

ГЛАВА 8

ПІДХОДИ ДО ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ і постановка задачі

Забезпечення якості програмного забезпечення (ПЗ) є невід'ємною складовою процесу розроблення і використовуються в ІТ-індустрії фахівцями з тестування і фахівцями щодо забезпечення якості. Як забезпечення якості, так і тестування програмного застосунку покликані гарантувати якість програмного продукту, яка відповідає вимогам замовника. Однак ці два поняття мають принципову відмінність. Тестування проводиться після того, як продукт був створений або для статичного тестування після того, як були визначені і зафіксовані у відповідному документі вимоги до програмного забезпечення. Забезпечення якості передбачає діяльність, яка забезпечує якість продукту на всіх етапах його створення.

Для розуміння відмінностей між цими складовими процесу розроблення програмного забезпечення потрібно дати чітке визначення цих понять, провести співвідношення між їх характеристиками, визначити методи оцінювання якості програмного забезпечення.

Успішне вирішення проблем якості програмного забезпечення можливе лише за умов відповідної організації роботи команд з забезпечення якості і тестувальників, тому в роботі будуть визначені відмінності між обов'язками цих команд, відмінності в плануванні випробувань і документації, а також розроблені рекомендації для поліпшення організації роботи команд.

1. Поняття забезпечення якості, контролю якості та тестування програмного забезпечення

Для з'ясування відмінностей між поняттями тестування і забезпечення якості програмного забезпечення, розглянемо пов'язані з ними поняття "тестування", "контроль якості" та "забезпечення якості".

Тестування програмного забезпечення (Testing) відповідно до ISO/IEC TR 19759: 2005 – це процес дослідження, випробування програмного забезпечення, який має на меті перевірку відповідності між реальною поведінкою програми і її очікуваним поведінкою на кінцевому наборі тестів, обраних певним чином.

Контроль якості (Quality Control, QC) відповідно до ISO 9000 - це частина менеджменту якості, зосереджена на виконанні вимог по відношенню до оцінки кількості дефектів, багів (при їх наявності) в продукті. Контроль якості являє собою набір процесів (дій), спрямованих на оцінку розробленого продукту (проекту документа, системи розвитку тощо) і відповідності вимогам замовника. Виконання цих процесів гарантує перевірку якості поставленого продукту і визначає, наскільки добре він спроектований і виконаний. Мета контролю якості полягає в пошуках дефектів і забезпеченні їх виправлення. Таким чином, тестування є невід'ємною частиною контролю якості (рис. 1).

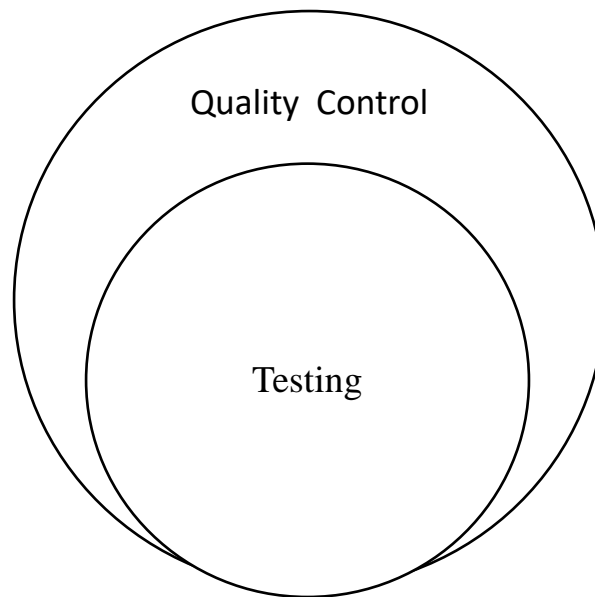


Рис. 1. Співвідношення понять "тестування" і "контроль якості"

Забезпечення якості (Quality Assurance, QA) в стандарті ISO 9000 визначається як частина менеджменту якості, орієнтовану на створення впевненості в тому, що вимоги до усунення багів будуть виконані. Метою забезпечення якості є гарантія того, що продукт буде відповідати вимогам замовника. Забезпечення якості складається з процесів, спрямованих на забезпечення якості розробки продукту на кожному з етапів життєвого циклу. Ці дії, як правило, передують розвитку продукту і тривають, поки процес перебуває в стані розвитку. На забезпечення якості лежить відповідальність за розробку і впровадження процесів і стандартів для поліпшення життєвого циклу розробки,

і забезпечення впевненості в тому, що ці процеси виконуються [1, 2]. Основне призначення забезпечення якості - запобігання дефектів на всіх етапах створення програмного забезпечення і постійне його вдосконалення. У той час, як забезпечення якості є діяльністю, спрямованою на забезпечення розробки якісного програмного забезпечення, контроль якості - діяльністю, яка фіксує і оцінює якість вже створеного продукту. На рис. 2. показано співвідношення між забезпеченням якості, контролем якості та тестуванням. Діяльність із забезпечення якості включає встановлення стандартів і процесів, перевірки якості, а також вибір відповідних інструментів.

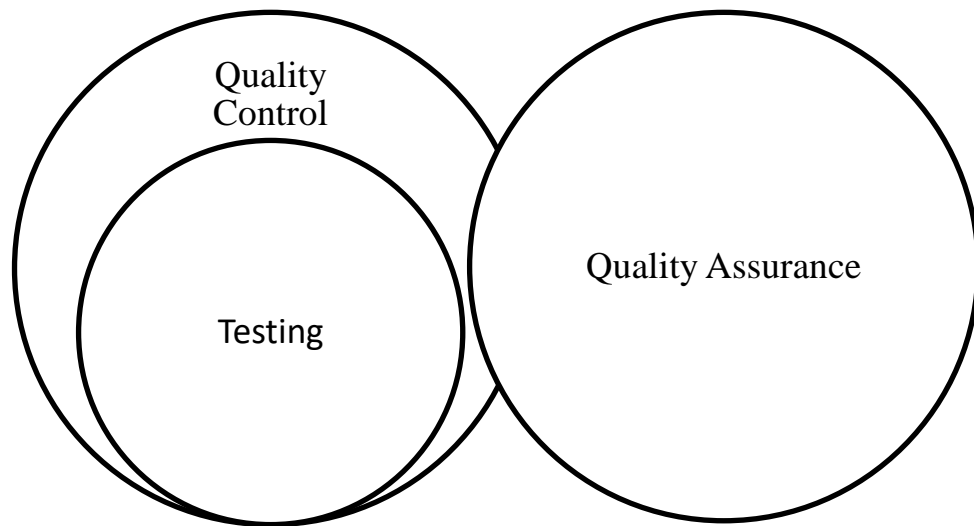


Рис. 2. Співвідношення між QA, QC і Testing

2. Багаторівнева модель якості програмного забезпечення

Якість програмного забезпечення визначається в стандарті ISO 9126 як вся сукупність його характеристик, що належать до можливості задовольняти висловлені або ті, що мають на увазі потреби всіх зацікавлених осіб.

Виділяють наступні аспекти якості ПЗ [3 – 5]:

1. Якість технологічних процесів розробки ПЗ, яка впливає на створення якісного ПЗ;
2. Внутрішня якість ПЗ, пов'язана з його характеристиками, без урахування поведінки програмного продукту;
3. Зовнішня якість, що характеризує ПЗ із точки зору його поведінки;

4. Якість ПЗ при використанні у різних контекстах, тобто якість програмного продукту, яка проявляється при його використанні користувачами в різних конкретних сценаріях роботи.

Для всіх цих аспектів якості створені метрики, які дозволяють їх оцінити

На рис. 3 показаний зв'язок різних аспектів якості ПЗ.

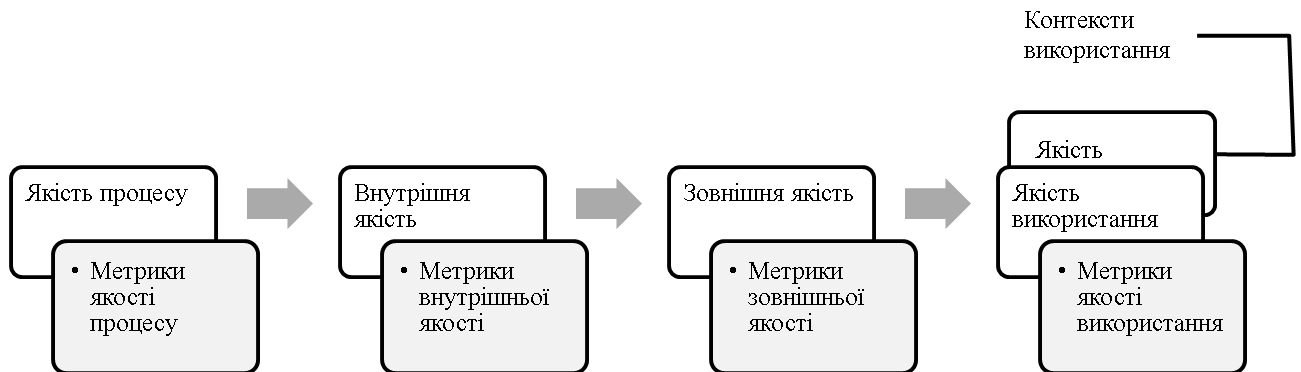


Рис. 3. Зв'язок різних аспектів якості ПЗ згідно ISO 9126

Крім того, в стандарті описана багаторівнева модель якості програмного забезпечення, яку можна використовувати для опису як внутрішньої, так і зовнішньої якості ПЗ (рис. 4). На верхньому рівні моделі виділено 6 основних характеристик якості ПЗ, кожна з яких має свої атрибути:

функціональність: здатність до взаємодії, функціональна придатність, відповідність стандартам і правилам, захищеність, точність;

надійність: завершеність, здатність до відновлення, відповідність стандартам, стійкість до відмов;

зручність використання: зрозумілість, зручність навчання, зручність роботи, привабливість, відповідність стандартам;

продуктивність: часова ефективність, ефективність використання ресурсів, відповідність стандартам;

зручність супроводу: аналіз, зручність внесення змін, стабільність, зручність перевірки, відповідність стандартам;

перенесення: адаптованість, зручність установки, здатність до співіснування, зручність заміни, відповідність стандартам.



Рис. 4. Багаторівнева модель якості ПЗ згідно ISO 9126

Для кожного атрибута визначається набір метрик, які дозволяють його оцінити. Метрики повинні мати такі властивості:

1) надійність; яка пов'язана з випадковою помилкою; метрика вільна від випадкової помилки, якщо випадкові зміни не впливають на результати метрики;

2) повторюваність; повторне використання метрики для того ж продукту і тими ж фахівцями з оцінювання при використанні тієї ж специфікації оцінювання (включаючи навколишнє середовище), того самого типу користувачів і оточення, має привести до тих же результатів з відповідними допусками; відповідні допуски повинні враховувати такі компоненти, як втома і результат накопичених знань;

3) однотипність; застосування метрики для того ж продукту різними фахівцями з оцінювання при використанні тієї ж специфікації оцінювання (включаючи навколишнє середовище), того самого типу користувачів і оточення, має привести до тих же результатів з відповідними допусками;

4) можливість застосування; метрика повинна чітко вказувати умови (наприклад, наявність певних атрибутів), які обмежують її вживання;

5) показовість; це здатність метрики ідентифікувати частини або елементи програми, які повинні бути поліпшені, на підставі порівняння вимірних і очікуваних результатів;

б) коректність; метрика повинна мати наступні властивості:

об'єктивність; результати метрики і її вхідні дані повинні бути засновані на фактах і бути не підвладні почуттям або думкам фахівців з оцінювання або тестування (виключаючи метрики задоволеності або привабливості, за допомогою яких вимірюються почуття і думки користувача);

неупередженість; вимірювання не повинно бути направлено на отримання будь-якого специфічного результату;

адекватність точності; точність визначається при проектуванні метрики і особливо при виборі описів фактів, які використовуються як основа для метрики; розробник метрики повинен описати точність і чутливість метрики;

7) значимість; вимірювання має давати значущі результати, які стосуються поведінки програми або характеристик якості.

Метрика повинна також бути ефективною по відношенню до вартості. Це означає, що більш дорогі метрики повинні забезпечувати кращі результати оцінки.

Розробник метрики повинен довести її обґрунтованість. Метрика повинна задовольняти хоча б одному з наступних критеріїв обґрунтованості метрики:

1) кореляція; зміна в значеннях параметрів якості (оперативно визначених за результатами вимірювання основних метрик), обумовлена зміною в значеннях метрики, має визначатися лінійною залежністю;

2) трасування; якщо метрика M безпосередньо пов'язана з величиною характеристики якості Q , то зміна величини $Q(T_1)$, наявної в момент часу T_1 , до величини $Q(T_2)$, отриманої в момент часу T_2 , має супроводжуватися зміною значення метрики від $M(T_1)$ до $M(T_2)$ в тому ж напрямку (наприклад, якщо збільшується Q , то M теж збільшується);

3) несуперечливість; якщо значення характеристик якості (оперативно отриманих за результатами вимірювання основних метрик) Q_1, Q_2, \dots, Q_n , пов'язані з продуктами або процесами $1, 2, \dots, n$, визначаються співвідношенням $Q_1 > Q_2 > \dots > Q_n$, то відповідні значення метрики повинні задовольняти співвідношенню $M_1 > M_2 > \dots > M_n$;

4) передбачуваність; якщо метрика використовується в момент часу T_1 для прогнозу значення (оперативно отриманого за результатами вимірювання основних метрик) характеристики якості Q в момент часу T_2 , то помилка прогнозування

$$(Q_p(T_2) - Q_f(T_2)) / Q_f(T_2),$$

де $Q_p(T_2)$ – прогнозне значення характеристики якості в момент T_2 ,
 $Q_f(T_2)$ – фактичне значення характеристики якості в момент T_2 ,

повинна потрапляти в допустимий діапазон помилок прогнозування;

5) селективність; метрика повинна бути здатною розрізняти високу і низьку якість програмного засобу.

3. Методи визначення показників якості програмного забезпечення

Методи визначення показників якості програмного забезпечення можна класифікувати виходячи з ряду факторів, які представлені в табл. 1.

Таблиця 1

Класифікація методів визначення показників якості програмного забезпечення

Ознака	Метод
Спосіб отримання інформації	Вимірювальний
	Реєстраційний
	Органолептичний
	Розрахунковий
Джерело отримання інформації	Експертний
	Соціологічний
	Традиційний

Вимірювальний метод визначення показників якості програмного забезпечення – це метод отримання інформації про його властивості та характеристики шляхом вимірювань за допомогою інструментальних засобів (наприклад, кількість операторів в програмі, кількість виконаних операторів, кількість операндів, час виконання програми за певних наборах вхідних даних тощо).

Реєстраційний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики ПС під час його випробування або функціонування, коли реєструються певні події, наприклад, кількість збоїв, відмов.

Органолептичний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення, заснований на сприйнятті таких органів почуттів людини, як зір і слух, наприклад, зручність використання.

Розрахунковий метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення, заснований на використанні емпіричних і теоретичних залежностей на ранніх етапах його розроблення, накопичених при випробуваннях, експлуатації та супроводі програмного забезпечення статистичних даних,. Таким чином може визначатися, наприклад, точність обчислень.

Експертний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі думок групи експертів-фахівців, компетентних у вирішенні даного завдання на основі їх досвіду і інтуїції. Експертний метод застосовується, коли виконати оцінку показників якості за допомогою інших методів неможливо або потребує значних трудовитрат. За допомогою даного методу рекомендується визначати, наприклад, показники зрозумілості і опанування програмного забезпечення.

Соціологічний метод визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі обробки спеціальних анкет-опитувальників. Цим методом, наприклад, можуть визначатися окремі показники зручності використання.

Традиційний метод визначення показників якості програмного забезпечення - це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі безпосереднього спостереження за їх функціонуванням в процесі роботи. Цим методом, наприклад, можуть визначатися деякі з показників функціональності і зручності використання.

4. Оцінювання якості програмного забезпечення

Існують різні методи оцінювання якості програмних продуктів [3 – 5]. Дані методи дозволяють отримати підсумкову інтегральну величину якості програмного забезпечення в цілому або окремих його характеристик, виражену в певних кількісних показниках. Найбільш поширеними є методи інтегральної оцінки якості програмного забезпечення, заснований на витратах та оцінки якості на основі ієрархічної моделі

Метод інтегральної оцінки якості програмного забезпечення, заснований на витратах відноситься до групи розрахункових методів. Відповідно до даного методу формулюється кількісний критерій якості програмного забезпечення T , орієнтований на його життєвий цикл (ЖЦ).

Витрати на розроблення, експлуатацію та супровід програмного забезпечення включають:

R – одноразові витрати на розроблення програмного забезпечення;

V – одноразові витрати на впровадження програмного забезпечення.

E – повторювані витрати S на експлуатацію програмного забезпечення за період часу експлуатації t_e протягом життєвого циклу T :

$$E = (T / t_e) * S,$$

C – повторювані через випадкові проміжки часу витрати на супровід, які складають в середньому n -ту частину від витрат R і m -ту частину від витрат V і здійснюються протягом життєвого циклу T в середньому через періоди часу t_c :

$$C = (n * R + m * V) * T / t_c,$$

B – випадкові втрати через недостовірність або несвоєчасність результату:

$$B = S_e * T / t_e,$$

Тут S_e – середня сума втрат, яка припадає на одноразову експлуатацію програмного забезпечення протягом його ЖЦ.

Таким чином загальна сума витрат Z в процесі життєвого циклу програмного забезпечення на програмного забезпечення, буде визначатися таким чином:

$$Z = R + V + (S_e + S) * T / t_e + (n * R + m * V) * T / t_c.$$

Як критерій якості пропонується використовувати мінімізацію сумарних витрат на розроблення, експлуатацію та супровід програмного забезпечення. Критерієм якості програмного забезпечення є мінімізація загальної суми витрат Z :

$$Z \rightarrow \min.$$

Основним недоліком даного методу є те, що фактичні значення витрат, які входять в формулу, можна визначити після завершення розробки програмного застосунку, і отже його не можна використовувати як інструмент в процесі його розроблення для досягнення заданого рівня якості.

Відповідно до **методу оцінки якості на основі ієрархічної моделі** вибір номенклатури показників якості для конкретного програмного застосунку здійснюється з урахуванням його призначення і вимог до сфери застосування в залежності від приналежності програмного забезпечення до того чи іншого підкласу, який визначається класифікатором програмних продуктів:

- операційні системи і засоби їх розширення;
- програмні засоби управління базами даних;
- інструментально-технологічні засоби програмування;
- програмні продукти для інтерфейсу і управління комунікаціями;
- програмні продукти організації обчислювального процесу (планування, контролю);
- сервісні програми;
- програмні продукти обслуговування обчислювальної техніки;
- прикладні програми для наукових досліджень;
- прикладні програми для проектування;
- прикладні програми для управління технічними пристроями і технологічними процесами;

прикладні програми для вирішення економічних завдань;
інші програмного продукту.

Оцінювання якості ПЗ полягає у виборі номенклатури показників, їх оцінюванні і зіставленні з базовими значеннями. Основу описуваного методу оцінювання якості становить чотирирівнева ієрархічна модель якості, яка включає:

- 1 рівень - характеристики якості ;
- 2 рівень - атрибути якості;
- 3 рівень - метрики;
- 4 рівень – оціночні показники (атрибути ПЗ).

Для кожної з обраних характеристик якості складається чотирирівнева ієрархічна модель, яка відображає взаємозв'язок характеристик, атрибутів, метрик і показників. Вид даної моделі залежить від фази ЖЦ.

Для практичного застосування моделі використовуються таблиці, які створюються для кожної характеристики. Так для оцінювання характеристики "Надійність" можна використовувати показники, які є в табл. 2.

Таблиця 2

Показники оцінки характеристики "Надійність"

Показник	Метод оцінювання	Форма оцінювання
Вплив помилок в вхідних даних на стійкість функціонування	Експертний	Шкала 0-1
Можливість оброблення помилок	Експертний	Шкала 0-1
Імовірність безвідмовної роботи, P	Реєстраційний, розрахунковий	$P=1 - q/n$, де n – кількість випробувань, q- кількість число зареєстрованих відмов

5. Вдосконалення роботи QA команд і команд тестувальників

Потреба в команді з забезпечення якості ПЗ зростає з розміром організації та рівнем її політики в області якості. Там, де потрібно така команда, дуже важливо, щоб функція QA залишалася незалежною від менеджменту проектом і операційної діяльності. Але зв'язки між QA команди з командою проекту повинні надавати їм потужну підтримку.

Деякі організації мають функцію QA, вбудовану в офіс управління проектами. Така модель також відповідає критеріям незалежності. Однак, за такої організації треба бути впевненим, що QA команда складається з кваліфікованих аналітиків з забезпечення якості.

З огляду на відмінності розглянутих понять тестування програмного забезпечення, контролю якості і забезпечення якості, є також відмінності між обов'язками QA команд і команд тестувальників.

Обов'язки команд. В обов'язки команд тестувальників входить:
планування тестування,
написання тестових сценаріїв і тест кейсів, перевірку тестів,
виконання тестів,
аналіз результатів тестування,
створення і аналіз звітності про результати тестування для різних рівнів випробувань.

В рамках своєї ролі з контролю якості тестувальники можуть висувати вимоги щодо:

перевірки зразків документів проекту,
діяльності з управління конфігураціями програмного забезпечення, дизайну, коду тощо.

В той же час QA команди виконують такі функції:
формування організаційної політики щодо якості, стандартів і процесів розроблення;

надання допомоги з підготовкою в області забезпечення якості та планів забезпечення якості проекту;

перевірка відповідності між процесами проекту і планами якості;
проведення регулярних перевірок проектних продуктів і процесів, а також регулярне представлення результатів аналізу оцінок якості для вищого керівництва;

загострення ситуації з відхиленням від керівних принципів або стандартів. В рамках своєї ролі щодо забезпечення якості QA-команда виконує контроль:

- незалежних оглядів;
- наявності процедур управління змінами проєктів;
- наявності процедур управління конфігураціями проєктів;
- наявності ретроспективи планування і виконання процесів життєвого циклу розробки;
- забезпечення якості на основі розвитку системи життєвого циклу;
- проведення безперервного поліпшення в процесі контролю якості і втілення рекомендацій, заснованих на отриманому раніше досвіді.

Виконання обов'язків QA команди ще означає їх розвиток командою, а тільки забезпечення їх реалізації.

Планування випробувань і документація. При плануванні випробувань тестувальники готують тестові стратегії і плани, засновані на базових тестових документах, таких як вимоги до програмного продукту і проєктні рішення. Ці документи планування тестування є основою випробування процесів на різних запланованих випробувальних рівнях. Для кожного рівня випробувань складаються тести, набори вхідних даних і очікуваних результатів, докладні графіки випробувань, екологічні вимоги, управління дефектами, управління тестуванням і ведення звітності. На відміну від цього, документація з забезпечення якості програмного продукту або плани якості включають більш широкий набір дій протягом усіх етапів розроблення. Це відіграє свою роль при розробленні методології управління проєктами.

Типовий проєкт плану якості включає в себе очікування клієнтів, критерії приймання, плановий контроль якості і аудиту процесів, плани управління конфігураціями і процедури управління змінами. При складанні планів щодо якості виходять з політики самої організації в цій сфері, стандартів або керівних принципів, які формують основи забезпечення якості. В ході створення проєкту моніторинг плану забезпечення якості проєкту здійснюється безперервно і на його основі оновлюються заплановані показники якості діяльності. Існують різні точки перетину між управлінням ризиками і якістю, і тому реєстр ризиків може зробити вагомий внесок в підготовку планів якості.

Рекомендації щодо поліпшення роботи команд. Перед QA командами встає цілий ряд проблем щодо організації їх роботи, а саме:

незалежність. Для успіху QA команди повинні бути незалежними від команди проекту. Це забезпечує QA команді можливість проведення об'єктивної оцінки проектів. В невеликих організаціях тестувальники і QA фахівці можуть перебувати в одній команді. Однак з'являється можливість створення конфлікту інтересів при моніторингу діяльності тестування. Рішенням проблеми в залежності від політики самої організації в сфері якості може бути створення окремої команди для формування звітності;

відносини всередині команди. Якщо аналітики забезпечення якості занадто процесно-орієнтовані і наполягають на процесах або документації, які можуть не мати особливого значення для проекту, це може погіршувати відносини з керівниками проекту. QA команді буде набагато легше працювати з проектними групами, якщо вони будуть працювати за принципом врахування цілей проекту. Крім того надання допомоги і сприяння з боку проектних команд формує основу для підтримки хороших відносин, що є важливим аспектом успішного тестування;

залучення потрібних фахівців. Для успішної діяльності QA команд провідну роль відіграє якісна кадрова політика. Люди з досвідом в області розробки ЖЦ програмного забезпечення є хорошими кандидатами для QA команди. Деякі знання в рамках стандартів ISO і принципів СММІ є додатковими корисними знаннями;

контрольні списки вимог. Стандартні контрольні списки є корисним механізмом для проведення аудиту проектів, особливо якщо вони розроблені відповідно до фазами ЖЦ. Для плідної співпраці з менеджерами проекту важливим є забезпечення участі в проекті зацікавлених сторін. Це дає можливість отримати від них зворотній зв'язок у відповідь на пропозиції щодо внесення змін до списків;

зв'язок і звітність. Хоча регулярна звітність для вищого керівництва дуже важлива, розроблення правильних шаблонів і показників для забезпечення топ-менеджерів необхідною інформацією гарантує, що цими звітами приділяється належна увага. Це найкраще досягається шляхом проведення зустрічей з відповідними представниками вищого керівництва, надання їм звітів і отримання від них відгуків і коментарів. Крім того, QA командам необхідно постійно

отримувати схвалення на внесення змін до процесів контролю якості та стандартів і забезпечувати ефективну взаємодію з зацікавленими сторонами;

постійне вдосконалення. Врахування попереднього досвіду забезпечує QA-команду основою для оцінювання процесів і рекомендаціями щодо забезпечення якості, включаючи постійні поліпшення. QA-команда повинна проявляти гнучкість, підтримувані хороші відносини із зацікавленими сторонами при внесенні поліпшення в управлінську звітність. Безперервні удосконалення можуть також вимагати внесення змін до методології розроблення системи, тому QA командам рекомендується зберігати методологію розробки IT-відділу.

Висновки

Зіставлення відмінностей між поняттями "тестування", "контроль якості" та "забезпечення якості" показало, що тестування є частиною контролю якості, а контроль якості співпадає з забезпеченням якості в площині перевірки якості.

Оцінювання якості програмного забезпечення повинно враховувати міжнародні стандарти в цій галузі, які визначають різні аспекти якості, такі як якість процесу, внутрішня якість, зовнішня якість та якість використання. Для визначення якості використовується багаторівнева модель, яка включає такі характеристики, функціональність, надійність, зручність використання, продуктивність, зручність супроводу, перенесення. Визначення якості програмного забезпечення здійснюється з використанням методів інтегральної оцінки якості та окремих показників якості.

Для оцінювання відмінностей між процесами з контролю якості та забезпечення якості був проведений аналіз їх обов'язків відповідних команд, планування їх роботи та документації, що дало можливість порівняти виконувані функції й умови роботи. Так QC підтверджує, що конкретні результати відповідають стандартам і специфікаціям, а QA є ширшою функцією, яка охоплює планування і контроль на протязі всього життєвого циклу розроблення. Тестування ж є невід'ємною частиною контролю якості. Для того, щоб в IT-компанії процеси управління якістю були ефективними команди QA і QC повинні співпрацювати.

Успішна QA команда може додати значну цінність для організації, а саме:
підвищення якості програмних продуктів;
послідовність в процесах постачання програмних продуктів;

вдосконалення організації процесів;
зменшення загальних витрат на поставку;
використання застосунків для документації щодо підтримки продукту.

В той же час потрібно враховувати, що для QA-команди потрібні додаткові витрати в штатному розкладі для аналітиків забезпечення якості ПЗ та ускладнення процесів, які можуть викликати розчарування в деяких співробітників

Література

1. Ушакова І.О. Проектування інформаційних систем : практикум / І. О. Ушакова. – Х. : ХНЕУ ім. С. Кузнеця, 2015. – 236 с. (Укр. мов.)
2. Ушакова І.О. Методика управління вимогами в гнучких методологіях / І.О. Ушакова. // Збірник наукових праць ХНУПС. – 2018.– Вип. 2(56). – С. 93 – 98.
3. Foidl H., Felderer M. Integrating software quality models into risk-based testing // Software Quality Journal. –V 26. -2018. – P. 809 – 847.
4. Jacob P.M., Mani P. A framework for evaluating performance of software testing tools // International Journal of Scientific and Technology Research. –V. 9. – Issue 2. – 2020. P. 2175–2180.
5. Pietrantuono R. On the testing resource allocation problem: Research trends and perspectives //Journal of Systems and Software. –V. 161. – 2020. – 42 p..