

Міністерство освіти та науки України
Харківський національний університет радіоелектроніки

Факультет _____ Інфокомунікацій _____
(повна назва)

Кафедра _____ Інформаційно-вимірювальних технологій _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

Оцінка якості програмного забезпечення на основі сучасних стандартів

(тема)

Виконав:

Студент 6 курсу, групи ЯССм-21-1

_____ Фоменко В. Д.

(прізвище, ініціали)

Спеціальність 152 Метрологія та інформаційно-вимірювальна техніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма «Якість, стандартизація та сертифікація»

(повна назва освітньої програми)

Керівник доц. Штефан Н. В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____

(підпис)

_____ Захаров І. П.

(прізвище, ініціали)

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інформаційно-вимірювальних технологій
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 152 – Метрологія та інформаційно-вимірювальна техніка
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма «Якість, стандартизація та сертифікація»
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Фоменко Валерії Дмитрівни
(прізвище, ім'я та батькові)

- 1 Тема роботи Оцінка якості програмного забезпечення на основі сучасних стандартів
затверджена наказом по університету від «__» _____ 20__ р. №__
- 2 Термін здачі студентом роботи до екзаменаційної комісії _____ 20__ р.
- 3 Вихідні дані до роботи:
- 4 Перелік питань, що потрібно опрацювати в роботі:
 1. Поняття про якість програмного забезпечення.
 2. Загальні підходи до вимірювання якості програмного забезпечення.
 3. Вимірювання якості програмного продукту.
 4. Оцінка якості систем та програмного забезпечення

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Виявлення літератури з теми дослідження	21.07.2022	
2	Розробка структури дипломної роботи	02.09.2022	
3	Аналіз виявленої літератури	05.09.2022	
4	Підготовка 1 розділу кваліфікаційної роботи	20.09.2022	
5	Підготовка 2 розділу кваліфікаційної роботи	09.10.2022	
6	Підготовка 3 розділу кваліфікаційної роботи	11.11.2022	
7	Підготовка 4 розділу кваліфікаційної роботи	06.12.2022	
8	Підготовка висновків та вступу	12.12.2022	
9	Редагування та оформлення кваліфікаційної роботи	15.12.2022	
10	Представлення кваліфікаційної роботи	22.12.2022	

Дата видачі завдання

Студент _____

(підпис)

Керівник роботи _____

(підпис)

Штефан Н. В. _____

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 65 сторінок, 4 основні частини, 11 рисунків, 13 таблиць, 1 додаток, 7 джерел згідно з переліком посилань.

Об'єкт дослідження: стандарти серії 25000 щодо якості програмного забезпечення.

Мета роботи: проаналізувати рекомендації та вимоги сучасних стандартів щодо вимірювання якості програмного забезпечення та з'ясувати можливість їх практичного застосування.

Методи дослідження: аналіз нормативної документації.

У роботі було розглянуто: серії стандартів ISO 25000 SQuaRE щодо якості програмного забезпечення; моделі якості програмного забезпечення; стандарти SQuaRE щодо вимірювання якості; що таке елемент показника якості програмного забезпечення; загальний підхід до вимірювання якості програмного забезпечення; еталонну модель вимірювання якості за стандартом ISO/IEC 25020; вибір та побудову показників якості; планування та здійснення вимірювань; вимірювання якості програмного продукту.

ВИМІРЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЕЛЕМЕНТ ПОКАЗНИКА ЯКОСТІ, МОДЕЛЬ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ПОКАЗНИК ЯКОСТІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ЗМІСТ

ВСТУП.....	7
1 ПОНЯТТЯ ПРО ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	9
1.1 Професійний підхід до якості програмного забезпечення	9
1.2 Серії стандартів ISO 25000 SQuaRE щодо якості програмного забезпечення	10
1.3 Еволюція моделей якості програмного забезпечення.....	11
1.4 Модель якості програмного забезпечення відповідно до SQuaRE	14
1.5 Відповідність ISO 25000 загальноприйнятій метрологічній термінології....	20
2 ЗАГАЛЬНІ ПІДХОДИ ДО ВИМІРЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	22
2.1 Стандарти SQuaRE щодо вимірювання якості.....	22
2.2 Елемент показника якості програмного забезпечення як основа вимірювання якості.....	22
2.3 Загальний підхід до вимірювання якості програмного забезпечення	26
2.4 Еталонна модель вимірювання якості за стандартом ISO/IEC 25020.....	27
2.4 Вибір та побудова показників якості	29
2.5 Планування та здійснення вимірювання	31
3 ВИМІРЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ	33
3.1 Показники функціональної придатності.....	33
3.2 Показники ефективності роботи.....	33
3.3 Показники сумісності	35
3.4 Показник зручності використання (юзабіліті)	36
3.5 Показники надійності	38
3.6 Показники безпеки.....	40
3.7 Показники ремонтпридатності	42
3.8 Показники переносимості	43
3.9 Вимірювання якості продукту по SQuaRE	45
□ 4 ОЦІНКА ЯКОСТІ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	47
4.1 Нормативна база та загальні поняття.....	47
4.2 Еталонна модель оцінки якості програмного продукту.....	47
4.3 Реалізація процесу оцінки якості програмного продукту.....	51
4.3.2 Документація	52
4.3.3 Вимоги оцінки	53

	6
4.3.4 Підготовка до проведення оцінки якості програмного продукту	54
4.3.5 Розроблення оцінки якості програмного продукту	56
4.3.6 Виконання оцінки якості програмного продукту	57
4.3.7 Завершення оцінки	58
ВИСНОВКИ.....	62
ПЕРЕЛІК ПОСИЛАНЬ	63
ДОДАТОК А	64

ВСТУП

Комп'ютерні системи стали невід'ємною частиною життєзабезпечення сучасної людини. Наслідки від використання неякісного програмного забезпечення можуть бути катастрофічними, починаючи від техногенних аварій, матеріальних втрат і небезпеки для людини і закінчуючи імідж-втратами компанії-розробника.

Початком активної розробки і використання програмних засобів як невід'ємної частини комп'ютерів можна вважати середину минулого століття. Потім в 1968 році на конференції НАТО (NATO Software Engineering Conference), яка проходила в німецькому місті Гарміш, вперше з'явився термін «програмна інженерія» (Software engineering) і була сформульована однойменна концепція. За період розвитку програмної інженерії, як самостійного напрямку в інженерній практиці, а потім і формування її як системної науки, одним з ключових було питання якості програмного забезпечення.

Під якістю програмного забезпечення розуміється здатність програмного продукту задовольняти встановленим або передбачуваним потребам [1]. Тому важливим є забезпечення спільного розуміння між розробниками і користувачами. Інженери повинні чітко розуміти зміст, вкладений в концепцію якості, характеристики і значення якості щодо розроблюваного або супроводжуваного їм програмного забезпечення. Сформулювати чіткі і зрозумілі вимоги до якості програмного забезпечення, а потім кількісно їх оцінити – одне з пріоритетних завдань забезпечення якості програмного забезпечення.

Вимірювання є ключовим фактором забезпечення якості програмного забезпечення. Для вимірювання якості програмного забезпечення необхідно провести кількісну оцінку характеристик програмного забезпечення та оцінити, наскільки програмне забезпечення відповідає встановленим вимогам [1].

Вимірювання є основним інструментом для керуючих дій в життєвому циклі систем та програмних засобів, для оцінки виконання планів та моніторингу. Вимірюванням мають підлягати зміни, що є основою постійного вдосконалення.

Використання стандартів надає багато потенційних переваг для будь-якої організації, особливо в таких ключових питаннях, як вимірювання програмних продуктів та інформаційно-вимірювальних систем.

Сучасні міжнародні стандарти є основою для формування вимог до якості програмного забезпечення і вимірювання якості програмного забезпечення. Це дозволяє знизити ризики при розробці, впровадженні і супроводі програмного забезпечення. Актуальність питання застосування міжнародних стандартів підтверджується ще і тим, що стандарти прийняті в Україні як національні.

1 ПОНЯТТЯ ПРО ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Професійний підхід до якості програмного забезпечення

Визначення поняття «якість», що надає можливість кількісної оцінки та найбільш точно відображає його суть, надано в стандарті ДСТУ ISO 9000 [2]:

якість – ступінь, до якого сукупність власних характеристик об'єкта задовольняє вимогам. Під вимогами розуміють сформульовані потреби чи очікування, загальнозрозумілі чи обов'язкові.

Сучасний підхід до якості програмного забезпечення відображений в серії стандартів ISO/IEC 25000 Software Product Quality Requirements and Evaluation або SQuaRE (Вимоги до якості систем і програмних засобів та її оцінювання).

Якість програмного забезпечення (software quality) – ступінь задоволення програмним продуктом заявленим та передбаченим потребам при використанні в зазначених умовах.

Це визначення має відмінність від визначення якості ISO 9000 головним чином тим, що визначення якості програмного забезпечення відноситься до задоволення заявлених потреб, у той час як визначення якості ISO 9000 відноситься до задоволення вимог.

Для можливості оцінки якості з точки зору користувачів вводиться поняття якості при використанні [3].

Якість при використанні (quality in use) – ступінь, з якою продукція або система можуть бути застосовані певними користувачами для задоволення їх вимог в досягненні цілей ефективності (в т. ч. і економічної), уникаючи ризику, задоволеності і охоплення контексту в заданих умовах використання.

Таким чином, під якісним програмним забезпеченням розуміється програмне забезпечення, яке не містить помилок або дефектів, доставлено вчасно і в рамках встановленого бюджету, відповідає вимогам і/або очікуванням і є ремонтпридатним.

1.2 Серії стандартів ISO 25000 SQuaRE щодо якості програмного забезпечення

Перші міжнародні стандарти в області якості програмного забезпечення були прийняті ще в 1991 році і з тих пір кілька разів переглядалися. На сьогоднішній день діє серія стандартів ISO 25000 SQuaRE (Systems and software Quality Requirements and Evaluation) – логічно організовані та уніфіковані серії стандартів, які забезпечують специфікацію вимог до якості програмного забезпечення, їх використання, дозволяють зменшити невизначеність при сумісній роботі організацій щодо розробки, впровадження та супроводу програмного забезпечення, наприклад, між замовниками розробки, розробниками та незалежними оцінювачами.

Організацію серії стандартів SQuaRE показано на рис. 1.1.

Розділ «Вимоги до якості» 2503n	Розділ «Модель якості» 2501n	Розділ «Оцінка якості» 2504n
	Розділ «Менеджмент якості» 2500n	
	Розділ «Вимірювання якості» 2502n	
Розділ розширення 25050 – 25099		

Рисунок 1.1 – Організація серії стандартів SQuaRE

Серія SQuaRE розділена на розділи (сімейства стандартів), характеристика яких наведена в таблиці 1.1.

Таблиця 1.1 – Характеристика розділів стандартів SQuaRE

Розділ	Призначення та короткий зміст
Розділ «Менеджмент якості» 2500n	визначають загальні моделі, терміни та визначення, що використовуються далі в усіх інших міжнародних стандартах серії SQuaRE. У розділі також представлені вимоги і методичні матеріали, що стосуються функцій підтримки, які відповідають за управління вимогами до програмного продукту, його специфікацією і оцінкою
Розділ «Модель якості» 2501n	представляють деталізовані моделі якості обчислювальних систем і програмного забезпечення, якості при використанні і якості даних. Крім того, представлено практичний

Розділ	Призначення та короткий зміст
Розділ «Вимірювання якості» 2502n	посібник з використання моделей якості включають в себе еталонну модель вимірювання якості програмного продукту, математичні визначення показників якості та практичне керівництво щодо їх використання. У цьому розділі представлені показники внутрішньої якості програмного забезпечення, показники зовнішньої якості програмного забезпечення і показники якості при використанні. Крім того, визначено і представлені елементи показників якості (ЕПЯ), що формують основу для перерахованих вище показників
Розділ «Вимоги до якості» 2503n	визначають вимоги до якості на основі моделей якості і показників якості. Такі вимоги до якості можуть використовуватися в процесі формування вимог до якості програмного продукту перед розробкою або як вхідні дані для процесу оцінки
Розділ «Оцінка якості» 2504n	формулюють вимоги, рекомендації та методичні матеріали для оцінки програмного продукту, що виконується як оцінювачами, так і замовниками або розробниками. Крім того, в них представлена підтримка документування показника вимірювання як модуля оцінки
Розділ розширення 25050 – 25099	включають в себе вимоги до якості готового комерційного (коробкового) програмного забезпечення і загального промислового формату для звітів по зручності використання

1.3 Еволюція моделей якості програмного забезпечення

Визначення якості програмного забезпечення пов'язано з поняттями «модель якості», «характеристики якості», «властивості якості» та «показники якості».

Модель якості – це сукупність класів характеристик. Характеристики можна розділити на підхарактеристики і, в деяких випадках, на підпідхарактеристики. Вимірювані властивості, пов'язані з якістю, називаються властивостями якості. Властивості якості пов'язані з відповідними показниками якості.

Характеристики (підхарактеристики, підпідхарактеристики), що увійшли в дані моделі, як правило, є основою при формуванні вимог до програмного забезпечення конкретних проектів.

За майже півстолітню історію розвитку програмної інженерії було запропоновано десятки різних моделей якості ПЗ, але тільки частина з них стала широко відомою і застосовується при формуванні вимог. У таблиці 1.2 систематизовано інформацію про такі моделі із зазначенням не тільки бібліографічних

даних, а й двох важливих параметрів – кількості рівнів ієрархії і кількості характеристик/підхарактеристик за рівнями/підрівнями.

Таблиця 1.2 – Найбільш відомі моделі якості програмного забезпечення

№ з/п	Назва моделі	Рік публікації	Кількість рівнів моделей	Кількість характеристик/підхарактеристик/підпідхарактеристик	Авторство
1	МакКола	1977	2	11/35	Джон МакКол
2	Боема	1978	3	3/8/18	Боем
3	Гези	1991	1	8	Карло Гези
4	FURPS	1992	2	5/25	Грейди Б. та компанія Hewlett Packard
5	IEEE	1993	2	6/19	IEEE
6	Дромера	1995	2	4/13	Дромер
7	ISO 9126-1	2001	2	6/19	ISO
8	QMOOD	2002	1	6	Дж. Бонсія
9	ISO 25010	2010	2	8/31	ISO

Модель МакКола

Запропонована МакКолом в 1977 році модель якості програмного забезпечення є першою в своїй області і призначена для оцінки якості програмного продукту за допомогою його характеристик.

Модель МакКола складається з трьох основних напрямків визначення якості програмного забезпечення:

- використання (коректність, надійність, ефективність, цілісність, практичність);
- модифікація (тестованість, гнучкість, супроводуваність);
- переносимість (мобільність, можливість багаторазового використання, функціональна сумісність).

Об'єктивно оцінити або виміряти чинники якості досить важко, тому МакКол ввів метрики якості, які з його точки зору легше вимірювати і оцінювати. Оцінки в його шкалою приймають значення від 0 до 10.

Кожна метрика впливає на оцінку декількох чинників якості. Числове вираження фактора є лінійною комбінацією значень, метрик, які впливають на

нього. Коефіцієнти цього виразу визначають по-різному для різних організацій, команд розробки, видів програмного забезпечення, процесів, що використовуються, і т.п.

Модель FURPS/FURPS +

Акронім FURPS, який використовується в позначенні моделі, позначає такі категорії вимог до якості ПЗ:

- Functionality (Функціональність) – особливості, можливості, безпека;
- Usability (Практичність) – людський фактор, ергономічність, призначена для користувача документація ;
- Reliability (Надійність) – частота відмов, відновлення інформації, прогнозованість;
- Performance (Продуктивність) – час відгуку, пропускна здатність, точність, доступність, використання ресурсів;
- Supportability (Експлуатаційна придатність) – тестованість, розширюваність, адаптованість, супроводуваність, сумісність, конфігурованість, обслуговуваність, вимоги до установки, що локалізується.

Символ «+» доповнює FURPS наступними атрибутами:

- обмеження проекту (обмеження по ресурсам, вимоги до мов і засобів розробки, вимоги до апаратного забезпечення);
- інтерфейс (обмеження, що накладаються на взаємодію із зовнішніми системами);
- вимоги до виконання;
- фізичні вимоги;
- вимоги до ліцензування.

Модель якості FURPS, запропонована Р. Грейді і Хьюлеттом Паккард, побудована схожим чином з моделями МакКола і Боема, але на відміну від них складається з двох шарів: перший визначає характеристики, а другий з атрибутами, які з ними пов'язані.

Основною концепцією, що лежить в основі FURPS, є декомпозиція характеристик програмного забезпечення на дві категорії вимог, а саме, функціональні (F) і нефункціональні (URPS) вимоги. Ці виділені категорії можуть бути використані як в якості вимог до програмного продукту, так і в оцінці його якості.

Модель якості ISO 25000

Модель якості програмного забезпечення ISO 25000 є «ідеальною» (еталонною) для всіх розглянутих моделей, тому що, по-перше, це остання запропонована модель, яка повинна враховувати всі особливості забезпечення і оцінювання якості сучасного програмного забезпечення, по-друге, дана модель описана в міжнародному спеціалізованому стандарті.

1.4 Модель якості програмного забезпечення відповідно до SQuaRE

Якість в стандартах SQuaRE описується чотирма моделями: модель якості при використанні і модель якості продукту, визначена ISO/IEC/IEEE 25010, модель якості IT-сервісів ISO/IEC/IEEE 25011, а також модель якості даних, визначена ISO/IEC/IEEE 25012.

Спільне використання моделей якості дає підставу вважати, що всі характеристики якості враховані. Ці моделі забезпечують багато характеристик якості, в яких зацікавлена велика кількість людей, таких як: розробники програмного забезпечення, системні інтегратори, власники, фахівці з обслуговування, підрядники, професіонали забезпечення і управління якістю та користувачі. Не всі характеристики якості з повної множини, що забезпечується цими моделями, значущі для конкретної зацікавленої сторони. Проте кожен категорію зацікавлених сторін слід враховувати під час аналізу та врахування важливості характеристик якості для кожної моделі до завершення формування набору характеристик якості, які будуть використовуватися, щоб встановити, наприклад, вимоги до продуктивності продукції і системи або критерії оцінки.

Метою моделі якості продукту є комп'ютерна система, в яку входить цільовий програмний продукт, а мета моделі якості при використанні – це сукуп-

на людино-машинна система, яка включає в себе і цільову комп'ютерну систему, і цільовий програмний продукт.

У цільову комп'ютерну систему входять також комп'ютерне обладнання, нецільові програмні продукти, нецільові дані і цільові дані, що, в свою чергу, є об'єктом аналізу моделі якості даних. Цільова комп'ютерна система є частиною інформаційної системи, до складу якої можуть бути також включена не одна комп'ютерна система і система зв'язку, такі як локальна мережа і Інтернет. До складу інформаційної системи в більшій людино-машинній системі (такої як корпоративна система, вбудована система або великомасштабна система управління) можуть входити користувачі, технічне і фізичне середовище використання.

Рамки цільової системи визначаються виходячи з області застосування вимог або оцінки і з того, хто розглядається в якості користувачів.

Модель якості при використанні визначає п'ять характеристик, пов'язаних з результатами взаємодії з системою: результативність, продуктивність, задоволеність, свободу від ризику і покриття контексту (рис.1.2). Кожна характеристика застосовна для різних видів діяльності зацікавлених осіб, наприклад, для взаємодії оператора або підтримки розробника.

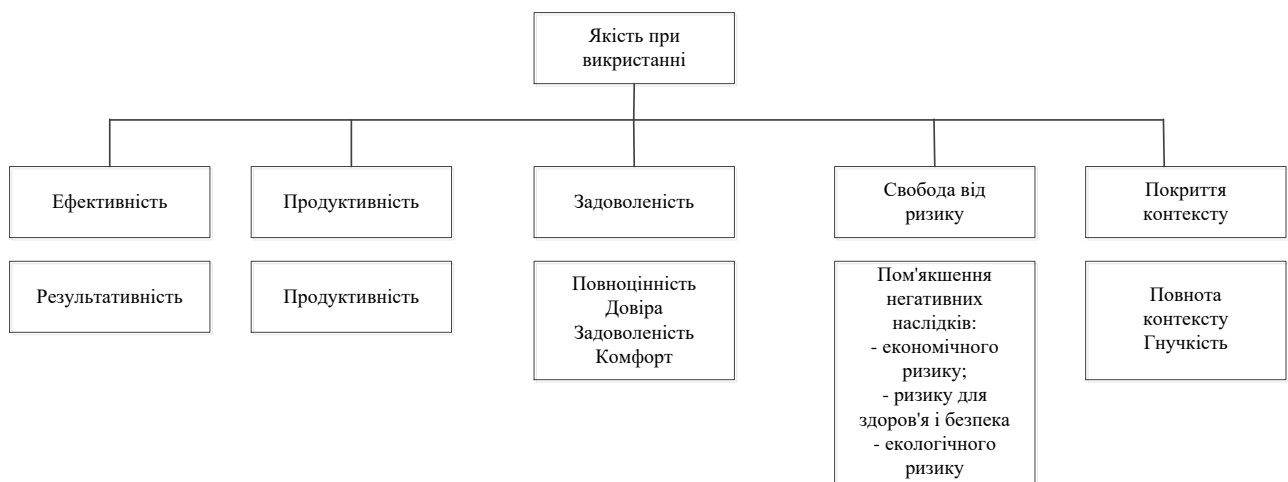


Рисунок 1.2 – Модель якості при використанні

Детальний опис характеристик на підхарактеристик моделі якості при використанні наведено в табл. 1.3.

Таблиця 1.3 – Модель якості при використанні за ISO/IEC 25010

Характеристика	Підхарактеристики
1. Ефективність (effectiveness) – точність і повнота, з якою користувачі досягають певних цілей	1.1 Результативність (effectiveness) – точність і повнота, з якою користувачі досягають певних цілей.
2. Продуктивність (efficiency) – зв'язок точності і повноти досягнення користувачами цілей з витраченими ресурсами	2.1 Продуктивність (efficiency) – зв'язок точності і повноти досягнення користувачами цілей з витраченими ресурсами
3. Задоволеність (satisfaction) – здатність продукту або системи задовольнити вимогам користувача в заданому контексті використання	3.1 Повноцінність (usefulness) – ступінь задоволеності користувача досягненням прагматичних цілей, включаючи результати використання і наслідки використання.
	3.2 Довіра (trust) – ступінь впевненості користувача або іншої зацікавленої особи в тому, що продукт або система будуть виконувати свої функції так, як це передбачалося.
	3.3 Задоволеність (pleasure) – ступінь задоволення користувача від задоволення персональних вимог.
	3.4 Комфорт (comfort) – ступінь задоволеності користувача фізичним комфортом.
4. Свобода від ризику (freedom from risk) – здатність продукту або системи пом'якшувати потенційний ризик для економічного становища, життя, здоров'я або навколишнього середовища.	4.1 Пом'якшення негативних наслідків економічного ризику (economic risk mitigation) – здатність продукту або системи пом'якшувати потенційний ризик для фінансового становища і ефективної роботи, комерційної нерухомості, репутації або інших ресурсів в передбачуваних умовах використання.
	4.2 Пом'якшення негативних наслідків ризику для здоров'я і безпеки (health and safety risk mitigation) – здатність продукту або системи пом'якшувати потенційний ризик для людей в передбачуваних умовах використання.
	4.3 Пом'якшення негативних наслідків екологічного ризику (environmental risk mitigation) – здатність продукту або системи пом'якшувати потенційний ризик для майна або навколишнього середовища в передбачуваних умовах використання.
5. Покриття контексту (context coverage) – ступінь, в якій продукт або система можуть бути використані з ефективністю, результативністю, свободою від ризику і відповідно до вимог як у початку певних умов використання, так і в умовах, що виходять за специфікацію.	5.1 Повнота контексту (context completeness) – ступінь, в якій продукт або система можуть бути використані з ефективністю, результативністю, свободою від ризику і відповідно до вимог при всіх зазначених умовах використання.
	5.2 Гнучкість (flexibility) – ступінь, в якій продукт або система можуть бути використані з ефективністю, результативністю, свободою від ризику і відповідно до вимог в умовах, що виходять за рамки спочатку визначених у вимогах.

Якість при використанні системи характеризує вплив продукції (система або програмний продукт) на зацікавлену сторону. Вона визначається якостями

програмного забезпечення, апаратних засобів, операційного середовища, а також характеристиками користувачів, завдань і соціального середовища. Усі ці фактори роблять свій внесок в якість системи при використанні.

Модель якості продукту зводить властивості якості системи/програмного продукту до восьми характеристик, якими є: функціональна придатність, рівень продуктивності, сумісність, зручність користування, надійність, захищеність, супроводуваність і переносимість (мобільність). Кожна характеристика, в свою чергу, складається з ряду відповідних підхарактеристик (рис. 1.3).



Рисунок 1.3 – Модель якості продукту

Детальний опис характеристик на підхарактеристик моделей якості продукту наведено в табл. 1.4.

Таблиця 1.4 – Модель якості програмного забезпечення за ISO/IEC 25010

Характеристика	Підхарактеристики
1. Функціональна придатність (functional suitability) – ступінь, в якій продукт або система забезпечують виконання функції відповідно до заявлених і якими мається на увазі потребами при використанні в зазначених умовах.	1.1 Функціональна повнота (functional completeness) - ступінь покриття сукупністю функцій всіх визначених завдань і цілей користувача
	1.2 Функціональна коректність (functional correctness) - ступінь забезпечення продуктом або системою необхідного ступеня точності коректних результатів.
	1.3 Функціональна доцільність (functional appropriateness) - ступінь функціонального спрощення виконання певних завдань і досягнення цілей.
2. Рівень продуктивності (performance efficiency) – продуктивність щодо суми використаних при певних	2.1 Часові характеристики (time behaviour) – ступінь відповідності вимогам за часом відгуку, часу обробки і показників пропускну здатності продукту або системи
	2.2 Використання ресурсів (resource utilization) – ступінь

Характеристика	Підхарактеристики
умовах ресурсів.	задоволення вимог по споживанню обсягів і видів ресурсів продуктом або системою при виконанні ними своїх функцій
3. Сумісність (compatibility) – здатність продукту, системи або компонента обмінюватися інформацією з іншими продуктами, системами або компонентами, і/або виконувати необхідні функції при спільному використанні одних і тих же апаратних засобів або програмного середовища.	<p>2.3 Потенційні можливості (capacity) – ступінь відповідності вимогами граничних значень параметрів продукту або системи.</p> <p>3.1 Співіснування (сумісність) (co-existence) – здатність продукту спільно функціонувати з іншими незалежними продуктами в загальному середовищі з поділом загальних ресурсів і без негативного впливу на будь-який інший продукт</p> <p>3.2 Функціональна сумісність (інтероперабельність) (interoperability) - здатність двох або більше систем, продуктів або компонент обмінюватися інформацією і використовувати таку інформацію.</p>
4. Зручність використання (usability) – ступінь, в якій продукт або система можуть бути використані певними користувачами для досягнення конкретних цілей з ефективністю, результативністю і задоволеністю в заданому контексті використання.	<p>4.1 Визначність придатності (appropriateness recognizability) - можливість користувачів зрозуміти, чи підходить продукт або система для їх потреб, порівняємо чи з функціональною доцільністю (functional appropriateness)</p> <p>4.2 Вивчаємість (learnability) - можливість використання продукту або системи певними користувачами для досягнення конкретних цілей навчання для експлуатації продукту або системи з ефективністю, результативністю, свободою від ризику і відповідно до вимог в зазначеному контексті використання</p> <p>4.3 Керованість (operability) - наявність в продукті або системі атрибутів, що забезпечують просте управління і контроль</p> <p>4.4 Захищеність від помилки користувача (user error protection) - рівень системного захисту користувачів від помилок</p> <p>4.5 Естетика користувацького інтерфейсу (user interface aesthetics) - ступінь «приємності» і «задоволеності» користувача інтерфейсом взаємодії з користувачем.</p> <p>4.6 Доступність (accessibility) - можливість використання продукту або системи для досягнення певної мети в зазначеному контексті використання широким колом людей з різними можливостями.</p>
5. Надійність (reliability) - ступінь виконання системою, продуктом або компонентом певних функцій при зазначених умовах протягом встановленого періоду часу	<p>5.1 Завершеність (maturity) - ступінь відповідності системи, продукту або компонента при нормальній роботі вимогам надійності</p> <p>5.2 Готовність (availability) - ступінь працездатності та доступності системи, продукту або компонента. Готовність визначається поєднанням завершеності, яка визначає частоту відмов, відмовостійкості та відновлюваності, яка, в свою чергу, визначає тривалість часу бездіяльності після кожного відмови.</p> <p>5.3 Відмовостійкість (fault tolerance) - здатність системи, продукту або компонента працювати як призначене, незва-</p>

Характеристика	Підхарактеристики
	<p>жаючи на наявність дефектів програмного забезпечення або апаратних засобів.</p> <p>5.4 Відновлюваність (recoverability) - здатність продукту або системи відновити дані і необхідний стан системи в разі переривання або збою.</p>
<p>6. Захист, захищеність (security) - ступінь захищеності інформації та даних, що забезпечується продуктом або системою шляхом обмеження доступу людей, інших продуктів або систем до даних відповідно до типів і рівнями авторизації</p>	<p>6.1 Конфіденційність (confidentiality) - забезпечення продуктом або системою обмеження доступу до даних тільки для тих, кому доступ дозволений</p> <p>6.2 Цілісність (integrity) - ступінь запобігання системою, продуктом або компонентом несанкціонованого доступу або модифікації комп'ютерних програм або даних</p> <p>6.3 Непідробленість (non-repudiation) - ступінь, з якою може бути доведений факт події або дії таким чином, що цей факт не може бути відкинутий коли-небудь пізніше.</p> <p>6.4 Відстеження (accountability) - ступінь, до якої дії об'єкта можна простежити однозначно</p> <p>6.5 Справжність (authenticity) - ступінь достовірності totoжності об'єкта або ресурсу необхідного об'єкту або ресурсу.</p>
<p>7. Супроводуваність, можливість модифікацій (maintainability) - результативність і ефективність, з якими продукт або система можуть бути модифіковані передбачуваними фахівцями з обслуговування</p>	<p>7.1 Модульність (modularity) - ступінь представлення системи або комп'ютерної програми у вигляді окремих блоків таким чином, щоб зміна одного компонента надавала мінімальний вплив на інші компоненти</p> <p>7.2 Можливість багаторазового використання (reusability) - ступінь, в якій актив може бути використаний в декількох системах або в створенні інших активів</p> <p>7.3 Аналізованість (analysability) - ступінь простоти оцінки впливу змін однієї чи більше частин на продукт або систему або простоти діагностики продукту для виявлення недоліків і причин відмов, або простоти ідентифікації частин, що підлягають зміні</p> <p>7.4 Модифікованість (modifiability) - ступінь простоти ефективного і раціонального зміни продукту або системи без додавання дефектів і зниження якості продукту</p> <p>7.5 Тестованість (testability) - ступінь простоти ефективного і раціонального визначення для системи, продукту або компонента критеріїв тестування, а також простоти виконання тестування з метою визначення відповідності цим критеріям.</p>
<p>8. Переносимість, мобільність (portability) - ступінь простоти ефективного і раціонального перенесення системи, продукту або компонента з одного середовища (апаратних засобів, програмного забезпечення, операційних умов або умов використання) в іншу</p>	<p>8.1 Адаптованість (adaptability) - ступінь простоти ефективною і раціональною адаптації для відрізняються або вдосконалених апаратних засобів, програмного забезпечення, інших операційних середовищ або умов використання</p>
	<p>8.2 Встановлюваність (installability) - ступінь простоти ефективною і раціональною, успішною установки і / або видалення</p>

Характеристика	Підхарактеристики
	ня продукту або системи в заданому середовищі 8.3 Взаємозамінність (replaceability) - здатність продукту замінити інший конкретний програмний продукт для досягнення тих же цілей в тих же умовах.

Моделі якості продукції і якості при використанні можуть бути використані для визначення вимог, вироблення показників і виконання оцінки якості. Певні характеристики якості можуть бути використані в якості контрольного списку для забезпечення детального дослідження вимог до якості, забезпечуючи таким чином основу для оцінки необхідних в процесі розробки систем наступних трудовитрат і дій. Характеристики в моделі якості при використанні і моделі якості продукту призначені для використання в якості набору при специфікації або оцінці якості програмного продукту або комп'ютерної системи.

Майже неможливо визначити або виміряти всі підхарактеристики для всіх частин більшої комп'ютерної системи або програмного продукту. Аналогічно в більшості випадків практично не застосовується визначення або вимірювання якості при використанні для всіх можливих сценаріїв завдань користувача. Відносна важливість характеристик якості залежить від цілей високого рівня і цілей проекту. У зв'язку з цим перед використанням для виділення з вимог тих характеристик і підхарактеристик, які найбільш важливі, модель повинна бути відповідним чином адаптована, а ресурси розподілені між різними типами показників в залежності від цілей зацікавлених осіб і цілей продукту.

1.5 Відповідність ISO 25000 загальноприйнятій метрологічній термінології

Стандарти ISO 25000 SQuaRE гармонізовані з ISO/IEC/IEEE 15939 Інженерія систем і програмних засобів. Процес вимірювання [4] за змістом, який визначає загальний процес і основу для вимірювання систем і програмного забезпечення, а також відповідну термінологію з інженерної точки зору. Слідуючи сучасним тенденціям ISO по гармонізації термінології, стандарт ISO/IEC/IEEE 15939 приймає і адаптує метрологічну термінологію, встановлену

VIM [4] для стандартів, пов'язаних з програмною і системною інженерією. Наступні поняття з ISO/IEC/IEEE 15939 повністю відповідають, адаптовані або засновані на визначеннях з VIM:

– основний показник (base measure) – показник, визначений у термінах атрибута і методу для його кількісного визначення. Базується на визначенні «основної величини»;

– похідний показник (derived measure) – показник, який визначається як функція двох або більше значень основних показників. Адаптований з визначення «похідна величина»;

– вимірювання (measurement) – набір операцій, призначених для визначення значення показника. Адаптований;

– метод вимірювання (measurement method) – логічна послідовність операцій, описана в загальних рисах і використовується в кількісній оцінці атрибута щодо заданої шкали. Базується на визначенні «методу вимірювань». Тип методу вимірювання залежить від природи операцій, використовуваних для кількісної оцінки атрибута. Розрізняють 2 типу: суб'єктивний – кількісна оцінка на основі людського судження; об'єктивний – кількісна оцінка на основі чисельних правил;

– процедура виміру (measurement procedure) – набір операцій з конкретним описом, будуть застосовані до запланованого певного вимірювання згідно з даним методом. Повністю відповідає;

– шкала (scale) – впорядкований набір безперервних або дискретних значень, або набір категорій, яким зіставляється атрибут. Базується на визначенні терміна «шкала (вимірювального інструмента)»;

– одиниця виміру (unit of measurement) – спеціальна величина, визначена і прийнята на основі угоди, з якою порівнюються інші величини того ж виду в порядку вираження їх значимості щодо цієї величини. Повністю відповідає.

2 ЗАГАЛЬНІ ПІДХОДИ ДО ВИМІРЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Стандарти SQuaRE щодо вимірювання якості

Розділ вимірювання якості (2502n) включає п'ять стандартів [5]:

- ДСТУ ISO/IEC 25020 – Структура вимірювання якості: забезпечує основу для проведення вимірювання якості;
- ДСТУ ISO/IEC 25021 – Елементи показника якості: надає формат для визначення ЕПЯ (елементів показника якості) і досить велика кількість прикладів ЕПЯ, які можна використовувати для побудови показників якості програмного забезпечення;
- ДСТУ ISO/IEC 25022 – Вимірювання якості при використанні: надає показники, включаючи пов'язані функції вимірювання характеристик якості в моделі якості при використанні;
- ДСТУ ISO/IEC 25023 – Вимірювання якості систем і програмних продуктів: надає показники, включаючи пов'язані функції вимірювання і ЕПЯ для характеристик якості в моделі якості продукту;
- ISO/IEC 25024 – Вимірювання якості даних: надає показники, включаючи пов'язані функції вимірювання і ЕПЯ для характеристик якості в моделі якості даних;
- ISO/IEC TS 25025 – Вимірювання якості ІТ-сервісів: надає показники для моделі якості ІТ-сервісів.

На рисунку 2.1 показано структуру розділу вимірювання якості.

2.2 Елемент показника якості програмного забезпечення як основа вимірювання якості

Елемент показника якості (Quality measure element (QME)) – це показник, визначений у термінах властивості і методу вимірювання для кількісного ви-

значення цієї властивості, включаючи вибірково перетворення за допомогою математичної функції.

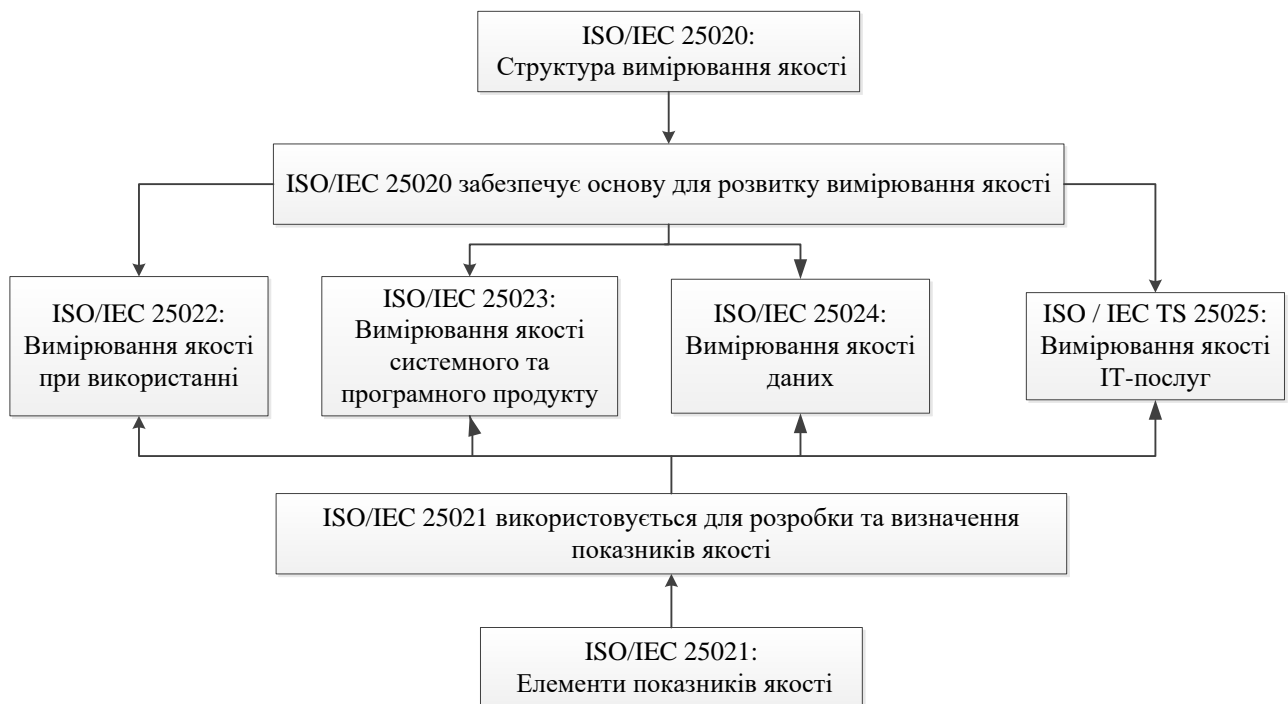


Рисунок 2.1 – Структура розділу вимірювання якості

Основні цілі визначення і використання елементів показника якості (ЕПЯ):

- забезпечувати настанову для організацій, які розробляють і реалізують свої власні ЕПЯ;
- просувати послідовне застосування заданого ЕПЯ для вимірювання і використання властивостей продукції, які відносяться до різних характеристик і підхарактеристик якості продукції;
- допомагати ідентифікувати безліч ЕПЯ, які унікально затребувані, для отримання всіх показників якості даної множини характеристик або підхарактеристик продукції.

ПЯ і відповідно ЕПЯ визначають для розуміння і вказівки характеристик і підхарактеристик якості.

Функція вимірювання застосована до ЕПЯ для генерації ПЯ. Метод вимірювання повинен бути застосований до властивості для встановлення та ідентифікації способу кількісного визначення ЕПЯ.



Рисунок 2.2 – Взаємозв'язок між властивістю кількісного визначення, методу вимірювання і ЕПЯ

Користувач методу вимірювання має ідентифікувати і збирати дані, пов'язані з кількісним визначенням властивості (рис.2.2). Залежно від контексту використання і цілей ЕПЯ ряд властивостей і підвластивостей може бути ідентифікований. Вони і є вхідними даними для методу вимірювань. Ці властивості визначаються і витягуються з артефактів, компонентів, вмісту або поведінки цільового об'єкта (наприклад, документації, коду).

На рис. 2.3 показано, що:

- якість продукції виражена як безліч характеристик якості, які складаються з підхарактеристик;
- показники якості продукції використовують для вказівки інтересу за допомогою характеристик і підхарактеристик якості;
- існує взаємозв'язок властивості кількісного визначення, методу вимірювання і ЕПЯ.

На рисунку 2.4 наведено приклад взаємозв'язку властивості кількісного визначення, методу вимірювання, ЕПЯ і ПЯ для характеристик надійності та зрілості.

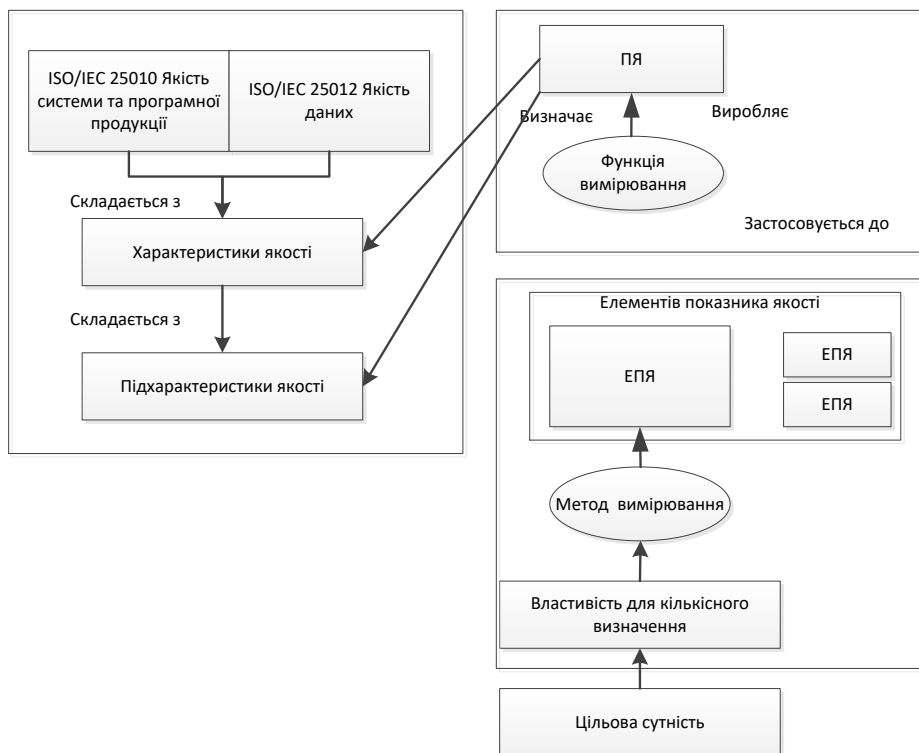


Рисунок 2.3 – Взаємозв'язок властивості кількісного визначення, методу вимірювання, ЕПЯ і ПЯ

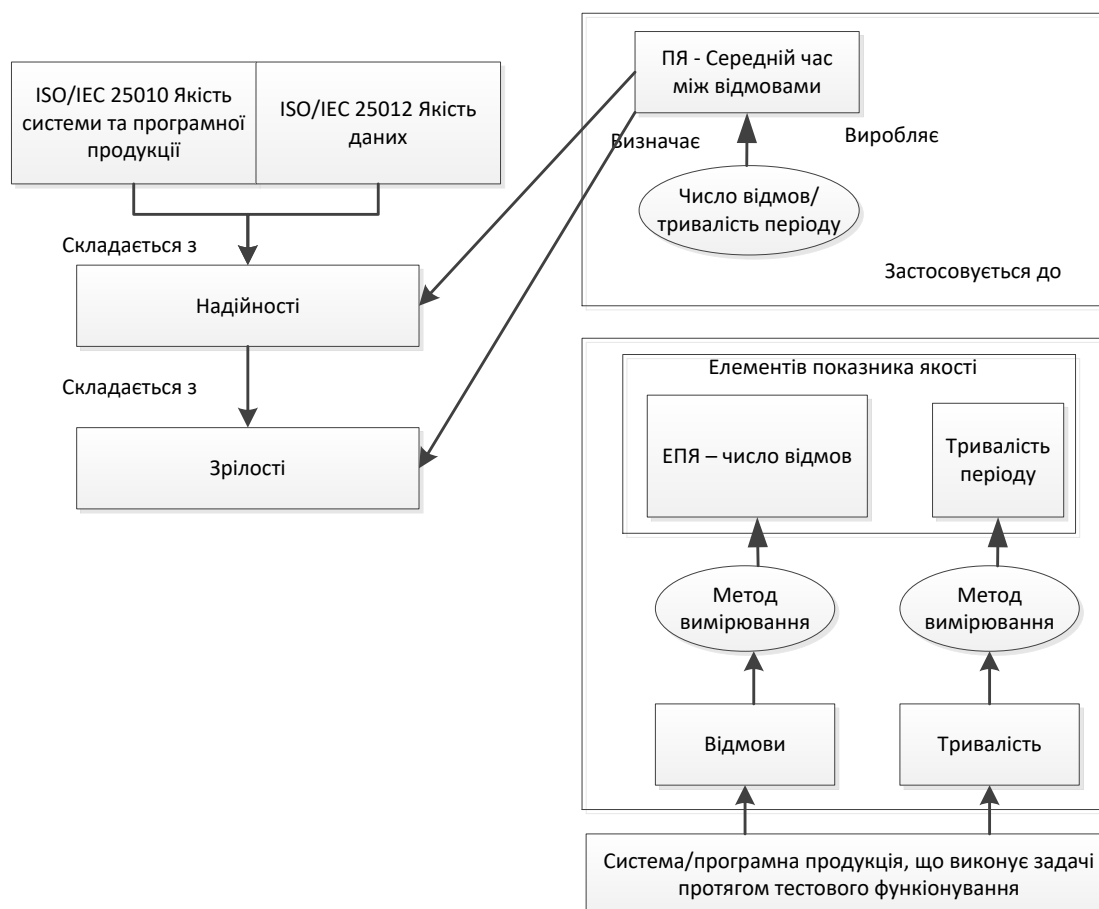


Рисунок 2.4 – Приклад взаємозв'язку властивості кількісного визначення, методу вимірювання, ЕПЯ і ПЯ

2.3 Загальний підхід до вимірювання якості програмного забезпечення

Потреби користувачів у якості включають вимоги щодо якості системи, що використовуються у конкретних контекстах використання. Ці ідентифіковані потреби можуть бути враховані при визначенні зовнішніх та внутрішніх показників якості з використанням характеристик та підхарактеристик програмного продукту.

Якість програмного продукту може бути оцінена шляхом вимірювання внутрішніх властивостей (як правило, статичних показників проміжних продуктів), або шляхом вимірювання зовнішніх властивостей (зазвичай, шляхом вимірювання поведінки коду при виконанні), або шляхом вимірювання якості використовуваних властивостей (коли продукт знаходиться в реальному або модельованому використанні). Відповідні внутрішні властивості програмного забезпечення є необхідною умовою для того, щоб досягти необхідної зовнішньої поведінки, а відповідна зовнішня поведінка є необхідною умовою для досягнення якості використання. (рис. 2.5)

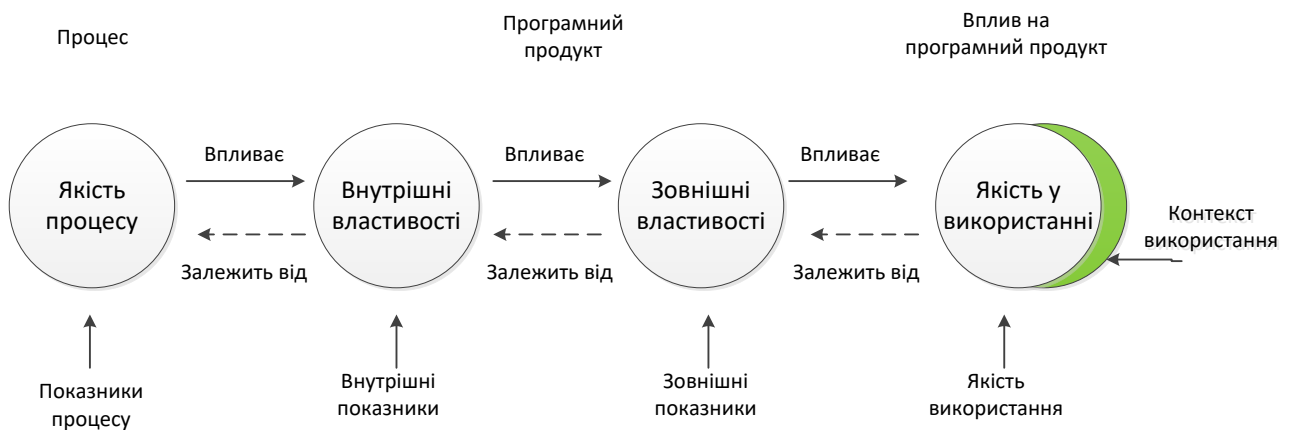


Рисунок 2.5 – Взаємозв'язок між видами показників якості

Внутрішні показники можуть бути застосовані до невиконавчої системи/програмного продукту на етапах його розробки (наприклад, запит на пропозицію, визначення вимог, специфікація проекту або вихідний код), які можуть бути перевірені шляхом огляду, перевірки, моделювання та/або автоматизова-

них інструментів. Внутрішні показники надають користувачам можливість виміряти якість проміжних продуктів і тим самим передбачити якість кінцевого продукту. Це дає можливість користувачеві виявляти проблеми якості та розпочинати коригувальні дії якомога раніше у життєвому циклі розробки. Наприклад, міри складності та кількості, серйозності та частота несправностей, виявлені під час обходу, є внутрішніми показниками якості програмного забезпечення, зробленими на самому продукті.

Зовнішні показники можуть застосовуватися для вимірювання якості системи/програмного продукту шляхом вимірювання поведінки системи, частиною якої вона є. Зовнішні показники можна застосовувати лише на етапах випробувань процесу життєвого циклу та на будь-яких етапах експлуатації. Вимірювання проводиться під час запуску системного/програмного продукту в системному середовищі, в якому він тестується та/або призначений для роботи. Наприклад, кількість несправностей, виявлених під час тестування, є зовнішнім показником якості програмного забезпечення, пов'язаного з кількістю несправностей, наявних у комп'ютерній системі.

2.4 Еталонна модель вимірювання якості за стандартом ISO/IEC 25020

Еталонна модель вимірювання якості описує взаємозв'язок між моделлю якості і побудовою показників якості на основі елементів показників якості, як показано на рис.2.6. Відношення складає еталонну модель для вимірювання якості системи та програмного продукту, якість даних та якість ІТ-послуг.

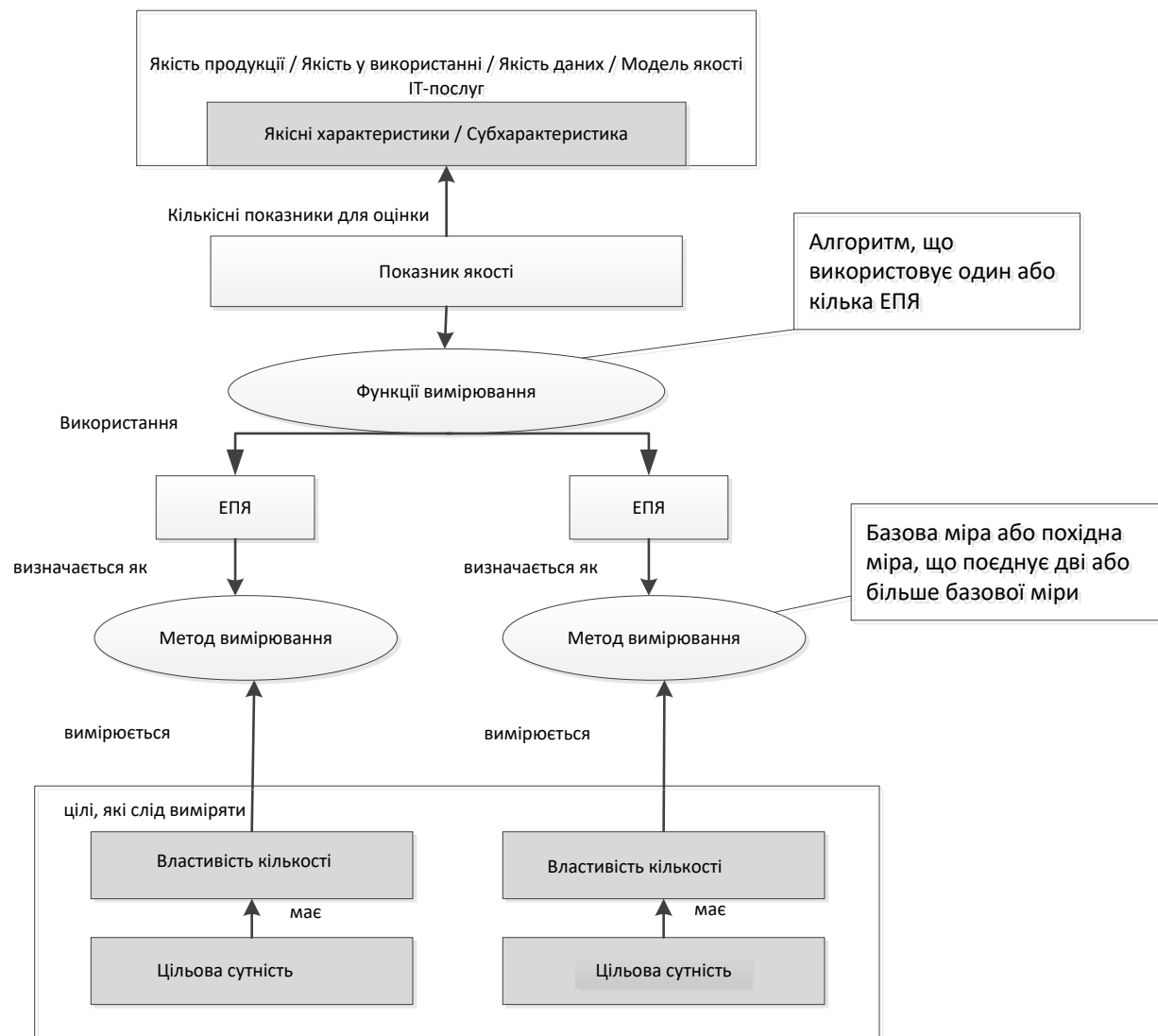


Рисунок 2.6 – Еталонна модель вимірювання якості (QM-RM)

Якість системи, програмного продукту, даних чи ІТ-послуг – це ступінь задоволення заявлених потреб різних зацікавлених сторін, що забезпечує цінність. Потреби користувачів у якості включають вимоги до якості програмного продукту в конкретних контекстах використання. Ці заявлені потреби представлені у серії стандартів SQuaRE моделями якості, які класифікують якість на характеристики, які далі поділяються на підхарактеристики. Якісні властивості вимірюються за допомогою методу вимірювання. Результат застосування методу вимірювання називається QME (ЕПЯ).

ПЯ (показники якості) будуються за допомогою застосування функції вимірювання до набору ЕПЯ. Результат застосування функції вимірювання називається QM (ПЯ). Таким чином, показники якості служать кількісними харак-

теристиками характеристик і підхарактеристик якості. Для вимірювання характеристик якості та субхарактеристик може використовуватися більше одного показника якості.

2.4 Вибір та побудова показників якості

ПЯ призначені для задоволення потреб розробників, менеджерів, прямих та непрямих користувачів та інших зацікавлених сторін в інформації. ПЯ, які потенційно задовольняють вимогам до якості, слід визначати зі стандартів у розділі вимірювання якості серії SQuaRE, таких як ISO/IEC 25022, ISO/IEC 25023 або ISO/IEC 25024. А потім, набір показників якості може бути додатково вдосконалений відповідно до ситуації.

Багато різних комбінацій базових та похідних показників можуть бути обрані для побудови додаткових засобів управління якістю, які виступають як показники або відповідають конкретним вимогам до якості.

При виборі серед альтернатив слід враховувати наступні фактори:

- відповідність пріоритетним інформаційним потребам;
- доцільність збору даних в організаційній одиниці;
- наявність людських ресурсів для збору та управління даними;
- простота збору даних.

Коли ПЯ використовуються для визначення вимог до якості, критичність таких вимог або ризик їх недостатності може розглядатися як один із критеріїв вибору ПЯ. Коли ПЯ використовується для оцінки якості, застосована строгість та терміни можуть розглядатися як критерії.

Під час визначення ПЯ, важливо враховувати, що якість системи – це ступінь задоволення заявлених потреб різних зацікавлених сторін системою, що забезпечує цінність. Ці заявлені потреби представлені у серії міжнародних стандартів SQuaRE моделями якості, які класифікують якість на характеристики, які, у деяких випадках, додатково поділяються на підхарактеристики. Повний набір якісних характеристик цих моделей не буде актуальним для кожної зацікавленої сторони. І все ж кожна категорія зацікавлених сторін повинна бути

представлена при перегляді та розгляді відповідності характеристик якості в кожній моделі перед завершенням набору характеристик якості, які будуть використані, наприклад, для встановлення вимог до продуктивності програмного забезпечення та системи або критеріїв оцінки.

Важливим є документування визначення кожного ПЯ, коли користувач виконує вимірювання системи, програмного продукту, даних та ІТ-послуги. Користувач повинен задокументувати додаткову детальну інформацію, описуючи ПЯ, для більш оперативної роботи:

а) Ідентифікатор: ідентифікаційний код ПЯ. Кожен посвідчення складається з наступних трьох частин:

- скорочений алфавіт, що представляє характеристики якості та, можливо, підхарактеристики.
- порядковий номер послідовного замовлення в межах підхарактеристики якості.
- тег використання:
- G: загальнодоступний, може використовуватися в широкому діапазоні ситуацій;
- S: спеціалізований для конкретних потреб.

б) Назва: Назва ПЯ.

в) Опис: Інформація, включаючи необхідну інформацію (мета заходу) та характеристику/підхарактеристику якості, що надається ПЯ, та (коли корисна) мету заходу.

г) Функція вимірювання: Формула, що показує, як ЕПЯ поєднуються для отримання ПЯ.

д) Метод вимірювання: Тип методу, який можна використовувати для отримання міри.

ЕПЯ використовуються впродовж усього життєвого циклу для вимірювання атрибутів самої системи та програмного продукту, наслідків використання системи та програмного продукту в конкретному контексті та споживаних ресурсів або заходів, що виконуються під час розробки, тестування та обслуговування системного та програмного продукту.

2.5 Планування та здійснення вимірювання

Користувачу розділу стандартів ISO/IEC 2502n необхідно планувати та проводити вимірювання для визначення значення ЕПЯ та ПЯ, які відповідають еталонній моделі на рис. 2.6.

Вимірювання якості планується з урахуванням таких ресурсів, як персонал, автоматизація вимірювань, програмне та апаратне середовище. План вимірювань має не містити дубльованих завдань для здійснення однакових заходів для задоволення різних інформаційних потреб.

Критерії вибору ПЯ та ЕПЯ повинні бути враховані в плані вимірювань, щоб зменшити ризик помилок та зменшити заплановані зусилля, враховуючи принаймні наступне:

- а) бюджет вимірювання;
- б) пріоритетність та суворість ПЯ та ЕПЯ, які відображають критичні вимоги до якості;
- в) графік та задіяні ресурси;
- г) застосування результату вимірювання;
- г) доречність та важливість ПЯ на основі вимог до якості та контексту використання.

Основними видами діяльності, пов'язаними з виконанням вимірювань, є:

- а) визначити модель якості відповідно до різних інформаційних потреб, які стосуються характеристик якості систем або програмного продукту, ІТ-послуг, даних або якості, що використовується;
- б) визначити кандидата та відібрати ПЯ з ЕПЯ для побудови цих ПЯ для визначеної моделі якості;
- в) повідомляти вимірювача або постачальника даних, щоб адекватно приймати ПЯ та ЕПЯ, планувати та співпрацювати з відповідними зацікавленими сторонами для захоплення ПЯ та ЕПЯ;
- г) генерувати значення ЕПЯ за допомогою методу вимірювання;
- г) обчислити значення ПЯ за допомогою функції вимірювання;

д) перевіряти та зберігати значення ПЯ та їх ЕПЯ з контекстною інформацією вимірювання;

е) вимірювати якісні характеристики та/або підхарактеристики за допомогою засобів контролю якості;

є) реєструвати результати та інформувати користувачів, яким потрібна інформація щодо якості для прийняття рішень під час проекту чи експлуатації.

3 ВИМІРЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

Для кожної характеристики якості відповідно до моделі якості продукту (рис.1.3) стандартом ISO/IEC 25023 запроваджуються відповідні показники якості.

3.1 Показники функціональної придатності

Показники функціональної придатності використовуються для оцінки ступеня, в якому продукт чи система надає функції, які відповідають заявленим потребам при використанні у визначених умовах.

До показників функціональної придатності відносяться показники, що наведені в табл. 3.1.

Таблиця 3.1 – Показники функціональної придатності

Назва	Опис	Функція вимірювання
Функціональне покриття	Яка частка зазначених функцій була реалізована?	$X = 1 - A/V$ A = Кількість відсутніх функцій V = Кількість зазначених функцій
Функціональна коректність	Яка частка функцій забезпечує правильні результати?	$X = 1 - A/V$ A = Кількість неправильних функцій V = Кількість розглянутих функцій
Функціональна доцільність цілі використання	Яка частка функцій, необхідних користувачеві, забезпечує належний результат для досягнення конкретної цілі використання?	$X = 1 - A/V$ A = Кількість відсутніх функцій або неправильний серед тих, які потрібні для досягнення конкретної цілі використання V = Кількість функцій, необхідних для досягнення конкретної цілі використання
Функціональна відповідність системи	Яка частка функцій, необхідних користувачам для досягнення своїх цілей, забезпечує належний результат?	$X = \sum_{i=1}^{n} A_i / n$ A _i = оцінка відповідності для цілі використання i, тобто вимірне значення функціональної доцільності цілі використання для i-ї конкретної мети використання n = кількість цілей використання

3.2 Показники ефективності роботи

Показники ефективності роботи використовуються для оцінки результативності щодо обсягу ресурсів, що використовуються за заявлених умов. Ресур-

си можуть включати інші програмні продукти, програмне та апаратне забезпечення конфігурації системи та матеріали (наприклад, папір для друку, носії інформації).

До показників ефективності роботи відносяться показники, що наведені в табл. 3.2.

Таблиця 3.2 – Показники ефективності роботи

Назва	Опис	Функція вимірювання
Середній час реакції	Скільки в середньому часу потрібно системі, щоб відповісти на призначені для користувача завдання або системну задачу?	$X = \sum_{i=1}^{n} (A_i) / n$ $A_i = \text{час, витрачений системою на відповідь на конкретне завдання користувача або системне завдання при } i\text{-му вимірюванні}$ $n = \text{кількість виміряних відповідей}$
Адекватність часу відгуку	Наскільки добре час відповіді системи відповідає зазначеному цільовому показнику?	$X = A/V$ $A = \text{середній час відгуку, виміряний показником середнього часу реакції}$ $V = \text{вказаний цільовий час відгуку}$
Середній час виконання	Який середній час потрібно для завершення роботи або асинхронного процесу?	$X = \sum_{i=1}^{n} (B_i - A_i) / n$ $A_i = \text{Час початку роботи } i$ $B_i = \text{час виконання роботи } i$ $n = \text{кількість вимірювань}$
Адекватність термінів виконання робіт	Наскільки добре час виконання робіт відповідає зазначеним цілям?	$X = A/V$ $A = \text{середній час обороту, виміряний за допомогою показника середнього часу виконання}$ $V = \text{вказаний цільовий час виконання}$
Середня пропускна здатність	Яка середня кількість виконаних робіт за одиницю часу?	$X = \sum_{i=1}^{n} (A_i / B_i) / n$ $A_i = \text{кількість завдань, виконаних протягом } i\text{-го часу спостереження}$ $B_i = i\text{-й період часу спостереження}$ $n = \text{кількість спостережень}$
Середнє використання процесора	Скільки часу процесора використовується для виконання заданого набору завдань порівняно з часом роботи?	$X = \sum_{i=1}^{n} (A_i / B_i) / n$ $A_i = \text{час процесора, фактично використаний для виконання заданого набору завдань під час спостереження } i$ $B_i = \text{час роботи для виконання завдань під час спостереження } i$ $n = \text{кількість спостережень}$
Середнє використання пам'яті	Скільки пам'яті використовується для виконання заданого набору завдань порівняно з доступною пам'яттю?	$X = \sum_{i=1}^{n} (A_i / B_i) / n$ $A_i = \text{розмір пам'яті, фактично використаний для виконання заданого набору завдань для обробки } i\text{-ї вибірки}$ $B_i = \text{обсяг пам'яті, доступний для виконання завдань під час } i\text{-ї обробки вибірки}$ $n = \text{кількість оброблених зразків}$

Назва	Опис	Функція вимірювання
Середнє використання пристроїв вводу/виводу	Скільки часу зайнятості пристрою вводу-виводу використовується для виконання заданого набору завдань порівняно з часом роботи вводу/виводу?	$X = \sum_{i=1}^{n} (A_i / B_i) / n$ A_i = тривалість часу зайнятості пристроїв вводу/виводу для виконання заданого набору завдань для i -го спостереження B_i = тривалість операцій вводу/виводу для виконання завдань для i -го спостереження n = Кількість спостережень
Використання смуги пропускання	Яка частка доступної пропускної здатності використовується для виконання заданого набору завдань?	$X = A / B$ A = пропускна здатність фактичної передачі, виміряна з часом для виконання заданого набору завдань B = пропускна здатність, доступна для виконання а заданий набір завдань
Ємність обробки транзакцій	Скільки транзакцій можна обробити за одиницю часу?	$X = A / B$ A = кількість транзакцій, виконаних за час спостереження B = тривалість спостереження
Ємність доступу користувача	Скільки користувачів може отримати доступ до системи одночасно в певний час?	$X = \sum_{i=1}^{n} A_i / n$ A_i = Максимальна кількість користувачів, які можуть одночасно отримати доступ до системи під час i -го спостереження n = Кількість спостережень
Доступ користувачів збільшує адекватність	Скільки користувачів можна успішно додати за одиницю часу?	$X = A / B$ A = кількість користувачів, успішно доданих під час спостереження B = тривалість спостереження

3.3 Показники сумісності

Показники сумісності використовуються для оцінки ступеня, в якому продукт, система або компонент можуть обмінюватися інформацією з іншими продуктами, системами або компонентами та/або виконувати необхідні функції, одночасно використовуючи одне і те ж апаратне чи програмне середовище.

До показників сумісності відносяться показники, що наведені в табл. 3.3.

Таблиця 3.3 – Показники сумісності

Назва	Опис	Функція вимірювання
Співіснування з іншими продуктами	Яка частка зазначених програмних продуктів може спільно використовувати навколишнє середовище з цим програмним продуктом без негативного впливу на їх якісні характеристики	$X = A / B$ A = Кількість інших зазначених програмних продуктів, з якими цей продукт може співіснувати B = Кількість інших програмних продуктів, визначених для спільного існування з цим продуктом в робочому середовищі

	чи функціональність?	
Можливість обміну форматами даних	Яка частка зазначених форматів даних обмінюється з іншим програмним забезпеченням чи системами?	$X = A/B$ A = Кількість форматів даних, якими можна обмінюватися з іншим програмним забезпеченням або системами B = Кількість форматів даних, визначених для обміну
Достатність протоколу обміну даними	Яка частка зазначених протоколів обміну даними підтримується?	$X = A/B$ A = Кількість підтримуваних протоколів обміну даними B = Кількість протоколів обміну даними, зазначених для підтримки
Адекватність зовнішнього інтерфейсу	Яка частка зазначених зовнішніх інтерфейсів (інтерфейсів з іншим програмним забезпеченням та системами) є функціональною?	$X = A/B$ A = Кількість зовнішніх інтерфейсів, які функціонують B = вказана кількість зовнішніх інтерфейсів

3.4 Показник зручності використання (юзабіліті)

Заходи юзабіліті використовуються для оцінки ступеня, в якому продукт або система можуть бути використані зазначеними користувачами для досягнення визначених цілей з ефективністю та задоволенням у певному контексті використання.

До показників зручності використання (юзабіліті) відносяться показники, що наведені в табл. 3.4.

Таблиця 3.4 – Показники зручності використання (юзабіліті)

Назва	Опис	Функція вимірювання
Повнота опису	Яка частка сценаріїв використання описана в описі продукту чи документах користувача?	$X = A/B$ A = Кількість сценаріїв використання, описаних в описі продукту або документах користувача B = Кількість сценаріїв використання продукту
Демонстраційне висвітлення	Яка частка завдань має демонстраційні функції, щоб користувачі могли визнати доцільність?	$X = A/B$ A = Кількість завдань з демонстраційними функціями B = Кількість завдань, які можуть отримати користь від демонстраційних функцій
Самоописність точки входу	Яка частка часто використовуваних цільових сторінок на веб-сайті пояснює мету веб-сайту?	$X = A/B$ A = Кількість цільових сторінок, які пояснюють призначення веб-сайту B = Кількість цільових сторінок на веб-сайті
Повнота керівництва користувача	Яка частка функцій описана досить детально в документації користувача та/або довідковому центрі, щоб дозволити корист-	$X = A/B$ A = Кількість функцій, описаних у документації користувача та/або довідковій установі, якщо потрібно B = Кількість реалізованих функцій, які пот-

Назва	Опис	Функція вимірювання
	тувачеві застосувати функції?	рібно задокументувати
Значення полів вводу за замовчуванням	Яка частка полів введення, які можуть мати значення за замовчуванням, автоматично заповнюються значеннями за замовчуванням?	$X = A/B$ A = Кількість полів введення, значення яких за замовчуванням були автоматично заповнені під час роботи B = Кількість полів введення, які можуть мати значення за замовчуванням
Зрозумілість повідомлень про помилки	Яка частка повідомлень про помилки вказує причину, через яку сталася помилка, і як її усунути?	$X = A/B$ A = Кількість повідомлень про помилки, в яких зазначається причина виникнення та пропонуються шляхи вирішення, де це можливо B = Кількість реалізованих повідомлень про помилки
Пояснювальний інтерфейс користувача	Яка частка інформаційних елементів та кроків, представлених користувачеві, дозволяють виконувати загальні завдання першим користувачем без попереднього вивчення чи навчання або звернення за допомогою до зовнішньої допомоги?	$X = A/B$ A = Кількість інформаційних елементів та кроків, які представлені таким чином, щоб користувач міг зрозуміти B = Кількість інформаційних елементів та кроків, необхідних для виконання загальних завдань для користувача, що вперше користується
Послідовність роботи	Наскільки інтерактивні завдання мають поведінку та зовнішній вигляд, що узгоджується як у межах завдання, так і серед подібних завдань?	$X = A/B$ A = Кількість повідомлень, які передають правильний результат або вказівки користувачеві B = Кількість реалізованих повідомлень
Функціональна настроюваність	Яку частку функцій та робочих процедур користувач може налаштувати для своєї зручності?	$X = A/B$ A = Кількість функцій та робочих процедур, які можна налаштувати для зручності користувача B = Кількість функцій та операційних процедур, для яких користувачі можуть отримати користь від налаштування
Настроюваність інтерфейсу користувача	Яку частку елементів інтерфейсу користувача можна налаштувати за зовнішнім виглядом?	$X = A/B$ A = Кількість елементів інтерфейсу користувача, які можна налаштувати B = Кількість елементів інтерфейсу користувача, які можуть отримати користь від налаштування
Можливість моніторингу	Яку частку функціональних станів можна контролювати під час роботи?	$X = A/B$ A = Кількість функцій, що мають можливість моніторингу стану B = Кількість функцій, які можуть отримати користь від можливості моніторингу
Скасувати можливість	Яка частка завдань, що має суттєві наслідки, надає можливість повторного підтвердження або скасування можливості?	$X = A/B$ A = Кількість завдань, які надають можливість скасування або запит на повторне підтвердження B = кількість завдань, для яких користувачі могли б отримати користь від повторного підтвердження або скасування можливості
Зрозуміла категоризація	Наскільки програмне	$X = A/B$

Назва	Опис	Функція вимірювання
інформації	забезпечення організовує інформацію за категоріями, які є звичними для передбачуваних користувачів та зручними для їхніх завдань?	$A =$ Кількість інформаційних структур, які знайомі та зручні для передбачуваних користувачів $B =$ Кількість використовуваних інформаційних структур
Послідовність зовнішнього вигляду	Яка частка користувацьких інтерфейсів з подібними елементами має подібний вигляд?	$X = 1 - A/B$ $A =$ Кількість інтерфейсів користувача з подібними елементами, але з різним зовнішнім виглядом $B =$ Кількість інтерфейсів користувача з подібними елементами
Підтримка пристрою введення	Якою мірою завдання можуть ініціювати всі відповідні способи введення (наприклад, клавіатура, миша чи голос)?	$X = A/B$ $A =$ Кількість завдань, які можуть ініціювати всі відповідні способи введення $B =$ Кількість завдань, які підтримує система
Уникнення помилки роботи користувача	Яка частина дій та входів користувача захищена проти спричинення будь-якої несправності системи?	$X = A/B$ $A =$ Кількість дій та входів користувача, які захищені від будь-якої несправності системи $B =$ Кількість дій та входів користувача, які можуть бути захищені від будь-якої несправності системи
Виправлення помилок входу користувача	Наскільки система надає запропоноване правильне значення для виявлених помилок входу користувача з ідентифікованою причиною?	$X = A/B$ $A =$ Кількість помилок введення, для яких система надає запропоноване правильне значення $B =$ Кількість виявлених помилок введення
Можливість відновлення помилок користувача	Яку частку помилок користувача може виправити чи відновити система?	$X = A/B$ $A =$ Кількість помилок користувача, які розроблені та протестовані для відновлення системою $B =$ Кількість помилок користувача, які можуть виникнути під час роботи
Естетика зовнішнього вигляду інтерфейсів користувача	Наскільки користувацькі інтерфейси та загальний дизайн мають естетичний вигляд?	$X = A/B$ $A =$ Кількість інтерфейсів дисплея, естетично приємних для користувачів зовні $B =$ Кількість інтерфейсів дисплея
Доступність для користувачів з обмеженими можливостями	Наскільки потенційні користувачі з особливими вадами можуть успішно використовувати систему (за допомогою допоміжних технологій, якщо це доречно)?	$X = A/B$ $A =$ Кількість функцій, які успішно використовуються користувачами з певною вадою $B =$ Кількість реалізованих функцій
Адекватність підтримуваних мов	Яка частка необхідних мов підтримується?	$X = A / B$ $A =$ Кількість фактично підтримуваних мов $B =$ Кількість мов, які потрібно підтримувати

3.5 Показники надійності

Показники надійності використовуються для оцінки ступеня виконання системою, продуктом чи компонентом певних функцій за певних умов протягом певного періоду часу.

Внутрішні показники надійності використовуються для прогнозування того, чи завершена система/програмний продукт, про який йде мова, задовольнить встановленим потребам в надійності під час розробки системи/програмного продукту.

Зовнішні показники якості надійності використовуються для оцінки атрибутів, пов'язаних з поведінкою системи, частиною якої є програмне забезпечення під час тестування виконання, щоб вказати ступінь надійності програмного забезпечення в цій системі під час роботи. Системи та програмне забезпечення в більшості випадків не відрізняються одна від одної.

До показників надійності відносяться показники, що наведені в табл. 3.5.

Таблиця 3.5 – Показники надійності

Назва	Опис	Функція вимірювання
Виправлення несправностей	Яка частка виявлених несправностей, пов'язаних із надійністю, була виправлена?	$X = A/B$ A = Кількість несправностей, пов'язаних з надійністю, виправлених на етапі проектування/кодування/випробування B = Кількість несправностей, пов'язаних з надійністю, виявлених на етапі проектування/кодування/тестування
Середній час між відмовою (СЧМЖ)	Що таке СЧМЖ під час роботи системи/програмного забезпечення?	$X = A/B$ A = час роботи B = Кількість системних/програмних збоїв, що насправді відбулися
Інтенсивність відмов	Яка середня кількість несправностей протягом визначеного періоду?	$X = A/B$ A = Кількість відмов, виявлених під час спостереження B = Тривалість спостереження
Покриття тесту	Який відсоток можливостей системи, програмного забезпечення, сценаріїв функцій або функцій, включених до відповідних наборів тестів, виконуються насправді?	$X = A/B$ A = Кількість можливостей системи чи програмного забезпечення, сценаріїв роботи або функцій, які фактично виконуються B = Кількість системних або програмних можливостей, сценаріїв роботи або функцій, які включені до відповідних наборів тестів
Доступність системи	На яку частку запланованого часу роботи системи дійсно доступна система?	$X = A/B$ A = фактично наданий час роботи системи B = час роботи системи, зазначений у графіку роботи
Середній час простою	Як довго система зали-	$X = A/B$

Назва	Опис	Функція вимірювання
	шається недоступною при виникненні несправності?	$A =$ Загальний час простою $B =$ Кількість поломок, що спостерігаються
Уникнення відмов	Яка частка моделей несправностей була взята під контроль, щоб уникнути критичних та серйозних збоїв?	$X = A/B$ $A =$ Кількість критичних та серйозних випадків відмови (на основі тестових випадків) $B =$ Кількість виконаних тестових випадків шаблону несправності (майже спричиняє несправність) під час тестування
Резервування компонентів	Яка частина компонентів системи встановлюється з резервуванням, щоб уникнути збою системи?	$X = A/B$ $A =$ Кількість системних компонентів, встановлених з резервуванням $B =$ кількість компонентів системи
Середній час повідомлення про несправність	Як швидко система повідомляє про виникнення несправностей?	$X = \sum_{i=1 \text{ to } n} (A_i - B_i) / n$ $A_i =$ Час, коли система повідомляє про несправність i $B_i =$ час, коли виявлено несправність i $n =$ Кількість виявлених несправностей
Середній час відновлення	Скільки часу потрібно для відновлення програмного забезпечення системи після несправності?	$X = \sum_{i=1 \text{ to } n} A_i / n$ $A_i =$ Загальний час для відновлення збитого програмного забезпечення/системи та відновлення операції для кожного збою i $n =$ кількість відмов
Повнота даних резервного копіювання	Яка частка елементів даних регулярно резервується?	$X = A/B$ $A =$ Кількість елементів даних, для яких виконується регулярне резервне копіювання $B =$ кількість елементів даних, що вимагають резервного копіювання для відновлення після помилок

3.6 Показники безпеки

Заходи безпеки використовуються для оцінки ступеня, в якій продукт або система захищає інформацію і дані, щоб люди або інші продукти або системи мали ступінь доступу до даних, відповідну їх типам і рівням авторизації.

До показників безпеки відносяться показники, що наведені в табл. 3.6.

Таблиця 3.6 – Показники безпеки

Назва	Опис	Функція вимірювання
Керування доступом	Яка частина конфіденційних даних захищена від несанкціонованого доступу?	$X = 1 - A/B$ $A =$ кількість конфіденційних даних, до яких можна отримати доступ без авторизації. $B =$ кількість елементів даних, що вимагають управління доступом
Правильність шифру-	Наскільки правильно здійс-	$X = A/B$

Назва	Опис	Функція вимірювання
вання даних	нюється шифрування/дешифрування елементів даних, як зазначено в специфікації вимог?	A = Кількість елементів даних, зашифрованих/розшифрованих правильно B = Кількість елементів даних, які потребують шифрування/дешифрування
Сила криптографічних алгоритмів	Яка частка криптографічних алгоритмів була перевірена?	X = 1-A/B A = Кількість криптографічних алгоритмів, порушених або неприпустимо ризикованих у використанні B = Кількість використаних криптографічних алгоритмів
Цілісність даних	Якою мірою попереджено пошкодження або зміна даних в результаті несанкціонованого доступу?	X = 1-A/B A = кількість елементів даних, які фактично пошкоджені несанкціонованим доступом B = кількість елементів даних, для яких необхідно запобігти пошкодження або зміну даних
Запобігання пошкодження внутрішніх даних	Якою мірою реалізовані доступні методи запобігання пошкодженню даних?	X = A/B A = Кількість фактично реалізованих методів запобігання пошкодженню даних B = Кількість доступних і рекомендованих методів запобігання пошкодженню даних
Запобігання переповненню буфера	Яку частину доступу до пам'яті за допомогою введення користувачем програмних модулів було здійснено, перевіряючи межі для запобігання переповненню буфера?	X = A/B A = Кількість звернень до пам'яті з призначеним для користувача введенням, межі яких перевірені B = кількість звернень до пам'яті з призначеним для користувача введенням в програмних модулях
Використання цифрового підпису	Яка частка подій, що потребують невідмовності, обробляється за допомогою цифрового підпису?	X = A/B A = кількість подій, що забезпечують невідмовність з використанням цифрового підпису. B = кількість подій, що вимагають невідмовності з використанням цифрового підпису
Повнота журналу аудиту користувача	Наскільки повним є контрольний журнал доступу користувачів до системи або даних?	X = A/B A = кількість доступів, записаних у всіх журналах B = кількість доступів до системи або даним, фактично протестованим
Зберігання системного журналу	Протягом якого відсотка необхідного періоду зберігання системний журнал зберігається в стабільному сховищі?	X = A/B A = Тривалість, протягом якої системний журнал фактично зберігається у стабільному сховищі B = Період зберігання, вказаний для збереження системного журналу в стабільному сховищі
Достатність механізму аутентифікації	Наскільки добре система ідентифікує особистість суб'єкта?	X = A/B A = кількість наданих механізмів аутентифікації (наприклад, ідентифікатор користувача) B = кількість зазначених механізмів аутентифікації
Відповідність правилам аутентифікації	Яка частка необхідних правил аутентифікації встановлена?	X = A/B A = кількість реалізованих правил аутентифікації

Назва	Опис	Функція вимірювання
		V = кількість зазначених правил аутентифікації

3.7 Показники ремонтпридатності

Показники ремонтпридатності використовуються для оцінки ступеня ефективності та дієвості, з якими продукт або система можуть бути змінені наміченими фахівцями з обслуговування.

До показників безпеки відносяться показники, що наведені в табл. 3.7.

Таблиця 3.7 – Показники ремонтпридатності

Назва	Опис	Функція вимірювання
Зчеплення компонентів	Наскільки сильно компоненти незалежні та скільки компонентів не мають впливів від змін інших компонентів у системі чи комп'ютерній програмі?	$X = A/V$ A = Кількість компонентів, які реалізовані без впливу на інші V = Кількість зазначених компонентів, які повинні бути незалежними
Адекватність цикломатичної складності	Скільки програмних модулів мають прийнятну цикломатичну складність?	$X = 1 - A/V$ A = Кількість програмних модулів, які мають цикломатичну оцінку складності, що перевищує вказаний поріг V = Кількість реалізованих програмних модулів
Можливість повторного використання активів	Скільки активів в системі можна використовувати повторно?	$X = A/V$ A = кількість активів, які спроектовані і реалізовані для повторного використання V = кількість активів в системі
Відповідність правилам кодування	Скільки модулів відповідає необхідним правилам кодування?	$X = A/V$ A = Кількість програмних модулів, що відповідають правилам кодування для певної системи V = Кількість реалізованих програмних модулів
Повнота системного журналу	Якою мірою система записує свої операції в журнали, щоб їх можна було відстежувати?	$X = A/V$ A = кількість журналів, які фактично записані в системі V = кількість журналів, для яких потрібні контрольні журнали під час роботи
Ефективність функції діагностики	Яка частина діагностичних функцій відповідає вимогам причинно-наслідкового аналізу?	$X = A/V$ A = кількість діагностичних функцій, корисних для причинно-наслідкового аналізу V = кількість реалізованих діагностичних функцій
Достатність функції діагностики	Яка частка необхідних функцій діагностики була реалізована?	$X = A/V$ A = Кількість реалізованих діагностичних функцій

Назва	Опис	Функція вимірювання
		$V =$ Кількість необхідних діагностичних функцій
Ефективність модифікації	Наскільки ефективно здійснюються модифікації порівняно з очікуваним часом?	$X = \sum_{i=1 \text{ to } n} (A_i / B_i) / n$ $A_i =$ Загальний робочий час, витрачений на внесення певного типу модифікації і $B_i =$ очікуваний час для внесення конкретного типу модифікації і $n =$ кількість виміряних модифікацій
Правильність модифікації	Яка частка модифікацій була впроваджена правильно?	$X = 1 - (A/V)$ $A =$ Кількість модифікацій, які спричинили інцидент або збій протягом певного періоду після впровадження $V =$ Кількість впроваджених модифікацій
Можливість модифікації	Якою мірою необхідні зміни вносяться в зазначений термін?	$X = A/V$ $A =$ кількість елементів, фактично змінених за вказаний період $V =$ кількість елементів, які необхідно змінити протягом зазначеного періоду
Повнота тестової функції	Наскільки повно реалізовані функції і можливості тестування?	$X = A/V$ $A =$ Кількість тестових функцій, реалізованих, як зазначено $V =$ Кількість необхідних тестових функцій
Автономне тестування	Як самостійно можна протестувати програмне забезпечення?	$X = A/V$ $A =$ кількість тестів, які можуть бути змодельовані за допомогою заглушки серед тестів, які залежать від інших систем $V =$ кількість тестів, які залежать від інших систем
Можливість перезапуску тесту	Наскільки легко можна провести перевірку працездатності з моменту перезапуску після технічного обслуговування?	$X = A/V$ $A =$ Кількість випадків, в яких супроводжуючий може призупинити і перезапустити виконання тестового прогону в бажаних точках для покрокової перевірки $V =$ кількість випадків, в яких виконання тестового прогону може бути призупинено

3.8 Показники переносимості

Показники переносимості використовуються для оцінки ступеня ефективності та дієвості, з якими система, продукт або компонент можуть бути перенесені з одного обладнання, програмного забезпечення або іншого операційного середовища або середовища використання в інше.

До показників переносимості відносяться показники, що наведені в табл.3.8.

Таблиця 3.8 – Показники переносимості

Назва	Опис	Функція вимірювання
Адаптивність обладнання до навколишнього середовища	Чи достатньо здатне програмне забезпечення або система адаптуватися до іншого апаратного середовища?	$X = 1 - A/V$ A = Кількість функцій, які не були завершені, або результатів, які були недостатні для задоволення вимог під час тестування V = кількість функцій, які були протестовані в різних апаратних середовищах
Адаптивність системного програмного забезпечення до навколишнього середовища	Чи достатньо здатне програмне забезпечення або система адаптуватися до іншого системного програмного середовища?	$X = 1 - A/V$ A = Кількість функцій, які не були завершені, або результатів, які були недостатні для задоволення вимог під час тестування V = кількість функцій, які були протестовані в різних програмних середовищах системи.
Адаптивність операційного середовища	Чи достатньо здатне програмне забезпечення або система адаптуватися до різного операційного середовища?	$X = 1 - A / V$ A = кількість функцій, які не були завершені, або результатів, які були недостатні для задоволення вимог під час експлуатаційного тестування в призначеному для користувача середовищі V = кількість функцій, які були протестовані в різних операційних середовищах
Економія часу на установки	Наскільки ефективний фактичний час установки в порівнянні з очікуваним?	$X = \sum_{i=1}^{ton} (A_i / B_i) / n$ A _i = Загальний час, витрачений на установку i B _i = очікуваний час для установки i n = кількість вимірених установок
Легкість установки	Чи можуть користувачі або фахівці з обслуговування налаштувати процедуру установки для своєї зручності?	$X = A/V$ A = Кількість випадків, коли користувачеві вдається налаштувати процедуру встановлення V = Кількість випадків, коли користувач намагався налаштувати процедуру встановлення для зручності користувача
Схожість використання	Яка частина призначених для користувача функцій заміненого продукту може бути виконана без додаткового навчання або обхідного шляху?	$X = A/V$ A = Кількість користувацьких функцій, які можна виконувати без будь-якого додаткового навчання або обхідних шляхів V = Кількість функцій користувача в заміненому програмному продукті
Еквівалентність якості продукції	Яка частина показників якості задовольняється після заміни попереднього програмного продукту на цей?	$X = A/V$ A = Кількість показників якості нового продукту, які краще або рівні заміненому продукту V = Кількість актуальних показників якості заміненого програмного продукту
Функціональна інклюзивність	Чи можна з легкістю використовувати аналогічні функції після заміни попереднього програмного продукту на цей?	$X = A / V$ A = кількість функцій, які дають такі ж результати, як і раніше V = кількість функцій, які повинні використовуватися в програмному продукті, якій замінюється
Можливість повторного використання/імпорту даних	Чи можна використовувати ті самі дані після заміни поперед-	$X = A/V$ A = Кількість даних, які можна використовувати безперервно, як і раніше

Назва	Опис	Функція вимірювання
	нього програмного продукту на цей?	$V =$ Кількість даних, які будуть постійно використовуватися в заміненіму програмному продукті

3.9 Вимірювання якості продукту по SQuaRE

На основі аналізу розділів моделі якості та вимірювання якості SQuaRE пропонується алгоритм вимірювання якості програмного забезпечення:

- 1) визначити моделі якості по ISO/IEC 25010 для ідентифікації відповідних характеристик якості програмного забезпечення;
- 2) вибрати показники якості для кожної характеристики якості, використовуючи ISO/IEC 25023.
- 3) використовуючи методи вимірювання з ISO/IEC 25021 виміряти елементи показника якості;
- 4) обрані показники якості визначаються шляхом застосування функції вимірювання до елементів показника якості.

У таблиці 3.9 показані приклади застосування стандартів SQuaRE для вимірювання якості програмного забезпечення.

Таблиця 3.9 – Застосування стандартів SQuaRE для вимірювання якості програмного забезпечення

Характеристики/ підхарактеристики якості	Показник якості. Опис	Функція вимірювання	Елемент показника якості	Метод вимірювання
Функціональна повнота	Функціональне покриття. Яка частина зазначених функцій реалізована?	$X = 1 - A / B$, A – кількість відсутніх функцій; B – кількість перебачених опцій	Кількість доступних функцій	Перегляд і аналіз окремих функцій системи/програмного забезпечення, доступних користувачеві з обмеженими можливостями для виклику і виконання, і підрахунок кількості функцій, які не вдалося успішно використувати

Часові характеристики	Середній час відповіді. Скільки в середньому часу потрібно системі, щоб відповісти на призначене для користувача завдання або системну задачу?	$X = \sum_{i=1}^n A_i / n,$ A_i – час, витрачений системою на відповідь на конкретне завдання користувача або системну задачу при і-му вимірюванні, n – кількість виміряних відповідей	Тривалість	Тривалість залежить від загальної кількості часу і прив'язана до Міжнародної системи одиниць (VIM)
-----------------------	---	---	------------	--

➤ 4 ОЦІНКА ЯКОСТІ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Нормативна база та загальні поняття

Вимоги та рекомендації з оцінки якості програмної продукції наведено в стандарті ДСТУ ISO/IEC 25040 [7]. Він надає опис процесу оцінки якості програмної продукції та містить вимоги зі застосування цього процесу.

Відповідно до стандарту ДСТУ ISO/IEC 25040:

оцінка – систематичне визначення ступеня, з яким якийсь об’єкт задовольняє встановленим критеріям

метод оцінки – процедура, яка описує дії, які будуть виконуватися оцінщиком для отримання результатів заданого вимірювання, зробленого для конкретних компонентів або продукції в цілому.

оцінка програмного забезпечення – систематичне дослідження ступеня, з яким програмна продукція здібна задовольняти встановленим вимогам.

4.2 Еталонна модель оцінки якості програмного продукту

Процес оцінки якості програмного продукту включає вхідні та вихідні дані, обмеження та ресурси для процесу оцінки якості програмного продукту, як показано на рисунку 4.1.

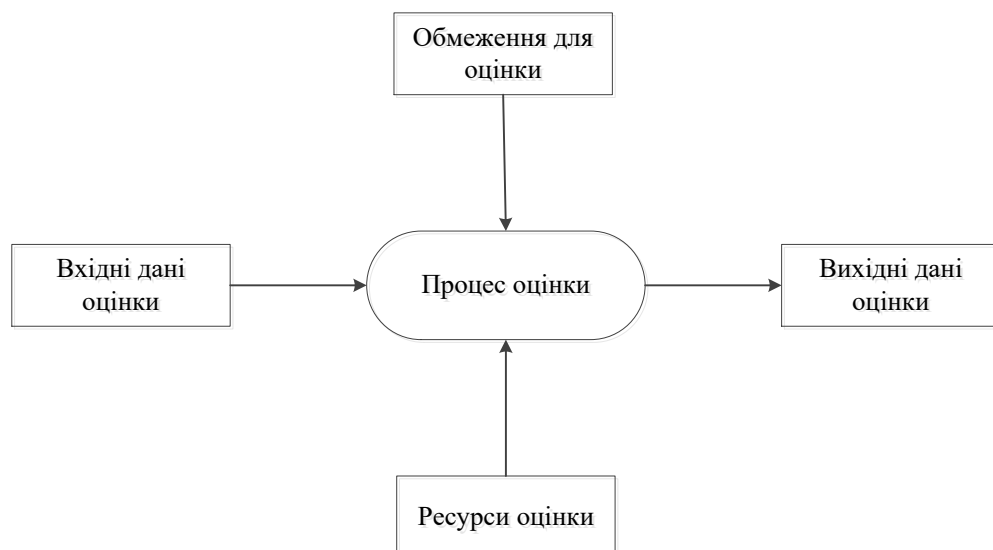


Рисунок 4.1 – Загальний вид оцінки якості програмного продукту

Обмеження для процесу оцінки якості програмного продукту можуть включати наступне:

- конкретні потреби користувача;
- ресурси;
- розклад;
- вартість;
- середовище;
- інструменти та методологію;
- створення звітів.

Ресурси для процесу оцінки якості програмного продукту можуть включати наступне:

- застосовні вимірювальні інструменти та методологію, включаючи модулі оцінки;
- застосовні документи SQaRE (ISO/IEC 25001, ISO/IEC 25010, ISO/IEC 2502n, ISO/IEC 25030, ISO/IEC 25041, ISO/IEC 25042);
- людські ресурси для оцінки якості програмного продукту;
- економічні ресурси з метою оцінки якості програмного продукту;
- інформаційну систему з метою оцінки якості програмного продукту;
- базу знань для оцінки якості програмного продукту.

Еталонну модель оцінки якості програмного продукту повинні застосовувати особи відповідальні за оцінку програмного продукту. Але вона підходить і для організацій, які виконують ролі покупців, розробників чи оцінювачів. Вона призначена для розробників, покупців та незалежних оцінювачів програмних продуктів, але не обмежена цим списком.

Еталонна модель оцінки якості програмного продукту має на увазі, що оцінка повинна ґрунтуватися на специфікації вимог до якості програмного продукту з використанням перед оцінкою стандарту ISO/IEC 25030 та роз'ясненням

цілей та критеріїв оцінки. Стандарт ISO/IEC 25030 забезпечує вимоги та рекомендації щодо специфікації вимог до якості програмного продукту.

При цьому застосовні інші стандарти SQuaRE, такі як ISO/IEC 25010 та ISO/IEC 2502n.

Оцінка якості програмного продукту може бути виконана організаціями розробника, покупцями або незалежним оцінювачем у процесі або після процесу розробки або в процесі придбання.

Еталонна модель процесу оцінки якості програмного продукту описує процес та деталізує дії та завдання, що забезпечують їх цілі, а також надає додаткову методичну інформацію, яку можна використовувати для проведення оцінки якості програмного продукту (рис. 4.2).

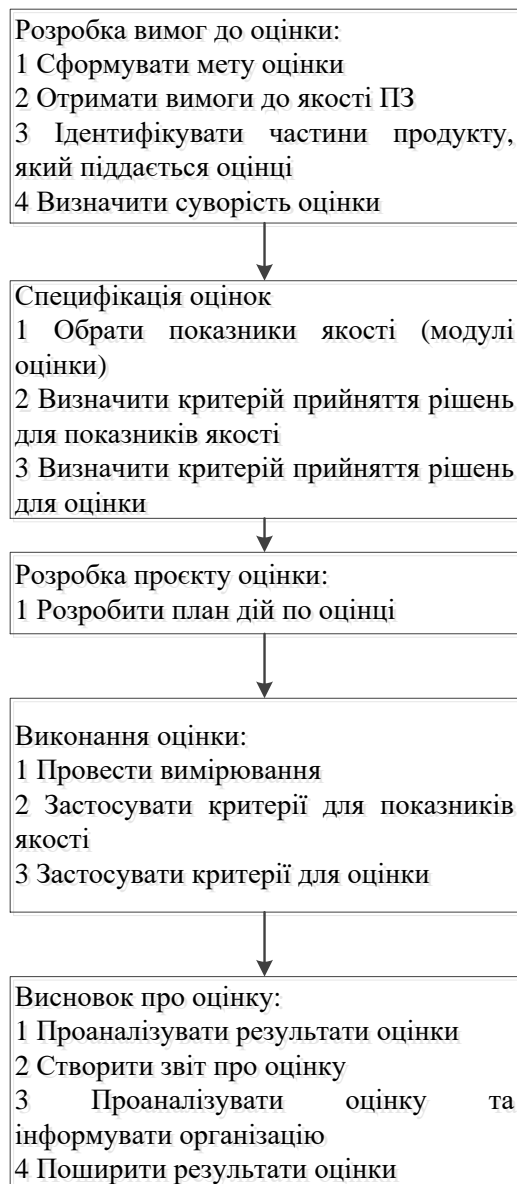


Рисунок 4.2 – Процес оцінки якості програмного продукту

Важливо, що реалізації процесу оцінки притаманна гнучкість, необхідна, щоб підлаштуватися під унікальність кожного застосунку, уникнути непотрібної роботи чи незначної роботи, і навіть забезпечити практичні засоби встановлення необхідної впевненості у програмному забезпеченні.

У різних зацікавлених сторін щодо оцінки якості програмного продукту, включаючи покупців, розробників, незалежних оцінювачів, постачальників, операторів та фахівців з обслуговування, різні цілі:

- покупець: замовляючи програмний продукт, що виготовляється, покупець може встановити вимоги до якості використання та вимоги до якості програмного забезпечення, визначити вимоги до постачальника та оцінити відповідність потенційно-можливої купівлі цим вимогам перед придбанням. При замовленні продукту, який буде розроблено, мета визначення вимог до якості полягає в тому, щоб забезпечити відповідність продукту заявленим потребам користувача. При покупці програмного продукту оцінку можна використовувати для того, щоб порівняти альтернативні продукти та гарантувати відповідність вибраного продукту вимогам якості;

- розробник: розробляючи програмний продукт, що виготовляється на замовлення, розробник може оцінити проміжні програмні продукти або кінцевий продукт для того, щоб забезпечити якість розробленого програмного забезпечення. Розробник може використовувати результати оцінки програмного забезпечення якості продукції для того, щоб продукти відповідали необхідним критеріям якості, які могли бути встановлені замовником або в порівнянні з іншими продуктами;

- незалежні оцінювачі: для забезпечення якості програмного забезпечення, оцінюючи цільовий програмний продукт, незалежний оцінювач може оцінити проміжні програмні продукти або кінцевий продукт. Незалежний підхід до процесу оцінки забезпечує вимоги та рекомендації для практичної реалізації оцінки програмного продукту в умовах, коли результати оцінки повинні бути зрозумілі, прийняті декількома сторонами за умови довіри результатам. Цей підхід і використовується оцінювачами під час виконання незалежної оцін-

ки програмного продукту. Подібна оцінка може бути виконана на вимогу розробника, покупця або іншої сторони;

– постачальник: постачальник може використовувати результати оцінки програмного продукту для забезпечення відповідності продуктів необхідним критеріям якості, які можуть бути встановлені покупцем, або в порівнянні з іншими продуктами;

– оператор: людина або організація, які керують системою, частиною якої є програмний продукт, можуть використовувати оцінку якості програмного забезпечення, щоб переконатися в тому, що вимоги до рівня якості виконуються в різних експлуатаційних режимах, та висловити свою думку щодо необхідності будь-яких змін відповідальним за технічне обслуговування особам;

– спеціаліст з обслуговування, людина або організація, які обслуговують систему, частиною якої є програмний продукт, можуть використовувати оцінку програмного забезпечення для перевірки виконання вимог до рівня якості, вимог щодо придатності до обслуговування та мобільності.

Підтримка оцінки - заходи, призначення яких допомогти в оцінці шляхом збору інформації щодо методів оцінки якості програмного продукту, інструментів для розробки та перевірки показників та стандартизації процесів оцінки та вимірювань.

ISO/IEC 25001 містить вимоги та рекомендації для підтримки процесів оцінки якості програмного продукту, а також для покращення процесу оцінки організаційного рівня за допомогою, наприклад, кваліфікації кожного проекту оцінки та власне процесу оцінки.

ISO/IEC 25041 містить спеціальні вимоги та рекомендації для розробників, покупців та оцінювачів.

ISO/IEC 25042 визначає структуру та склад модулів оцінки. Модулі оцінки описують як технологію оцінки, так і порядок застосування технології

4.3 Реалізація процесу оцінки якості програмного продукту

4.3.1 Мета та забезпечення процесу оцінки якості програмного продукту

Мета – провести оцінку якості програмного продукту.

Оцінювач повинен реалізувати дії та завдання, описані в підрозділі 4.3 відповідно до застосовних організаційних політик та процедур, що стосуються процесу оцінки якості програмного продукту (рис.4.2).

Інфраструктура повинна дозволяти збір даних та модифікацію процесу на основі аналізу даних.

Персонал, який бере участь в оцінці, повинен мати необхідні навички та підготовку.

Для того щоб забезпечити повторюваність, відтворюваність, неупередженість та об'єктивність результатів оцінки, оцінювач повинен діяти в організаційних умовах, які забезпечують усі необхідні гарантії для отримання достатньої якості своєї діяльності.

4.3.2 Документація

Кожна дія в оцінці якості програмного продукту має бути задокументована.

Протокол повинен включати докладний звіт про дії, які виконує оцінювач при реалізації плану оцінки якості програмного продукту.

Протокол повинен містити вичерпну інформацію, необхідну для управління оцінкою якості програмного продукту.

Протокол оцінки повинен містити будь-які проміжні дані, на яких базується будь-яка інтерпретація. Рішення, прийняті під час інтерпретації, повинні також бути включені в протокол оцінки, як це визначено в плані оцінки.

Протокол повинен мати достатню інформацію для кожної дії для ефективного здійснення наступних дій оцінки якості програмного продукту.

Протокол повинен бути організований таким чином, щоб забезпечити документування оцінки якості програмного продукту та дозволити повторно опрацьовувати результати оцінки.

Повинен бути підготовлений звіт про оцінку якості програмного продукту, в якому мають бути записані заходи щодо оцінки та результати оцінки.

У разі використання для виконання дій оцінки будь-якого інструменту необхідно, щоб посилання на інструмент було включено до звіту про оцінку. Посилання має складатися з ідентифікації інструмента, його постачальника та версії інструменту.

Більш докладне посилання на інструмент, що використовується, має бути включено до протоколу оцінки. У нього повинні бути включені докладна конфігурація інструменту, і вся інша інформація, необхідна для того, щоб було можливо повторити заходи щодо оцінки та отримати той самий проміжний результат.

4.3.3 Вимоги оцінки

Вхідною інформацією для цієї дії має бути таке:

- потреби в оцінці якості програмного продукту;
- специфікація вимог щодо якості програмного продукту;
- програмний продукт, включаючи проміжні продукти, який оцінюється.

Результатами цієї дії має бути таке:

- специфікація цілей оцінки якості програмного продукту;
- специфікація вимог оцінки якості програмного продукту;
- специфікація високорівневого плану оцінки якості програмного продукту.

Мета оцінки якості програмного продукту має бути задокументована як основа для подальших дій та завдань оцінки.

Оцінка якості програмного продукту безпосередньо підтримує і розробку, і придбання програмного забезпечення, яке відповідає потребам користувача та споживачів. Кінцева мета полягає в тому, щоб переконатися, що продукт забезпечує необхідну якість – що він відповідає встановленим потребам користувачів (включаючи операторів, споживачів результатів програмного забезпечення та фахівців з обслуговування програмного забезпечення).

Якість програмного продукту може бути оцінена в межах певної структури якості на всіх етапах життєвого циклу, що відносяться до процесу розробки

програмного забезпечення та процесу придбання та визначених у процесах життєвого циклу програмного забезпечення ISO/IEC 12207 та процесах життєвого циклу систем ISO/IEC 15288.

Якість програмного продукту можна оцінити або як якість проміжного продукту, або як якість кінцевого продукту. Мета оцінки якості проміжного програмного продукту:

- забезпечити якість продукту;
- ухвалити рішення про приймання проміжного програмного продукту від субпідрядника;
- оцінити поточну доцільність проєкту розробки;
- ухвалити рішення про завершення етапу життєвого циклу та часу переходу продукту до наступного етапу;
- передбачити чи оцінити якість кінцевого програмного продукту;
- зібрати інформацію щодо проміжних програмних продуктів, щоб контролювати і керувати процесом.

Ціль оцінки якості кінцевого програмного продукту може полягати в наступному:

- ухвалити рішення про приймання продукту;
- визначити термін виходу продукту;
- порівняти продукт з конкурентоспроможними продуктами;
- вибрати продукт з числа альтернативних продуктів;
- оцінити позитивні та негативні сторони продукту при його використанні;
- визначити причину невдачі в дослідженні;
- вирішити, що час поліпшити чи замінити продукт.

4.3.4 Підготовка до введення оцінки якості програмного продукту

Вхідними даними для цієї дії має бути таке:

- специфікація цілей оцінки якості програмного продукту;
- специфікація вимог щодо оцінки якості програмного продукту;

- специфікація високорівневого плану оцінки якості програмного продукту.

Результатами цієї дії має бути таке:

- специфікація вибраних показників якості.
- специфікація критеріїв ухвалення рішення для показників якості програмного продукту;
- специфікація критеріїв ухвалення рішення для оцінки якості програмного продукту;
- специфікація переглянутого високорівневого плану оцінки якості програмного продукту.

Ця дія складається з таких завдань.

Оцінювач повинен вибрати показники якості (модулі оцінки) таким чином, щоб покрити всі вимоги щодо оцінки якості програмного забезпечення.

Вимоги щодо оцінки якості програмного продукту повинні бути визначені для кожного відповідного компонента програмного продукту, щоб можна було визначити відповідний показник (міру) якості, який використовується далі для оцінки якості програмного продукту.

Методи оцінки якості програмного продукту мають бути задокументовані з урахуванням дій, які будуть робитися для отримання результатів оцінки. Якщо обраний метод оцінки буде ґрунтуватися на використанні програмного інструменту, то цей інструмент повинен бути визначений у плані оцінки. У ідентифікацію інструменту повинні бути включені щонайменше назва інструменту, номер його версії та його джерело (наприклад, постачальник). Опис методу оцінки має бути завершено ідентифікацією компонентів продукту, до яких цей метод буде застосовано. Якщо для експертної оцінки результатів вимірювань потрібна інтерпретація (дешифрування) результатів, то специфікації оцінки має бути визначена процедура інтерпретації.

Специфікація оцінки повинна включати:

- область застосування оцінки з посиланнями на компоненти продукту, як це визначено у специфікації продукту;

- перехресні посилання між інформацією, необхідною для виконання оцінки, компонентами продукту та іншими відповідними документами, перерахованими у специфікації продукту;
- специфікацію вимірювань та перевірок, які будуть виконуватись та посилання на компоненти продукту, для яких вони повинні бути виконані;
- відображення між специфікацією вимірювань та перевірок та вимогами оцінки разом із посиланням на стандарти чи обґрунтування для кожного вимірювання чи перевірки у списку.

Для конкретних обраних показників мають бути визначені критерії рішення.

Оцінювач повинен підготувати процедуру для подальшого узагальнення з окремими критеріями для різних характеристик якості, кожна з яких може або ґрунтуватися на окремих підхарактеристиках та показниках якості, або бути виваженою комбінацією підхарактеристик та показників якості. Результати узагальнення слід використовувати як основу для оцінки якості програмного продукту.

4.3.5 Розроблення оцінки якості програмного продукту

Вхідними даними для цієї дії має бути таке:

- специфікація вимог щодо оцінки якості програмного продукту;
- специфікація вимог щодо якості програмного продукту;
- специфікація вибраних показників якості (модулі оцінки);
- специфікація критеріїв рішення для показників якості програмного продукту;
- специфікація критеріїв рішення для оцінки якості програмного продукту;
- специфікація переглянутого високорівневого плану оцінки якості програмного продукту.

Результатами цієї дії має бути таке:

- специфікація детального плану оцінки якості програмного продукту;

- специфікація методів оцінки якості програмного продукту.

Розробка оцінки складається з таких завдань.

4.3.6 Виконання оцінки якості програмного продукту

Вхідними даними для цієї дії має бути таке:

- специфікація детального плану оцінки якості програмного продукту;
- специфікація вимог щодо оцінки якості програмного продукту;
- специфікація вимог щодо якості програмного продукту;
- специфікація вибраних показників якості;
- специфікація критеріїв рішення для показників якості програмного продукту;
- специфікація критеріїв рішення для оцінки якості програмного продукту;
- специфікація методів оцінки якості програмного продукту;
- програмний продукт, який буде оцінений, включаючи проміжні продукти.

Результатами цієї дії має бути таке:

- результати вимірювання якості програмного продукту;
- результати оцінки.

Ця дія складається з таких завдань:

1) провести вимірювання

Відповідно до плану оцінки необхідно провести обрані вимірювання якості програмного продукту та компонента для отримання результуючих значень на шкалах вимірювань;

2) застосувати критерії рішення для показників якості

До вимірних значень необхідно застосувати критерії рішення показників якості програмного продукту.

3) Застосувати критерії рішення для оцінки

Набір критеріїв рішення має бути зведений у підхарактеристиці та характеристики, внаслідок чого оцінка результатів зводиться до формулювання, якою мірою програмний продукт відповідає вимогам якості.

Результати оцінки мають:

- створити відповідний ступінь впевненості у тому, що програмний продукт здатний задовольнити вимоги оцінки;
- виявити будь-які конкретні недоліки щодо вимог оцінки та всі необхідні додаткові оцінки для визначення рамок цих недоліків;
- виявити будь-які конкретні обмеження або умови, що накладаються на використання програмного продукту;
- визначити будь-які слабкі місця в самій оцінці та в будь-якій іншій додатковій оцінці;
- ідентифікувати будь-які варіанти використання програмного продукту, розкриті оцінкою.

4.3.7 Завершення оцінки

Вхідними даними для цієї дії має бути таке:

- специфікація вимог щодо оцінки якості програмного продукту;
- специфікація фактичних результатів плану оцінки якості програмного продукту;
- специфікація методів оцінки якості програмного продукту;
- результати оцінки.

Результатами цієї дії має бути таке:

- звіт про оцінку якості програмного продукту.

Ця дія складається з таких завдань.

Оцінювач і сторона, що запитує, повинні виконати об'єднаний аналіз результатів оцінки.

Залежно від того, кому призначено звіт про оцінку, він має містити таке.

- вимоги щодо оцінки якості програмного продукту;
- вимоги до якості програмного продукту;

- план оцінки якості програмного продукту;
- результати виконаних вимірювань та досліджень;
- проміжні результати або рішення інтерпретації, якщо це визначено щодо оцінки;
- будь-які обмеження, що обмежують вплив, недоліки або виключення в дії оцінки, у тому числі їх вплив на використання, налаштування, зміну або загальне обслуговування програмного продукту з часом.
- оцінювачів та їх кваліфікацію;
- будь-які розбіжності між оціненими версіями продукту та відповідними вхідними даними оцінки, наприклад, документація або навчальні посібники;
- вирішення проблем або обхідні рішення у разі неповноцінності;
- будь-яку іншу інформацію, необхідну для повторення чи відтворення оцінки;
- результати оцінки.

В результаті аналізу дій оцінки звіт про оцінку має визначати:

- 1) для кожного недоліку – повний відповідний аналіз і як недолік було вирішено. Усунення недоліків може включати наступне:
 - один із методів оцінки встановив, що недолік не є значним;
 - можна знайти задовільне обхідне рішення, що пом'якшує вплив недоліку, наприклад, модифікація продукту, відключення або видалення непотрібних функцій, відновлення відсутніх конструктивних вимог із використанням реверсивного проектування;
 - вихідна вимога не є обов'язковою, і недолік може бути прийнятним;
 - недолік прийнятний за умови, що використання програмного продукту буде контролюватись особливими умовами або обмеженнями;
 - для усунення нестачі або прогалин в оцінці потрібні додаткові заходи оцінки;
- 2) будь-які додаткові оцінки, які виконуються для усунення всіх виявлених недоліків:

- визначити область дії або вплив недоліку;
- упевнитися, що жодних недоліків немає,
- перевірити, що обхідне рішення технічно здійсненне та/або підходить і прийнятно;
- перевірити правильність та прийнятну продуктивність програмного забезпечення після внесення зміни до проєкту або самого продукту для виправлення недоліку;

3) якщо необхідно обмежити або контролювати використання програмного продукту:

- обмеження заважає програмному продукту виконувати обов'язкові вимоги;
- обмеження впливає на проєкт застосунку, бюджет та розклад;
- обмеження вимагає додаткової роботи оцінки;
- обмеження забезпечує можливість будь-якої відмови програми;

4) будь-які винятки з області застосування оцінки та/або обмежень для результатів кожної оцінки, такі як:

- «Ця оцінка не містить детального аналізу функціональних можливостей продукту.» або
- «Цей програмний продукт вважається кваліфікованим до необхідного рівня цілісності, якщо повну оцінку необхідної функціональності продукту завершено успішно».

5) повинні бути отримані узагальнені результати всіх заходів, що дозволяють зробити загальний висновок щодо оцінки програмного продукту.

Оцінювач повинен розглянути результати оцінки та придатність процесу оцінки, індикаторів та вчинених дій. Зворотний зв'язок для аналізу має використовуватися для вдосконалення процесу та методів оцінки (модулі оцінки). При необхідності покращення модулів оцінки потрібно включити збір даних для додаткових показників, щоб провадити їх для подальшого використання.

Після завершення оцінки даними оцінки необхідно розпорядитися відповідно до вимог запитуючої сторони.

Це має бути зроблено одним із наступних способів залежно від типу даних:

- документи, надані для оцінки, повинні бути або повернені стороні, що запитує, або заархівовані на зазначений період часу, або знищені безпечним способом;
- звіт про оцінку та протокол оцінки мають бути заархівовані на вказаний час;
- всі інші дані мають бути заархівовані на вказаний період часу або знищені у безпечний спосіб.

Якщо для деяких даних закінчиться зазначена тривалість архівації, то вони повинні бути знову заархівовані на вказаний період часу або знищені безпечним способом.

ВИСНОВКИ

Розділ «Моделі якості» 2501n стандартів SQuaRE описує моделі якості програмного забезпечення, які підтримують чітке визначення вимог до якості програмного забезпечення. Характеристики в моделі якості при використанні і моделі якості продукту призначені для використання в якості набору при специфікації або оцінці якості програмного продукту або комп'ютерної системи.

Вимірювання якості програмного забезпечення засноване на двох поняттях: показник якості та елемент показника якості. Еталонну модель вимірювання якості описує стандарт ISO/IEC 25020.

Стандарти ISO/IEC 25022, ISO/IEC 25023, ISO/IEC 25024 для кожної характеристики якості моделі визначає показники якості і функцію вимірювання.

Функція вимірювання пов'язує показники якості з елементами показника якості, який безпосередньо вимірюється. Досить широкий перелік елементів показників якості містить стандарт ISO/IEC 25021.

Стандарт ДСТУ ISO/IEC 25040 містить вимоги та рекомендації щодо оцінювання якості програмного продукту та роз'яснює загальні поняття. Він надає опис процесу для оцінки якості програмного продукту та визначає вимоги до застосування цього процесу. Процес оцінки можна використовувати для різних цілей і підходів. Процес може бути використаний для оцінки якості попередньо розробленого програмного забезпечення, комерційного готового програмного забезпечення або спеціального програмного забезпечення та може використовуватися під час або після процесу розробки.

Основні переваги серії стандартів SQuaRE полягають в тому, що вони надають методики координації для вимірювання та оцінки якості програмних продуктів, керівництво по специфікаціям вимог до якості програмного забезпечення продукту і гармонізацію зі стандартом ISO/IEC 15939 в формі еталонної моделі вимірювання якості.

ПЕРЕЛІК ПОСИЛАНЬ

- 1 ДСТУ ISO/IEC 25000:2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Настанова до SQuaRE: чинний з 01.01.2018. – Київ. : УкрНДНЦ.
- 2 ДСТУ ISO 9000:2015. Системи управління якістю. Основні положення та словник термінів. – Київ. : УкрНДНЦ, 2016. – 49 с.
- 3 ДСТУ ISO/IEC 25010:2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів: чинний з 01.01.2018. – Київ. : УкрНДНЦ.
- 4 ДСТУ ISO/IEC/IEEE 15939:2018. Інженерія систем і програмних засобів. Процес вимірювання: чинний з 01.01.2019. – Київ. : УкрНДНЦ.
- 5 ДСТУ ISO/IEC 25023:2019. Інженерія систем і програмних засобів. Вимоги до якості систем програмних засобів та їхнього оцінювання (SQuaRE). Вимірювання якості систем та програмних продуктів: чинний з 01.11.2019. – Київ. : УкрНДНЦ.
- 6 ДСТУ ISO/IEC 25020:2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Рамкова модель і настанова щодо вимірювання: чинний з 01.01.2018. – Київ. : УкрНДНЦ.
- 7 ДСТУ ISO/IEC 25040:2016 (ISO/IEC 25040:2011, IDT) Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Процес оцінювання