадити регулярні перевірки безпеки.
2. Покращити механізми аутентифікації.

Задача 10

Створіть чекліст для попередньої оцінки відповідності ПЗ стандарту ISO/IEC 25010. Включіть основні критерії оцінки якості ПЗ.

Розв'язок

Критерій	Відповідає / Не відповідає
Функціональна придатність	✓
Продуктивність	✓
Сумісність	✗
Юзабіліті	✓
Надійність	✓
Захищеність	✗
Супроводжуваність	✓
Переносимість	✓

Рисунок 6 – Приклад чекліста

Приклад чекліста для попередньої оцінки відповідності програмного забезпечення стандарту ISO/IEC 25010

Мета чекліста: Оцінка відповідності програмного продукту вимогам стандарту **ISO/IEC 25010** «Системи якості програмного забезпечення». Чекліст дозволяє перевірити ключові характеристики якості програмного продукту, які визначаються стандартом.

Таблиця 1 – Функціональна придатність (Functional Suitability)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи забезпечує програмний продукт виконання всіх функцій, заявлених у вимогах?	Так / Ні	
2	Чи підтримує програмний продукт правильне виконання кожної функції при різних умовах використання?	Так / Ні	
3	Чи забезпечує програмний продукт функціональну сумісність з іншими системами, якщо це вимагається?	Так / Ні	
4	Чи надає програмний продукт точні та актуальні результати?	Так / Ні	

Таблиця 2 – Придатність для використання (Performance Efficiency)

Ч. ч.	Критерійне	Відповідність	Примітки
1	Чи відповідає програмний продукт вимогам щодо швидкості виконання операцій (відповідність SLA)?	Так / Ні	
2	Чи ефективно використовує програмний продукт ресурси (процесор, пам'ять)?	Так / Ні	
3	Чи здатний продукт обробляти необхідну кількість одночасних запитів / операцій?	Так / Ні	
4	Чи відповідає програмний продукт вимогам щодо часу відгуку в критичних умовах?	Так / Ні	

Таблиця 3 – Сумісність (Compatibility)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи працює програмний продукт на всіх заявлених платформах (ОС, браузерах тощо)?	Так / Ні	
2	Чи забезпечує програмний продукт безперешкодну інтеграцію з іншими системами (якщо потрібно)?	Так / Ні	
3	Чи має програмний продукт можливість автоматичної адаптації до нових версій операційних систем чи інтерфейсів?	Так / Ні	

Таблиця 4 – Зручність (Usability)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи є інтерфейс програмного продукту зручним для користувача?	Так / Ні	
2	Чи легко зрозуміти основні функції та можливості програми без додаткових інструкцій?	Так / Ні	
3	Чи підтримує продукт багатомовність?	Так / Ні	
4	Чи забезпечує продукт належну підтримку для користувачів з обмеженими можливостями?	Так / Ні	

Таблиця 5 – Надійність (Reliability)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи гарантує програмний продукт мінімальний рівень помилок у роботі?	Так / Ні	
2	Чи є система відновлення після збоїв та аварійних ситуацій?	Так / Ні	
3	Чи проводиться тестування продукту на стійкість до несприятливих умов використання?	Так / Ні	
4	Чи виявляються та фіксуються всі критичні помилки в програмному продукті?	Так / Ні	

Таблиця 6 – Безпека (Security)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи захищено програмне забезпечення від несанкціонованого доступу до даних?	Так / Ні	
2	Чи використовуються сучасні методи шифрування для захисту даних?	Так / Ні	
3	Чи є система аутентифікації та авторизації користувачів?	Так / Ні	
4	Чи здійснюється моніторинг безпеки та ведення журналу подій?	Так / Ні	

Таблиця 7 – Підтримка (Maintainability)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи є код продукту документованим і зрозумілим для майбутніх розробників?	Так / Ні	
2	Чи є механізми для легкого оновлення продукту?	Так / Ні	
3	Чи проводиться регулярне тестування продукту для виявлення потенційних проблем у майбутньому?	Так / Ні	
4	Чи забезпечує продукт зручну систему управління версіями?	Так / Ні	

Таблиця 8 – Переносимість (Portability)

Ч. ч.	Критерій	Відповідність	Примітки
1	Чи забезпечує програмний продукт здатність до роботи на різних середовищах (хмарні платформи, різні ОС)?	Так / Ні	
2	Чи підтримує продукт функціональність для міграції на інші платформи без значних втрат у продуктивності?	Так / Ні	
3	Чи є механізм для легкого імпорту та експорту даних з іншими системами?	Так / Ні	

Загальний висновок:**Відповідність вимогам ISO/IEC 25010:**

- повна відповідність;
- часткова відповідність;
- відсутня відповідність.

Рекомендації для покращення якості:

- переглянути / оновити функціональність;
- підвищити ефективність;
- впровадити додаткові заходи безпеки;
- покращити зручність для користувачів.

Цей чекліст дозволяє систематизовано оцінити програмний продукт за основними характеристиками якості, визначеними стандартом **ISO/IEC 25010**. Результати оцінки допоможуть виявити слабкі місця в продукті та вжити заходів для їхнього покращення.

Індивідуальне завдання № 1

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Створіть чекліст для попередньої оцінки відповідності ПЗ стандарту ISO/IEC 25010. Включіть основні критерії оцінки якості ПЗ.

Тест 1

Аналіз якості програмних продуктів. Стандарти якості

Час проходження – 20 хв.

1. Який документ встановлює вимоги до системи управління якістю:

- а) ISO/IEC 27001;
- б) ISO 9001;
- в) IEEE 829?

2. Що з наведеного є моделлю оцінки зрілості процесів:

- а) ISO/IEC 25010;
- б) CMMI;
- в) COBIT?

3. Яка мета сертифікації програмного забезпечення:

- а) підвищення прибутку;
- б) підтвердження відповідності стандартам;
- в) оптимізація алгоритмів?

4. Який міжнародний стандарт описує якість програмного забезпечення:

- а) ISO/IEC 12207;
- б) ISO/IEC 25010;
- в) ISO/IEC 15504?

5. Яка характеристика НЕ входить до моделі якості ISO/IEC 25010:

- а) надійність мережі;
- б) ефективність;
- в) супроводжуваність?

6. Що оцінює характеристика «зручність використання» (usability):

- а) надійність системи;
- б) зручність взаємодії з користувачем;
- в) час виконання?

7. Який документ описує процеси життєвого циклу ПЗ:

- а) ISO/IEC 12207;
- б) ISO/IEC 25010;
- в) ISO 9001?

8. Що таке «ефективність продуктивності» згідно з ISO/IEC 25010:

- а) кількість виявлених дефектів;
- б) час відгуку, використання ресурсів, ємність;
- в) час тестування?

9. Що є результатом процесу сертифікації ПЗ:

- а) звіт про аудит;
- б) специфікація коду;
- в) сертифікат відповідності?

10. Який стандарт визначає функціональну придатність ПЗ:

- а) ISO/IEC 12207;
- б) ISO/IEC 25010;
- в) ISO 31000?

11. До якого типу характеристик належить «супроводжуваність»:

- а) функціональні;
- б) нефункціональні;
- в) безпекові?

12. Що таке СММІ:

- а) методологія тестування;
- б) система управління версіями;
- в) модель зрілості процесів розробки?

13. Який рівень в СММІ є найвищим:

- а) початковий;
- б) керований;
- в) оптимізувальний?

14. Який документ встановлює структуру оцінювання характеристик якості ПЗ:

- а) IEEE 829;
- б) ISO/IEC 25010;
- в) ISO 9241?

15. Що з наведеного НЕ є метою тестування якості:

- а) виявлення дефектів;
- б) шифрування даних;
- в) підтвердження відповідності?

16. Який підхід найбільш прийнятний для оцінки якості великих ПЗ-проектів:

- а) ситуативне тестування;
- б) програмування без тестів;
- в) модель CMMI або SPICE?

17. Що означає термін «сумісність» у контексті якості:

- а) робота ПЗ без помилок;
- б) здатність взаємодіяти з іншими системами;
- в) швидкість виконання?

18. Що вважається внутрішньою якістю ПЗ:

- а) структура коду та архітектура;
- б) відгуки користувачів;
- в) локалізація?

19. Який міжнародний стандарт стосується оцінювання процесів розробки:

- а) ISO/IEC 27001;
- б) ISO/IEC 25010;
- в) ISO/IEC 15504 (SPICE)?

20. Який з наведених документів є стандартом тестування програмного забезпечення:

- а) ISO/IEC 9126;
- б) IEEE 829;
- в) ISO 14001?

Практична робота № 2

Графічні моделі якості. Методи оцінки значень показників якості

1. Графічні моделі якості.

Графічні моделі якості дозволяють візуалізувати взаємозв'язки між різними характеристиками програмного забезпечення. До основних видів таких моделей належать:

- 1) ієрархічна модель якості** (наприклад, модель ISO 9126 або її наступник ISO 25010);
- 2) павутинна діаграма якості** (Spider Chart);
- 3) дерево характеристик якості;**
- 4) діаграма Парето для визначення критичних параметрів якості.**

Задача 1. Ієрархічна модель якості.

Модель якості програмного забезпечення (ISO 9126-1).

Це формалізований підхід до оцінки якості ПЗ, який розподіляє основні характеристики на рівні та підрівні за певними стандартами (наприклад, ISO 9126).

Вона використовується для кількісного аналізу та оцінки якості за певними критеріями, наприклад, за допомогою метрик.

Стандарт ISO 9126-1 визначає **6 основних характеристик** якості ПЗ, кожна з яких розбивається на **підхарактеристики**.

Таблиця 9 – Характеристики якості ПЗ за стандартом ISO 9126-1

Основна характеристика	Підхарактеристики	Опис
Функціональність (Functionality)	– придатність (Suitability); – точність (Accuracy); – взаємодія з іншими системами (Interoperability); – безпека (Security); – дотримання стандартів (Compliance).	Чи виконує ПЗ потрібні функції відповідно до вимог.
Надійність (Reliability)	– зрілість (Maturity); – відмовостійкість (Fault Tolerance); – відновлюваність (Recoverability).	Як ПЗ працює у різних умовах і як воно справляється з помилками.
Зручність використання (Usability)	– зрозумілість (Understandability); – навченість (Learnability); – керованість (Operability).	Наскільки ПЗ просте у використанні та навчанні.
Ефективність (Efficiency)	– часова ефективність (Time Behavior); – використання ресурсів (Resource Utilization).	Чи використовує ПЗ доступні ресурси ефективно.
Супроводжуваність (Maintainability)	– аналізованість (Analyzability); – змінюваність (Changeability); – стабільність (Stability); – тестованість (Testability).	Наскільки легко підтримувати та модифікувати ПЗ.
Переносимість (Portability)	– адаптованість (Adaptability); – встановлюваність (Installability); – замінюваність (Replaceability).	Наскільки ПЗ можна використовувати на різних платформах.



Рисунок 7 – Ієрархічна модель якості (стандарт ISO 9126-1)

Індивідуальне завдання № 2:

побудуйте ієрархічну модель якості програмного забезпечення (ISO 25010).

Задача 2. Побудова павутинної діаграми якості:

1. Виберіть 5 основних характеристик якості ПЗ (наприклад, надійність, продуктивність, зручність використання, безпека, підтримуваність).
2. Присвойте кожній характеристиці значення від 0 до 10.
3. Побудуйте павутинну діаграму, що відображає ці значення.

Інтерпретація павутинної діаграми якості програмного забезпечення:

- а) **розподіл значень:** чим ближче значення до зовнішнього краю діаграми, тим вища якість за відповідною характеристикою;
- б) **баланс характеристик:** якщо діаграма має правильну форму (наближену до кола), це означає, що всі характеристики збалансовані. Якщо ж є сильні відхилення, потрібно звернути увагу на слабкі сторони продукту;
- в) **порівняння різних продуктів:** можна накласти декілька діаграм для різних програмних продуктів і порівняти їхні сильні та слабкі сторони;
- г) **аналіз необхідності покращень:** якщо одна з характеристик значно поступається іншим, це вказує на потребу у вдосконаленні саме в цьому аспекті.

Таблиця 10 – Приклад розрахунків

Характеристика	Значення (0-10)
Надійність	8
Продуктивність	7
Зручність	9
Безпека	6
Підтримуваність	5



Рисунок 8 – Павутинна діаграма

Індивідуальне завдання № 2:

оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Побудуйте павутинну діаграму якості характеристик якості ПЗ (ISO 25010).

Задача 3. Дерево характеристик якості ПЗ (ISO 9126-1).

Дерево характеристик якості ПЗ та ієрархічна модель якості мають схожий підхід до подання якості програмного забезпечення, але вони мають різні концептуальні підходи та рівень деталізації.

Дерево характеристик якості – це **гнучкіший спосіб** подання характеристик якості, що дозволяє візуалізувати взаємозв'язки між різними характеристиками **без жорсткої регламентації**.

Використовується для **концептуального аналізу** якості ПЗ, а не лише для оцінки за певними критеріями.

У дереві кожен рівень відображає **розвиток** певної характеристики, що допомагає **глибше зрозуміти, як вони впливають одна на одну**.

Таблиця 11 – Основні відмінності моделей

Характеристика	Ієрархічна модель якості	Дерево характеристик якості
1	2	3
Підхід	Формалізований, стандартизований	Гнучкий, концептуальний
Структура	Чітка ієрархія рівнів	Дерево взаємозалежностей
Використання	Для стандартизованої оцінки якості	Для аналізу та концептуального представлення
Глибина деталізації	Чітко визначена за стандартами	Гнучка, може бути деталізована довільно
Основний стандарт	ISO/IEC 25010, ISO/IEC 9126	Вільне моделювання
Мета	Кількісна оцінка якості за метриками	Візуалізація характеристик і їхнього впливу

Приклад дерева характеристик якості ПЗ:

Якість ПЗ

1. Функціональність:
 - придатність;
 - зручність;
 - зрозумілість інтерфейсу;
 - інтуїтивність.
2. Надійність:
 - Відмовостійкість;
 - безперервність роботи.
3. Безпека:
 - шифрування даних;
 - контроль доступу.

Індивідуальне завдання № 2.

Побудуйте дерево характеристик якості ПЗ (ISO 25010).

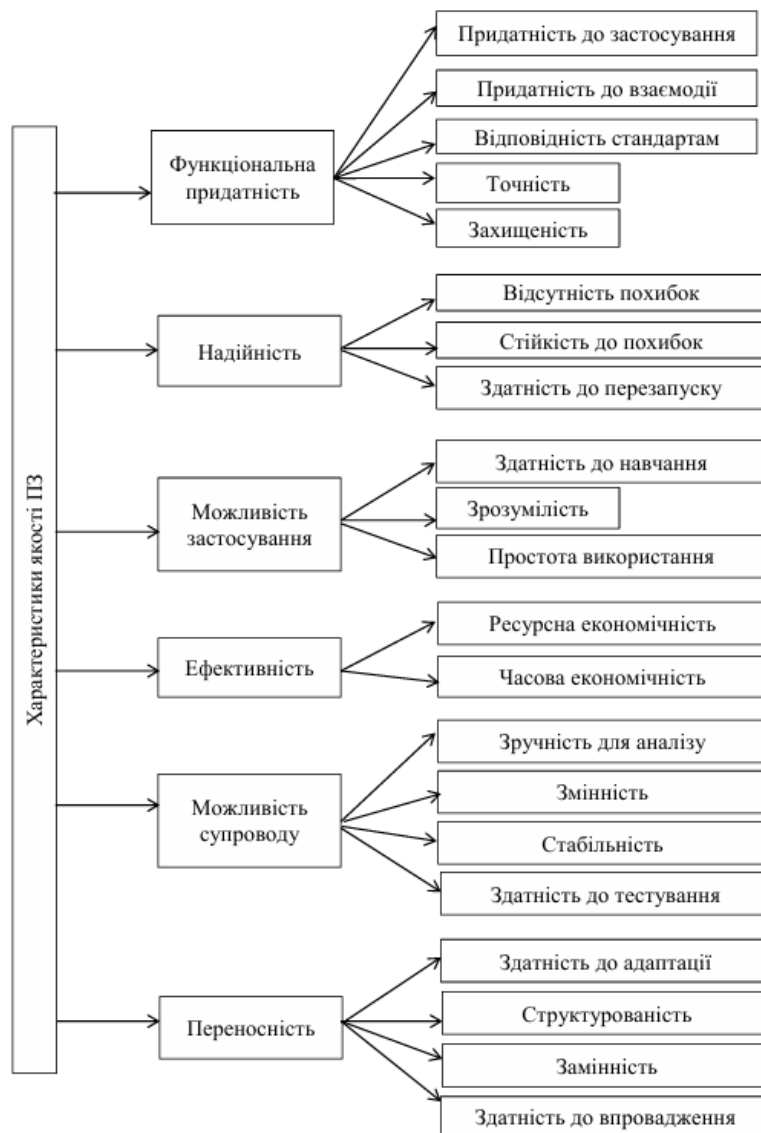


Рисунок 9 – Дерево характеристик якості ПЗ

Задача 4. Діаграма Парето для визначення критичних параметрів якості.

Діаграма Парето – це графічний інструмент, який використовується для визначення найбільш критичних проблем у програмному забезпеченні. Вона ґрунтується на правилі 80/20, згідно з яким 80 % наслідків спричинені 20 % причин.

В аналізі якості ПЗ діаграма Парето допомагає визначити найважливіші проблеми, які впливають на якість. Дозволяє пріоритезувати завдання в тестуванні та усуненні дефектів. Використовується для візуалізації частоти помилок та їхнього впливу.

Побудова діаграми Парето для аналізу дефектів ПЗ

- 1. Збір даних:** визначаємо категорії дефектів (наприклад, проблеми продуктивності, помилки інтерфейсу, вразливості безпеки).
- 2. Обчислення частоти:** підраховуємо, скільки разів кожен тип дефекту зустрічається в аналізованому програмному продукті.
- 3. Сортування:** впорядковуємо дефекти за спаданням частоти.
- 4. Побудова графіка:**

Таблиця 12 – Приклад розрахунків та діаграми

Категорія дефектів	Кількість (шт.)	Накопичуваний відсоток
Помилки інтерфейсу	40	40 %
Вразливості безпеки	30	70 %
Проблеми продуктивності	20	90 %
Логічні помилки	10	100 %

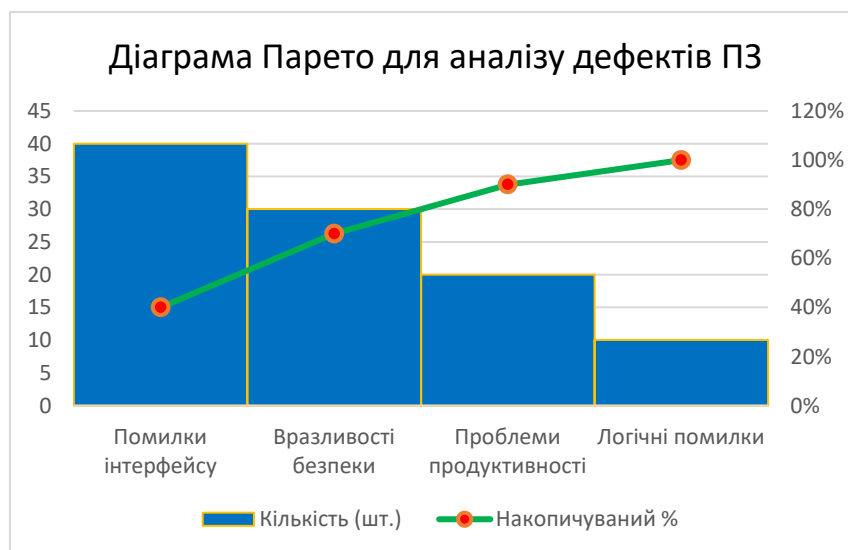


Рисунок 10 – Діаграма Парето

Інтерпретація:

- 1. 40 % усіх дефектів пов'язані з інтерфейсом,** тому їхнє виправлення матиме найбільший ефект.
- 2. Безпека також є критичною,** оскільки її проблеми займають 30 %.
- 3. 90 % проблем пояснюються лише трьома категоріями,** тому їх потрібно вирішувати в першу чергу.

Індивідуальне завдання № 2.

Оберіть **ВЛАСНИЙ** програмний продукт (курсову роботу тощо). Визначте категорії дефектів. Побудуйте діаграму Парето для аналізу дефектів ПЗ.

2. Методи оцінки показників якості.

Оцінка якості програмного забезпечення (ПЗ) є важливим етапом у забезпеченні його ефективності, надійності та відповідності вимогам користувачів.

Існує кілька методів оцінки значень показників якості програмного забезпечення:

- **експертні методи** (метод анкетування, метод Делфі);
- **методи статистичного аналізу** (коефіцієнти кореляції, середнє квадратичне відхилення);
- **методи вимірювання метрик коду** (кількість дефектів, покриття тестами, цикломатична складність);
- **методи аналізу користувацьких відгуків.**

1. Експертні методи.

Включають **суб'єктивні оцінки фахівців**, які аналізують ПЗ за визначеними критеріями.

Метод анкетування.

1. Передбачає розробку спеціальних **анкет** із запитаннями щодо різних аспектів якості.
2. Відповіді збираються від групи **експертів або користувачів.**
3. Після аналізу відповідей формується загальна оцінка якості.

Приклад.

4. Анкета для оцінки якості програмного забезпечення

Інформація про респондента:

Ваша роль у використанні програмного забезпечення:

- ☐ користувач;
- ☐ розробник;
- ☐ тестувальник;
- ☐ інше (уточніть). _____

Скільки часу ви використовуєте це програмне забезпечення:

- ☐ менше одного місяця;
- ☐ 1–3 місяці;
- ☐ 3–6 місяців;
- ☐ більше 6-ти місяців?

Оцінка функціональності (Functional Suitability).

Як ви оцінюєте повноту функцій програмного забезпечення?

(Чи є всі необхідні функції для виконання вашої роботи?):

- ☐ 1 – дуже незадоволений, не вистачає важливих функцій;
- ☐ 2 – задоволений, але не вистачає деяких функцій;
- ☐ 3 – загалом задоволений, хоча деякі функції могли б бути кращими;
- ☐ 4 – задоволений, всі основні функції є;
- ☐ 5 – повністю задоволений, функціональність відповідає всім вимогам?

Як ви оцінюєте коректність виконання основних функцій?

(Чи працює програмне забезпечення без помилок?):

- ☐ 1 – часто виникають помилки;
- ☐ 2 – іноді виникають помилки;
- ☐ 3 – зрідка виникають помилки;
- ☐ 4 – майже не виникають помилки;
- ☐ 5 – взагалі не виникають помилки.

Оцінка юзабіліті (Usability).

Наскільки інтуїтивно зрозумілий інтерфейс програмного забезпечення:

- ☐ 1 – дуже складний, важко зрозуміти;
- ☐ 2 – інтерфейс важкий для розуміння, але можна звикнути;
- ☐ 3 – інтерфейс достатньо зручний, але з окремими проблемами;
- ☐ 4 – інтерфейс зручний, легко працювати;
- ☐ 5 – інтерфейс дуже зручний і інтуїтивно зрозумілий?

Як ви оцінюєте швидкість навчання роботи з програмою:

- ☐ 1 – потрібно багато часу для освоєння;
- ☐ 2 – довго звикати до роботи;
- ☐ 3 – середній час навчання;
- ☐ 4 – швидко освоїти основи;
- ☐ 5 – можна почати працювати майже одразу?

Оцінка продуктивності (Performance Efficiency).

Як ви оцінюєте швидкість виконання основних операцій програмного забезпечення:

- ☐ 1 – дуже повільно, багато затримок;
- ☐ 2 – іноді повільно, затримки;
- ☐ 3 – здебільшого прийнятна швидкість;
- ☐ 4 – швидкість хороша, мінімальні затримки;
- ☐ 5 – дуже швидко, немає затримок?

Чи висуває програма високі вимоги до ресурсів комп'ютера:

- ☐ 1 – програма займає занадто багато ресурсів;
- ☐ 2 – програма може спричинити навантаження на систему;
- ☐ 3 – використання ресурсів середнє;
- ☐ 4 – програма використовує мінімум ресурсів;
- ☐ 5 – програма оптимізована, не використовує зайвих ресурсів?

Оцінка надійності (Reliability).

Як часто програма виходить з ладу або не працює належно:

- ☐ 1 – часто;
- ☐ 2 – іноді;
- ☐ 3 – зрідка;
- ☐ 4 – дуже рідко;
- ☐ 5 – ніколи?

Як ви оцінюєте здатність програми відновлюватися після збоїв:

- ☐ 1 – програма не відновлюється або дуже складно відновлюється;
- ☐ 2 – відновлення можливе, але займає багато часу;
- ☐ 3 – відновлення працює, але є деякі проблеми;
- ☐ 4 – відновлення працює достатньо добре;
- ☐ 5 – програма бездоганно відновлюється після збоїв?

Оцінка безпеки (Security).

Як ви оцінюєте рівень захисту даних у програмному забезпеченні:

- ☐ 1 – дуже низький рівень захисту;
- ☐ 2 – захист даних недостатній;
- ☐ 3 – середній рівень захисту;
- ☐ 4 – високий рівень захисту;
- ☐ 5 – надійний захист даних?

Як ви оцінюєте захист від несанкціонованого доступу:

- ☐ 1 – захист дуже слабкий;
- ☐ 2 – захист не на достатньому рівні;
- ☐ 3 – захист середній;
- ☐ 4 – захист достатньо сильний;
- ☐ 5 – захист надійний?

Оцінка підтримуваності (Maintainability).

Як ви оцінюєте можливість внесення змін до програмного забезпечення:

- ☐ 1 – зміни вносяться дуже важко;
- ☐ 2 – внесення змін займає багато часу;
- ☐ 3 – можна внести зміни, але це непросто;
- ☐ 4 – зміни вносяться легко і швидко;
- ☐ 5 – зміни вносяться дуже швидко і без проблем?

Як ви оцінюєте документацію до програмного забезпечення?

- ☐ 1 – документація відсутня або дуже погана;
- ☐ 2 – документація є, але недостатньо детальна;
- ☐ 3 – документація є, але деякі аспекти залишаються незрозумілими;
- ☐ 4 – документація достатньо повна і зрозуміла;
- ☐ 5 – документація дуже детальна і зрозуміла?

Які ще аспекти якості програмного забезпечення ви вважаєте важливими, але вони не були включені в анкету?

[Відкрите питання]

Індивідуальне завдання № 2.

Оберіть **ВЛАСНИЙ** програмний продукт (курсову роботу тощо). Застосуйте **метод анкетування** для оцінки значень показників якості програмного забезпечення.

Метод Делфі.

1. Використовується для досягнення консенсусу серед експертів.
2. Проводиться **кілька раундів анонімного опитування**.
3. Після кожного етапу експертам надається **зведена інформація** про попередні відповіді, що дозволяє поступово **звужити розбіжності** в оцінках.

Методи статистичного аналізу.

Застосовуються для **кількісної оцінки** якості ПЗ на основі зібраних даних.

Приклади методів:

1. **Коефіцієнти кореляції** – показують взаємозв'язок між різними характеристиками якості.
2. **Середнє квадратичне відхилення** – визначає варіативність оцінок якості.
3. **Аналіз тенденцій** – відстежує зміни в якості протягом часу.

Методи вимірювання метрик коду.

Використовуються для оцінки **внутрішньої якості** ПЗ за допомогою програмних метрик.

Приклади метрик:

1. **Кількість дефектів** – визначає частоту помилок у коді.
2. **Покриття тестами** – відсоток коду, який перевіряється автоматичними тестами.
3. **Цикломатична складність** – міра складності програмного коду.
4. **Щільність коментарів** – співвідношення коментарів до коду.

Методи аналізу користувацьких відгуків.

Зосереджені на зборі та аналізі досвіду **реальних користувачів**.

Основні підходи:

1. **Аналіз відгуків у магазинах додатків** (Google Play, App Store).
2. **Моніторинг звернень у службу підтримки.**
3. **Оцінка Net Promoter Score (NPS)** – показує рівень лояльності користувачів.

Тест 2

Зовнішні характеристики якості. Внутрішні характеристики якості. Взаємозв'язок характеристик. Стандарти якості.

Час проходження – 20 хв.

1. Яка з наведених характеристик належить до зовнішніх характеристик якості ПЗ:

- а) читаність коду;
- б) функціональна придатність;
- в) структурованість коду;
- г) відповідність стилю програмування?

2. Яка з наведених характеристик є внутрішньою характеристикою якості ПЗ:

- а) надійність;
- б) використання ресурсів;
- в) модульність;
- г) юзабіліті?

3. Яка характеристика є спільною для внутрішніх та зовнішніх характеристик якості ПЗ:

- а) безпека;
- б) зручність використання;
- в) модульність;
- г) оцінюваність?

4. Яка характеристика якості ПЗ визначає, наскільки легко можна модифікувати або розширювати програмний код:

- а) супроводжуваність;
- б) ефективність;
- в) функціональна придатність;
- г) використання ресурсів?

5. Який стандарт описує характеристики якості програмного забезпечення:

- а) ISO/IEC 9001;
- б) ISO/IEC 12207;
- в) ISO/IEC 25010;
- г) CMMI?

6. Яка з характеристик якості ПЗ належить до ефективності:

- а) час реакції системи;
- б) взаємодія з іншими системами;
- в) зручність інтерфейсу;
- г) захист від атак?

7. Що означає взаємозв'язок між характеристиками якості ПЗ:

- а) всі характеристики є незалежними одна від одної;
- б) поліпшення однієї характеристики завжди покращує інші;
- в) деякі характеристики можуть впливати одна на одну як позитивно, так і негативно;
- г) взаємозв'язку між характеристиками якості не існує?

8. Яка внутрішня характеристика ПЗ найбільше впливає на його надійність:

- а) модульність;
- б) стійкість до збоїв;
- в) юзабіліті;
- г) сумісність?

9. Як взаємопов'язані продуктивність (ефективність) та зручність використання (юзабіліті):

- а) висока продуктивність завжди означає хорошу юзабіліті;

- б) оптимізація продуктивності може впливати на юзабіліті, наприклад, якщо інтерфейс спрощено;
- в) юзабіліті не залежить від продуктивності;
- г) чим гірша юзабіліті, тим вища продуктивність?

10. Який термін описує здатність програмного забезпечення працювати на різних платформах без змін у коді:

- а) надійність;
- б) переносимість;
- в) функціональна придатність;
- г) ефективність?

11. Який міжнародний стандарт встановлює вимоги до систем управління якістю:

- а) ISO/IEC 25010;
- б) ISO 9001;
- в) CMMI;
- г) ISO/IEC 27001?

12. Який з цих стандартів є основним для інформаційної безпеки:

- а) ISO/IEC 15504;
- б) ISO/IEC 12207;
- в) ISO/IEC 27001;
- г) ISO/IEC 90003?

13. Яка основна мета стандарту CMMI:

- а) забезпечення управління якістю ПЗ;
- б) оцінка та покращення процесів розробки ПЗ;
- в) визначення вимог до інформаційної безпеки;
- г) сертифікація компаній з розробки програмного забезпечення?

14. Що регулює стандарт ISO/IEC 15504 (SPICE):

- а) управління ризиками;
- б) процеси розробки програмного забезпечення;
- в) оцінку безпеки програмних продуктів;
- г) вимоги до апаратного забезпечення?

15. Який стандарт є основним для оцінки відповідності якості ПЗ у межах розробки:

- а) ISO/IEC 12207;
- б) ISO/IEC 25010;
- в) ISO 9001;
- г) CMMI?

16. Що називають рівнями зрілості в CMMI:

- а) етапи оцінки безпеки ПЗ;
- б) показники відповідності якості ПЗ;

- в) п'ятиступенева модель удосконалення процесів;
- г) методи тестування ПЗ?

17. Який стандарт містить моделі якості для оцінки життєвого циклу ПЗ:

- а) ISO/IEC 12207;
- б) ISO 9001;
- в) ISO/IEC 27001;
- г) ISO/IEC 15504?

18. Яке з тверджень щодо ISO/IEC 25010 є правильним:

- а) стандарт описує вимоги до процесів розробки;
- б) стандарт визначає характеристики якості програмного забезпечення;
- в) стандарт застосовується лише до безпекових систем;
- г) стандарт використовується для управління персоналом?

19. Яким є основне призначення сертифікації за ISO 9001:

- а) оцінка рівня безпеки компанії;
- б) перевірка якості внутрішніх процесів управління та розробки;
- в) оцінка відповідності програмного забезпечення міжнародним вимогам;
- г) сертифікація ПЗ для державних установ?

20. Яка з характеристик ISO/IEC 25010 стосується здатності ПЗ працювати в різних умовах експлуатації:

- а) функціональна придатність;
- б) сумісність;
- в) надійність;
- г) ефективність?

Практична робота № 3

Метрики якості програмного забезпечення

Метрики якості програмного забезпечення поділяються на кілька основних категорій:

- 1) метрики коду;
- 2) продуктивності;
- 3) надійності;
- 4) супроводжуваності;
- 5) метрики тестування.

Наведемо основні метрики разом із формулами для їхнього обчислення.

1 Метрики коду (Code Metrics)

1.1 Щільність коментарів (Comment Density, CD)

Оцінює рівень документованості коду.

$$CD = \frac{N_C}{N_L} \cdot 100 \% , \quad (1)$$

де N_C – кількість рядків коментарів,

N_L – загальна кількість рядків коду.

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Оцініть рівень документованості коду.

1.2 Цикломатична складність (McCabe's Cyclomatic Complexity, CC)

Визначає мінімальну кількість тестів, необхідних для повного покриття коду.

Цикломатичну складність можна розрахувати за допомогою графів потоку керування або відносно функцій, модулів, методів чи класів у програмі.

$$CC = E - N + 2P , \quad (2)$$

де E – кількість ребер у графі потоку керування,

N – кількість вершин у графі,

P – кількість компонент (зазвичай $P = 1$ для однієї програми).

Або

$$CC = P + 1 , \quad (3)$$

де P – кількість предикатних вузлів (операторів, умовних операторів (if / else if / else)).

Задача 1

Функція містить 10 умовних операторів (if, while, for). Оцінити складності коду за метрикою цикломатичної складності (CC).

Розв'язок

$$CC = P + 1 ,$$

де P – кількість предикатних вузлів (вузол, який містить умову).

$$CC = \text{Кількість умовних операторів} + 1 = 10 + 1 = 11.$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Оцініть складності коду за метрикою цикломатичної складності.

1.3 Щільність повторного використання коду (Code Reusability, CR)

$$CR = \frac{N_R}{N_T} \cdot 100 \% \quad (4)$$

де N_R – кількість повторно використаних модулів (рядків);

N_T – загальна кількість модулів (рядків, LOC).

Задача 2

Загальний обсяг коду проєкту – 50 000 LOC , з яких 20 000 LOC – це повторно використаний код. Обчисліть коефіцієнт повторного використання.

Розв'язок

$$N_R = 20\,000,$$

$$N_T = 50\,000$$

$$CR = \frac{N_R}{N_T} \cdot 100 \% = \frac{20\,000}{50\,000} \cdot 100 \% = 40 \% .$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Обчисліть коефіцієнт повторного використання складності коду.

2. Метрики продуктивності (Productivity Metrics).

2.1. Час відгуку (Response Time, RT).

Вимірює час між відправленням запиту і отриманням відповіді:

$$RT = T_{end} - T_{start} \quad (5)$$

де T_{end} – час отримання відповіді;

T_{start} – час надсилання запиту.

2.2 Пропускна здатність (Throughput, TP)

Кількість запитів, які система може обробити за одиницю часу:

$$TP = \frac{N_R}{T}, \quad (6)$$

де N_R – кількість запитів;

T – час виконання (секунди, хвилини тощо).

2.3 Продуктивність розробки (Development Productivity, DP)

$$DP = \frac{N_{LOC}}{T}, \quad (7)$$

де N_{LOC} – кількість написаних рядків коду;

T – час у днях.

Задача 3

Команда розробників реалізувала 1 500 LOC за 3 тижні. Обчисліть продуктивність команди в LOC на день.

Розв'язок

$$DP = \frac{N_{LOC}}{T} = \frac{1\,500}{3 \times 5} = 100 \frac{LOC}{\text{день}}$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Обчисліть продуктивність розробки.

3 Метрики надійності (Reliability Metrics)

3.1 Середній час між відмови (Mean Time Between Failures, MTBF)

Середній час між відмовами системи:

$$MTBF = \frac{\sum T_{up}}{N_f}, \quad (8)$$

де T_{up} – загальний час безвідмовної роботи;

N_f – кількість відмов.

Задача 4

Система за останні 1 000 годин роботи зазнала 5 відмов. Обчисліть середній час між відмовами.

Розв'язок

$$MTBF = \frac{\sum T_{up}}{N_f} = \frac{1\,000}{5} = 200 \text{ год.}$$

3.2 Середній час виправлення дефекту (Mean Time To Repair, MTTR)

Визначає середній час, необхідний для усунення відмови:

$$MTTR = \frac{T_{repair}}{N_f}, \quad (9)$$

де T_{repair} – загальний час виправлення дефектів;

N_f – кількість відмов.

Задача 5

Для виправлення 10 дефектів команда витратила 50 годин. Обчисліть середній час виправлення одного дефекту.

Розв'язок

$$MTTR = \frac{T_{repair}}{N_f} = \frac{50}{10} = 5 \text{ год.}$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Обчисліть середній час виправлення одного дефекту.

3.3 Імовірність безвідмовної роботи (Reliability, $R(t)$)

Визначає імовірність того, що система працюватиме без збоїв у певний момент часу:

$$R(t) = e^{-\lambda t}, \quad (10)$$

де $\lambda = \frac{1}{MTBF}$ – інтенсивність відмов;

t – час експлуатації.

3.4 Щільність дефектів (Defect Density, DD)

Оцінює кількість дефектів на одиницю коду.

$$DD = \frac{N_D}{N_{LOC}}, \quad (11)$$

де N_D – кількість виявлених дефектів;

N_{LOC} – кількість рядків коду.

Задача 6

Програмний модуль містить 2 500 рядків коду (LOC – Lines of Code) і має 50 виявлених дефектів. Обчисліть коефіцієнт дефектності.

Розв'язок

$$DD = \frac{N_D}{N_{LOC}} = \frac{50}{2\,500} = 0,02 \text{ дефектів на LOC.}$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Обчисліть щільність дефектів.

4. Метрики супроводжуваності (Maintainability Metrics)

4.1 Індекс підтримуваності (Maintainability Index, MI)

Оцінює легкість супроводу програмного коду:

$$MI = 171 - 5.2 \cdot \ln(AVG_V) - 0.23 \cdot CC - 16.2 \cdot \ln(LOC), \quad (12)$$

де AVG_V – середня кількість змінних у функціях;

CC – цикломатична складність;

N_{LOC} – кількість рядків коду.

Задача 7

Обчислити індекс підтримуваності (MI), якщо відомо:

- середня кількість змінних у функціях $AVG_V = 10$;
- цикломатична складність $CC = 15$;
- $LOC = 500$.

Розв'язок

Формула для індексу підтримуваності:

$$MI = 171 - 5.2 \cdot \ln(10) - 0.23 \cdot 15 - 16.2 \cdot \ln(500) = 171 - 11.96 - 3.45 - 100.44 = 55.15$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Обчисліть індекс підтримуваності.

4.2 Коефіцієнт підтримуваності (Maintainability Ratio)

$$K_{\Pi} = \frac{\text{Кількість коду, що відповідає стандартам}}{\text{Загальна кількість коду}} \cdot 100 \% \quad (13)$$

Задача 7

У проєкті 80 % коду написано згідно з код-стилем, 20 % – не відповідає стандартам. Обчисліть коефіцієнт підтримуваності.

Розв'язок

$$\begin{aligned} K_{\Pi} &= \frac{\text{Кількість коду, що відповідає стандартам}}{\text{Загальна кількість коду}} \cdot 100 \% = \frac{80}{100} \cdot 100 \% \\ &= 80 \% . \end{aligned}$$

4.3 Коефіцієнт стабільності коду (Stability Index, SI)

$$Stability\ Index = \frac{\text{Кількість стабільних класів}}{\text{Загальна кількість класів}} \cdot 100 \%. \quad (14)$$

Задача 8

За останній реліз зміни торкнулися 500 рядків з 5 000. Обчисліть стабільність коду.

Розв'язок

$$Stability\ Index = \frac{5\,000 - 500}{5\,000} \cdot 100 \% = \frac{4\,500}{5\,000} \cdot 100 \% = 90 \%.$$

Індивідуальне завдання № 3

Оберіть ВЛАСНИЙ програмний продукт (курсову роботу тощо). Обчисліть коефіцієнт стабільності коду.

5 Метрики тестування (Testing Metrics)

5.1 Покриття коду тестами (Code Coverage, CCov)

$$CCov = \frac{N_T}{N_L} \cdot 100 \%, \quad (15)$$

де N_T – кількість рядків коду, що були виконані тестами;

N_L – загальна кількість рядків коду.

Задача 9

Всього в програмному модулі 1 200 рядків коду. Написані тести покривають 900 рядків. Обчисліть рівень покриття тестами.

Розв'язок

$$CCov = \frac{N_T}{N_L} \cdot 100 \% = \frac{900}{1\,200} \cdot 100 \% = 75 \%.$$

5.2 Відсоток успішних тестів (Test Pass Rate, TPR)

$$TPR = \frac{N_P}{N_T} \cdot 100 \%, \quad (16)$$

де N_P – кількість успішних тестів;

N_T – загальна кількість тестів.

Тест 3

Метрики якості програмного забезпечення

Час проходження – 20 хв.

1. Що таке метрика якості програмного забезпечення:

- а) числовий показник, що вимірює певну характеристику якості;
- б) стандарт розробки програмного забезпечення;
- в) процес тестування ПЗ;
- г) методологія проектування ПЗ?

2. Яка з наведених характеристик не є метрикою якості ПЗ:

- а) час відгуку системи;
- б) кількість помилок на 1 000 рядків коду;
- в) рівень задоволеності користувачів;
- г) кількість співробітників у команді розробки?

3. Який основний стандарт містить класифікацію метрик якості програмного забезпечення:

- г) ISO/IEC 9001;
- б) ISO/IEC 25010;
- в) ISO/IEC 27001;
- г) CMMI?

4. Які виокремлюють групи метрик якості:

- а) тільки зовнішні;
- б) внутрішні, зовнішні та метрики якості використання;
- в) лише метрики продуктивності;
- г) тільки метрики безпеки?

5. Що є основною метою використання метрик якості ПЗ:

- а) автоматизація процесу розробки;
- б) оцінка та покращення характеристик якості програмного продукту;
- в) мінімізація витрат на розробку;
- г) визначення рівня компетенції розробників?

6. Яка метрика використовується для оцінки продуктивності ПЗ:

- а) середній час відгуку;
- б) число помилок у коді;
- в) час завершення проєкту;
- г) кількість реалізованих функцій?

7. Що таке узагальнена модель якості:

- а) модель, що визначає загальні підходи до вимірювання якості ПЗ;
- б) модель фінансової ефективності розробки;
- в) опис методів тестування ПЗ;
- г) методологія управління персоналом?

8. Яка модель є однією з найвідоміших узагальнених моделей якості:

- а) модель ISO 9001;
- б) модель Боема – МакКолла;
- в) модель CMMI;
- г) модель SPICE?

9. Яка з наведених метрик є внутрішньою:

- а) час реакції системи;
- б) кількість коментарів у коді;
- в) кількість задоволених користувачів;
- г) споживання оперативної пам'яті?

10. Яка з наведених моделей є графічною моделлю якості:

- а) модель якості Бейлі – Бейєра;
- б) модель оцінки ризиків;
- в) модель SCRUM;
- г) каскадна модель розробки?

11. Яка з графічних моделей якості представляє взаємозв'язок між показниками у вигляді ієрархії:

- а) дерево якості;
- б) діаграма Ганта;
- в) спіральна модель;
- г) модель COCOMO?

12. Що є основним методом оцінки значень показників якості:

- а) інтуїтивне оцінювання;
- б) математичний аналіз;
- в) оцінка на основі рейтингу розробників;
- г) опитування клієнтів?

13. Який метод оцінки якості ґрунтується на зборі статистичних даних під час тестування:

- а) метод експертного оцінювання;
- б) метод статистичного аналізу;
- в) метод SWOT-аналізу;
- г) метод метрик процесів?

14. Який підхід застосовується для оцінки зовнішніх характеристик якості ПЗ:

- а) аналіз вихідного коду;
- б) оцінка взаємодії користувача із системою;
- в) аналіз архітектури;
- г) оцінка якості коментарів у коді?

15. Яка з наведених метрик відноситься до якості використання:

- а) час виконання запитів;
- б) рівень задоволеності користувачів;
- в) кількість класів у програмному модулі;
- г) кількість рядків коду?

16. Яке співвідношення між внутрішніми та зовнішніми метриками якості:

- а) внутрішні метрики безпосередньо впливають на зовнішні характеристики;
- б) внутрішні метрики не пов'язані із зовнішніми;
- в) зовнішні метрики визначають внутрішні;
- г) внутрішні метрики важливі лише для команди розробки?

17. Яка характеристика не оцінюється у графічних моделях якості:

- а) надійність;
- б) структурованість коду;
- в) фінансова ефективність розробки;
- г) безпека?

18. Який стандарт використовується для оцінки процесів розробки ПЗ:

- а) ISO/IEC 15504 (SPICE);
- б) ISO/IEC 25010;
- в) ISO/IEC 27001;
- г) ISO 9001?

19. Яка графічна модель використовується для оцінки взаємозв'язків між характеристиками якості:

- а) дерево якості;
- б) діаграма потоків даних;
- в) діаграма класів;
- г) модель водоспадного процесу?

20. Що є ключовою перевагою використання метрик якості ПЗ:

- а) зниження витрат на тестування;
- б) об'єктивне вимірювання якості та виявлення проблем;
- в) мінімізація часу на розробку;
- г) спрощення процесу програмування?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Якість програмного забезпечення та тестування: базовий курс : навч. посіб. / За ред. С. Я. Крепич, І. Я. Співак / для бакалаврів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення». – Тернопіль : ФОП Паляниця В. А., 2020. – 478 с.

2. Дорогий Я. Ю. Сучасні технології автоматизованого проектування і верифікації програм: Конспект лекцій : навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Інженерія програмного забезпечення комп'ютерних систем» / КПІ ім. Ігоря Сікорського ; уклад. : Я. Ю. Дорогий, О. О. Дорога-Іванюк. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 89 с.

Електронне навчальне видання

Методичні рекомендації

до проведення практичних занять та виконання самостійної роботи
з навчальної дисципліни

«ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Частина 1

*(для здобувачів 3 курсу першого (бакалаврського) рівня
вищої освіти денної форми навчання
зі спеціальності 126, F6 – Інформаційні системи і технології)*

Укладач **БУЛАЄНКО** Марина Володимирівна

Відповідальний за випуск *О. Б. Костенко*

Редактор *О. В. Михаленко*

Комп'ютерне верстання *І. В. Волосожарова*

План 2023, поз. 224М

Підп. до друку 28.05.2025. Формат 60 × 84/16.
Ум. друк. арк. 2,8.

Видавець і виготовлювач:

Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Чорноглазівська (Маршала Бажанова), 17, Харків, 61002.

Електронна адреса: office@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК № 5328 від 11.04.2017.