

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів забезпечення якості ІТ-проєкту з розробки
підсистеми «Лабораторія контролю якості» ІС сирзаводу

(тема)

Виконала:

здобувач 2 року навчання,
групи УПГІТМ-23-1

Катерина ГУСИНА

(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проєктами
в галузі ІТ

(повна назва освітньої програми)

Керівник: ст.викл.каф. ІУС Катерини ЧИРКОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ІУС



(підпис)

Костянтин ПЕТРОВ

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Інформаційних управляючих систем _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)Освітня програма _____ Управління проектами в галузі інформаційних
технологій _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____  _____
(підпис)

“ 21 ” квітня 20 25 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**здобувачеві _____ Гусині Катерині Віталіївні _____
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів забезпечення якості ІТ-проекту з розробки
підсистеми «Лабораторія контролю якості» ІС сирзаводу _____

затверджена наказом по університету від “ 28 ” березня 2025 р. № 235Ст _____


2. Термін подання здобувачем роботи до екзаменаційної комісії “ 06 ” червня 2025 р. _____


3. Вихідні дані до роботи науково-технічна література, публікації та інтернет-ресурси
за темою кваліфікаційної роботи; матеріали передатестаційної практики _____4. Перелік питань, що потрібно опрацювати у роботі проведення аналізу предметної
області, формування постановки задачі дослідження, дослідження існуючих методів,
розробка комбінованого методу забезпечення якості підсистеми «Лабораторія контролю
якості» ІС сирзаводу, апробація методу _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання на кваліфікаційну роботу	21.04.2025 – 25.04.2025	Виконано
2	Проведення аналізу предметної області дослідження	26.04.2025 – 30.04.2025	Виконано
3	Формування постановки задачі	01.05.2025 – 03.05.2025	Виконано
4	Аналіз особливостей ІС у сфері харчової промисловості	04.05.2025 – 08.05.2025	Виконано
5	Огляд існуючих методів забезпечення якості	09.05.2025 – 11.05.2025	Виконано
6	Огляд специфічних вимог до підсистем контролю якості	12.05.2025 – 14.05.2025	Виконано
7	Порівняння методологій та підходів забезпечення якості	15.05.2025 – 20.05.2025	Виконано
8	Обґрунтування доцільності застосування підходів та їх комбінації	21.05.2025 – 24.05.2025	Виконано
9	Оцінка ефективності методів забезпечення якості	25.05.2025 – 27.05.2025	Виконано
10	Апробація комбінованого методу	27.05.2025 – 30.05.2025	Виконано
11	Оформлення пояснювальної записки	30.05.2025 – 02.06.2025	Виконано
12	Захист кваліфікаційної роботи	06.06.2025	

Дата видачі завдання 21 квітня 2025 р.

Здобувач 
(підпис)

Керівник роботи  ст.викл.каф. ІУС Катерина ЧИРКОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 89 с., 4 рис., 4 табл., 1 дод., 12 джерел.

ЗАБЕЗПЕЧЕННЯ ЯКОСТІ, ІТ-ПРОЄКТ, ПІДСИСТЕМА, ІНФОРМАЦІЙНА СИСТЕМА, ЛАБОРАТОРІЯ

Об'єктом дослідження кваліфікаційної роботи є процес забезпечення якості з розробки підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу.

Предметом дослідження є дослідження методів забезпечення якості з розробки підсистеми «Лабораторія контролю якості» в інформаційній системі сирзаводу.

Метою кваліфікаційної роботи є дослідження методів забезпечення якості ІТ-проєкту та розробка методу для покращення ефективності контролю якості підсистеми «Лабораторія контролю якості» для інформаційної системи сирзаводу.

У роботі було проведено дослідження методів забезпечення якості ІТ-проєкту з розробки підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу. Виконано постановку задач дослідження, проаналізовано особливості інформаційних систем у харчовій промисловості, зокрема роль процесу забезпечення якості з урахуванням специфіки функціонування лабораторного контролю. Визначено специфічні вимоги до підсистеми, проведено порівняльний аналіз і обґрунтовано доцільність застосування комбінованого підходу. Надано пропозиції щодо його оптимального впровадження для забезпечення стабільної якості програмного забезпечення в умовах виробничого середовища.

ABSTRACT

Master's thesis: 89 pages, 4 figures, 4 tables, 1 appendix, 12 sources.

QUALITY ASSURANCE, IT PROJECT, SUBSYSTEM, INFORMATION SYSTEM, LABORATORY.

The object of research of the qualification work is the quality assurance process for the development of the "Quality Control Laboratory" subsystem of the cheese factory information system.

The subject of the study is the study of quality assurance methods for the development of the "Quality Control Laboratory" subsystem in the cheese factory information system.

The purpose of the qualification work is to study methods for ensuring the quality of an IT project and develop a method to improve the efficiency of quality control of the "Quality Control Laboratory" subsystem for the cheese factory's information system.

The work conducted a study of quality assurance methods for an IT project to develop the "Quality Control Laboratory" subsystem of the cheese factory information system. The research tasks were formulated, the features of information systems in the food industry were analyzed, in particular, the role of the quality assurance process taking into account the specifics of the functioning of laboratory control. Specific requirements for the subsystem were determined, a comparative analysis was conducted, and the feasibility of using a combined approach was substantiated. Proposals were made for its optimal implementation to ensure stable software quality in a production environment.

ЗМІСТ

	С.
Скорочення та умовні позначки	8
Вступ.....	9
1 Огляд проблеми забезпечення якості ІТ-проєкту харчової промисловості.....	11
1.1 Аналіз актуальності дослідження використання методів забезпечення якості ІТ-проєкту	11
1.2 Аналіз структури інформаційної системи сирзаводу	12
1.3 Аналіз стандартів харчової галузі.....	22
1.4 Аналіз існуючих методологій управління ІТ-проєктами в контексті забезпечення якості	24
1.5 Постановка задачі дослідження	29
2 Дослідження методів забезпечення якості ІТ-проєкту з розробки підсистеми «Лабораторія контролю якості» ІС сирзаводу	31
2.1 Формування проблем та специфічних вимог до комбінованого методу з розробки підсистеми «Лабораторія контролю якості».....	31
2.2 Аналіз ризиків у розробці підсистеми.....	33
2.3 Визначення контрольних точок	35
2.4 Розробка комбінованого методу забезпечення якості підсистеми «Лабораторія контролю якості» ІС сирзаводу	43
3 Апробація комбінованого методу забезпечення якості ІТ-проєкту.....	52
3.1 Опис ІТ-проєкту з розробки підсистеми «Лабораторія контролю якості» ІС сирзаводу	52
3.2 Окремі задачі у контексті розробки проєкту та причини використання реалізації	56
3.3 Апробація комбінованого методу забезпечення якості	60

4	Опис практичного використання отриманих практичних та теоретичних результатів КР	73
4.1	Перевірка застосованості методу на практичному кейсі	73
4.2	Аналіз ефективності запропонованого підходу	74
4.3	Використання методу для навчання персоналу	75
4.4	Розширення методу на інші підсистеми ІС підприємства.....	77
	Висновки	79
	Перелік джерел посилання	80
	Додаток А Графічний матеріал кваліфікаційної роботи.....	82

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ІС – інформаційна система

ККТ – критична контрольна точка

ПЗ – програмне забезпечення

API – Application Programming Interface

BAS ERP – Business Automation Software ERP

CSRF – Cross-Site Request Forgery

CD – Continuous Delivery

CI – Continuous Integration

DevOps – development & operations

ERP – Enterprise Resource Planning

GAMP – Good Automated Manufacturing Practice

HACCP – Hazard Analysis and Critical Control Points

ISO – International Organization for Standardization

IEC – International Electrotechnical Commission

ISTQB – International Software Testing Qualification Board

MES – Manufacturing Execution Systems

RFID – Radio-Frequency Identification

SAP – Systems, Applications, and Products in Data Processing

SQL – Structured Query Language

UI – user interface

XSS – Cross-Site Scripting

ВСТУП

Сучасні інформаційні системи відіграють ключову роль у процесах автоматизації підприємств харчової промисловості, забезпечуючи ефективне управління виробничими циклами, обліком та контролем якості продукції. Зокрема, у сирозаводах, де якість продукції є визначальним фактором конкурентоспроможності, впровадження інформаційних систем дозволяє автоматизувати контрольні процедури, мінімізувати людський фактор та підвищити точність обліку.

Одним із важливих компонентів таких систем є підсистема «Лабораторія контролю якості», яка забезпечує моніторинг параметрів сировини, напівфабрикатів та готової продукції відповідно до встановлених стандартів. Для забезпечення її надійної та стабільної роботи необхідно застосовувати ефективні методи забезпечення якості ІТ-проєкту, що охоплюють усі етапи життєвого циклу розробки – від проєктування та тестування до впровадження та підтримки.

Актуальність дослідження обумовлена необхідністю підвищення якості програмного забезпечення підсистеми, що дозволить зменшити ризики виникнення помилок, забезпечити її коректне функціонування та відповідність вимогам харчової промисловості. У роботі розглядаються сучасні методи та підходи до забезпечення якості ІТ-проєктів, що дозволяють оптимізувати процес розробки, підвищити ефективність тестування та вдосконалити механізми контролю якості [1].

Метою кваліфікаційної роботи є дослідження методів забезпечення якості та розробки методу забезпечення якості ІТ-проєкту з розробки підсистеми «Лабораторія контролю якості» ІС сирзаводу. Об'єктом дослідження є процес забезпечення якості з розробки підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу, а предметом дослідження є дослідження методів забезпечення якості з розробки підсистеми

«Лабораторія контролю якості» в інформаційній системі сирзаводу.

Кваліфікаційна робота оформлена згідно методичних вказівок та за правилами ДСТУ [2,3].

1 ОГЛЯД ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ІТ-ПРОЄКТУ ХАРЧОВОЇ ПРОМИСЛОВОСТІ

1.1 Аналіз актуальності дослідження використання методів забезпечення якості ІТ-проєкту

Забезпечення якості є однією з ключових складових успішної реалізації будь-якого ІТ-проєкту, особливо коли йдеться про розробку критично важливих підсистем, які мають прямий вплив на технологічні або виробничі процеси. У сучасних умовах активної цифрової трансформації промисловості значення якісного програмного забезпечення стрімко зростає, адже від його стабільності, точності та відповідності вимогам залежить не лише ефективність роботи підприємства, а й дотримання стандартів безпеки, нормативів і вимог ринку. Саме тому дослідження методів забезпечення якості в ІТ-проєктах є актуальним і виправданим як у теоретичному, так і в практичному контексті.

Особливої актуальності ця тема набуває у сфері харчової промисловості, де якість програмного забезпечення напряму пов'язана з якістю кінцевої продукції. Підсистеми, що здійснюють лабораторний контроль, несуть на собі відповідальність за фіксацію, збереження та аналіз параметрів сировини і готової продукції [4]. Будь-яка помилка в роботі такого програмного продукту може мати серйозні наслідки: від втрати даних до порушення вимог до безпечності харчових продуктів, що, в свою чергу, може вплинути на здоров'я споживачів і репутацію підприємства. У цьому контексті забезпечення високої якості підсистеми контролю якості стає не просто технічним завданням, а стратегічною необхідністю.

Ще одним важливим фактором, який підсилює актуальність теми, є різноманіття підходів до управління якістю, що використовуються в сучасній ІТ-індустрії. Методології Agile, Development & Operations (DevOps), а також стандарти на зразок International Software Testing Qualification Board (ISTQB)

пропонують різні бачення ролі якості в життєвому циклі програмного забезпечення [5]. Деякі з них фокусуються на ранньому виявленні дефектів, інші – на автоматизації перевірок або формалізації тестування. Але не кожен підхід може бути ефективним у конкретних умовах виробництва, особливо коли мова йде про лабораторні системи, які мають забезпечувати простежуваність, відповідність технічним регламентам, зручність перевірки результатів і довготривале збереження інформації.

В умовах, коли вимоги до якості та безпеки виробництва зростають, а кількість складних, багаторівневих інформаційних систем постійно збільшується, існує потреба в ґрунтовному аналізі та обґрунтованому виборі або поєднанні методів, які найкраще відповідають завданням конкретного проєкту. Це потребує як аналітичного підходу до порівняння практик, так і врахування контексту – організаційного, технологічного й нормативного.

Таким чином, дослідження методів забезпечення якості ІТ-проєктів, зокрема на прикладі підсистеми лабораторного контролю якості, є актуальним з огляду на потребу в створенні надійного, відповідального та регламентованого програмного забезпечення. Воно має на меті не лише теоретичне узагальнення сучасних підходів, а й прикладне застосування результатів для підвищення ефективності розробки в умовах виробництва, що вимагає чіткого контролю й гарантування відповідності стандартам.

1.2 Аналіз структури інформаційної системи сирзаводу

Інформаційні системи в харчовій промисловості давно стали невід’ємною частиною ефективного управління виробництвом, контролю якості, логістики та обліку. Різні компанії впроваджують власні рішення або використовують адаптовані програмні продукти, спеціально розроблені для цієї галузі.

Одним із поширених прикладів є система Systems, Applications, and Products in Data Processing (SAP), яку використовують на великих підприємствах харчової галузі, зокрема й в молочному виробництві. Ця система дозволяє вести облік сировини, контролювати всі виробничі етапи, керувати фінансами та персоналом у єдиному середовищі. Наприклад, компанія Nestlé активно використовує рішення на базі SAP для комплексного управління своїми підприємствами в усьому світі, включаючи виробничі потужності з виготовлення молочних та сирних продуктів.

Інший приклад – система Trace One, яка застосовується для забезпечення простежуваності продукції в усьому ланцюгу постачання. Її використовують як малі, так і великі харчові компанії для контролю походження інгредієнтів, відстеження партій сировини, ведення сертифікатів та відповідності стандартам. Такий підхід дозволяє не лише підтримувати якість, а й оперативно реагувати у разі виявлення дефекту або необхідності відкликання продукції.

На локальному рівні в Україні часто використовуються адаптовані Enterprise Resource Planning (ERP) системи, як-от Business Automation Software ERP (BAS ERP): Підприємство з галузевими модулями для харчової промисловості. Наприклад, на деяких українських сирзаводах впроваджено модулі контролю якості та автоматичного складування, які працюють у зв'язці з обліковою системою. Це дозволяє підприємству автоматизувати процеси перевірки сировини, оптимізувати виробничий цикл та оперативно формувати партії продукції відповідно до вимог замовників.

Також існують спеціалізовані Manufacturing Execution Systems (MES) системи, які використовуються для управління саме виробничими процесами. Наприклад, на молокопереробних підприємствах застосовуються рішення на базі Siemens SIMATIC IT або Wonderware, які дозволяють автоматизувати контроль за пастеризацією, дозріванням, охолодженням та пакуванням сиру. Дані системи можуть інтегруватися з лабораторними модулями для перевірки якості продукції в режимі реального часу.

Загалом, сучасні підприємства харчової промисловості активно впроваджують інформаційні системи як загального, так і вузькогалузевого призначення. Успішне застосування таких рішень дозволяє досягати високих стандартів якості, ефективності та безпеки на всіх рівнях виробничої діяльності.

Під час розробки підсистеми «Лабораторія контролю якості» для інформаційної системи сирзаводу можна зіткнутися з низкою серйозних складностей, які потребують попереднього аналізу, технічної підготовки та уважного врахування вимог як технологічного, так і організаційного характеру.

Перш за все, однією з основних складностей є забезпечення точності та достовірності даних, які надходять в систему з лабораторного обладнання. У більшості випадків йдеться про фізико-хімічні або мікробіологічні показники сировини та готової продукції. Ці дані можуть бути зчитані автоматично через спеціальні інтерфейси або вводитися вручну лаборантами. В обох випадках є ризик похибок: у разі автоматизації – через несумісність обладнання або нестабільність каналів зв'язку, у разі ручного введення – через людський фактор. Тому ще на етапі проектування важливо передбачити механізми перевірки, дублювання та валідації результатів.

Іншою не менш важливою складністю є відповідність нормативним стандартам та регламентам, які регулюють якість молочних продуктів. Лабораторна підсистема повинна не лише зберігати результати аналізів, а й автоматично звіряти їх із допустимими значеннями, що передбачені державними стандартами, санітарними нормами чи технічними умовами. Ба більше, необхідно враховувати, що нормативи можуть змінюватися, а отже, система повинна мати гнучку структуру, яка дозволяє оновлювати базу стандартів без повного перепроектування програмного забезпечення.

Ще один критичний аспект – забезпечення простоти та зручності користування для лаборантів. Часто користувачі лабораторних систем не мають спеціальної ІТ-підготовки, тому інтерфейс має бути максимально

інтуїтивним, із чітко структурованими формами введення, автоматичними підказками та зменшеним числом обов'язкових дій. Водночас необхідно зберегти достатню функціональність: можливість додавати коментарі, прикріплювати фото зразків, формувати звіти у різних форматах.

Серйозним викликом також є інтеграція з іншими підсистемами сирзаводу. Лабораторна система не повинна існувати ізольовано – результати її аналізів повинні передаватися у виробничу підсистему, щоб у разі виявлення відхилень можна було миттєво змінити параметри технологічного процесу. Так само важлива інтеграція з аналітичним модулем, у якому формуються загальні звіти по якості продукції, динаміці відхилень, роботі персоналу лабораторії.

Варто враховувати й питання збереження та безпеки даних. Результати лабораторних досліджень належать до критично важливої інформації для підприємства, тому необхідно реалізувати механізми резервного копіювання, контролю доступу та аудиту змін. У разі втрати чи підміни таких даних можуть виникнути не лише фінансові збитки, але й репутаційні ризики, особливо у разі виявлення невідповідної якості продукції вже після її реалізації.

Також важливо враховувати можливість масштабування підсистеми в майбутньому. На момент запуску лабораторія може працювати з базовим набором аналізів, однак із розширенням асортименту продукції, змінами в технологічному процесі або встановленням нового обладнання може з'явитися потреба в додаткових функціях. Тому архітектура підсистеми повинна передбачати резерви для розширення функціоналу, підключення нових модулів та оновлення алгоритмів перевірки.

У підсумку, розробка підсистеми лабораторного контролю якості на сирзаводі потребує не лише програмної реалізації, а й глибокого розуміння виробничого процесу, стандартів харчової безпеки, особливостей роботи персоналу та взаємодії з іншими підсистемами. Успішне впровадження такої підсистеми здатне істотно підвищити загальний рівень якості продукції,

зробити виробництво більш контрольованим і прозорим, а також посилити конкурентні переваги підприємства на ринку.

Інформаційна система сирзаводу являє собою складну структуру, що охоплює усі етапи виробництва, контролю якості, постачання, фінансового обліку, управління персоналом та аналітики. Для більш глибокого розуміння її архітектури та принципів побудови, доцільним є поділ на певні класи за відповідними ознаками. Найбільш логічною є класифікація за трьома напрямками: функціональним призначенням, рівнем автоматизації та об'єктом Автоматизації.

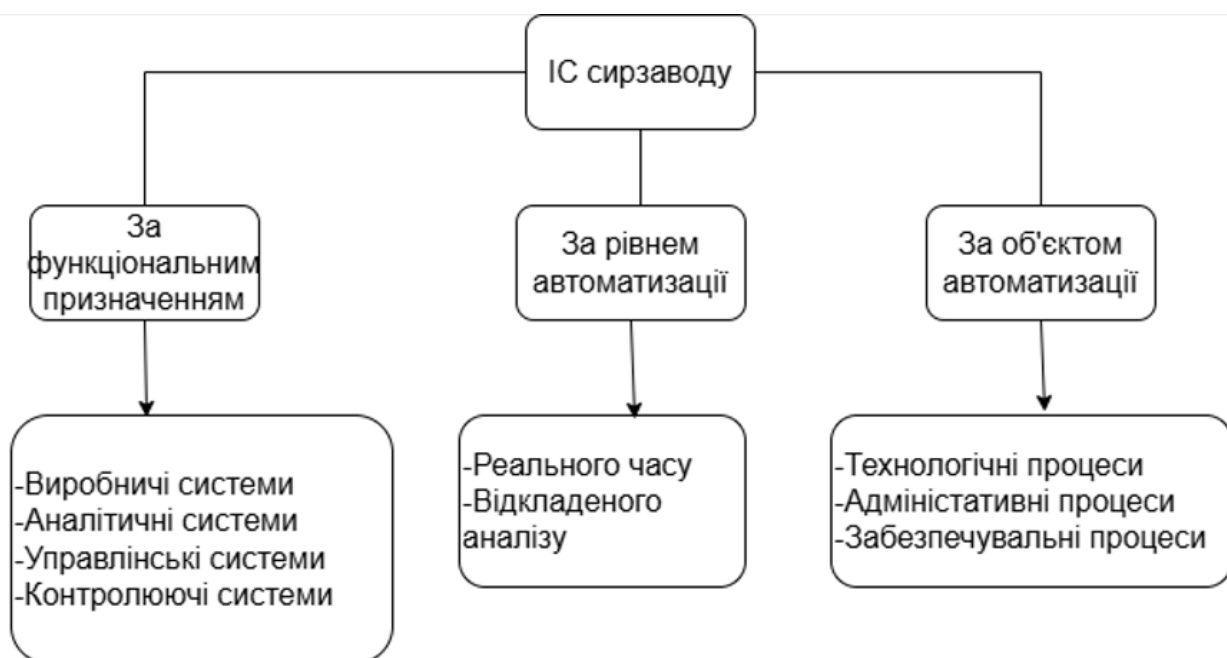


Рисунок 1.1 – Класифікація ІС сирзаводу

Залежно від функціонального призначення, інформаційні системи сирзаводу виконують різні ролі, кожна з яких має своє місце у загальній системі управління підприємством. Виробничі системи забезпечують контроль за основними технологічними процесами на підприємстві. Вони управляють операціями, пов'язаними з переробкою молока, дозріванням сиру, фасуванням та пакуванням продукції. Дані системи інтегровані з обладнанням,

збирають інформацію з датчиків і дозволяють здійснювати оперативне управління параметрами виробництва.

Аналітичні системи зосереджені на обробці великих обсягів даних, які надходять з усіх підрозділів підприємства. Вони забезпечують керівництво підприємства інструментами для аналізу ефективності, прогнозування попиту, оцінки якості продукції та виявлення тенденцій у виробництві. Такі системи часто реалізуються у вигляді панелей управління з інтерактивною візуалізацією показників діяльності.

Управлінські системи зосереджені на забезпеченні стабільного функціонування підприємства на адміністративному рівні. Вони охоплюють планування та облік ресурсів, управління фінансами, кадрами, логістикою та постачанням. Через ці системи здійснюється облік робочого часу працівників, розрахунок заробітної плати, формування податкової та фінансової звітності. Їхня робота безпосередньо впливає на стабільність діяльності сирзаводу.

Контролюючі системи, які тісно пов'язані з лабораторними процесами, виконують завдання перевірки якості сировини та готової продукції. Вони дозволяють оперативно виявляти відхилення від норм, здійснювати автоматичну реєстрацію результатів аналізів і формувати протоколи випробувань. Завдяки цим системам підприємство може гарантувати безпечність і відповідність своєї продукції санітарним та технологічним стандартам.

Окремим напрямом класифікації є поділ систем за рівнем автоматизації. Частина інформаційних систем сирзаводу працює у режимі реального часу – насамперед це виробничі та частково контролюючі підсистеми, які реагують на зміну параметрів практично миттєво. Їхня робота критична для підтримання стабільного технологічного процесу. Інша частина систем функціонує у форматі відкладеного аналізу. До таких відносяться, зокрема, аналітичні системи, що узагальнюють великі масиви даних і формують висновки, які використовуються у стратегічному плануванні.

Ще одним критерієм поділу є об'єкт автоматизації. У цьому розрізі системи можна умовно розділити на ті, що обслуговують технологічні процеси, адміністративну діяльність та забезпечувальні функції. До першої групи належать системи, що забезпечують контроль та керування виробничими потужностями. До другої – модулі управління персоналом, фінансами, бухгалтерським та кадровим обліком. Третю групу становлять допоміжні інформаційні рішення, зокрема ті, що обслуговують лабораторії, IT-інфраструктуру, канали зв'язку та резервне копіювання.

Таким чином, класифікація інформаційних систем сирзаводу дозволяє впорядкувати і структурувати складну сукупність технічних і програмних рішень, що забезпечують ефективну діяльність підприємства. Вона є основою для проектування, вдосконалення та інтеграції нових компонентів у вже існуючу архітектуру цифрового управління.

Інформаційна система сирзаводу складається з кількох взаємопов'язаних підсистем, кожна з яких виконує певні функції для забезпечення ефективної роботи підприємства.

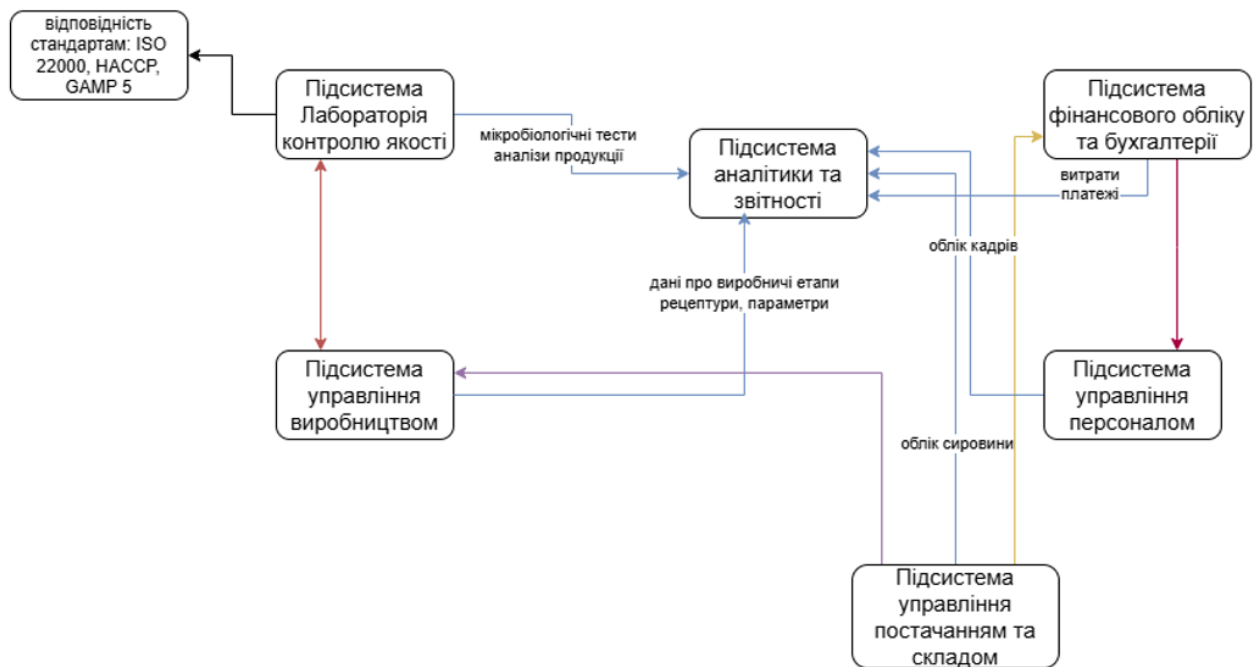


Рисунок 1.2 – Схема функціональної структури ІС сирзаводу

Підсистема управління виробництвом відповідає за контроль усіх технологічних процесів на заводі. Вона забезпечує моніторинг параметрів під час виробництва сиру, таких як температура, кислотність, рівень вологості та інші важливі показники. Автоматизація виробничих процесів дозволяє мінімізувати втручання людини, що знижує ризик помилок і підвищує стабільність якості продукції. Завдяки інтеграції з іншими підсистемами, особливо з лабораторною підсистемою, система управління виробництвом може вносити коригування в реальному часі, наприклад, змінювати параметри ферментації чи дозування компонентів, якщо якість сировини вимагає таких змін.

Особливість даної підсистеми у вигляді безперервного моніторингу всіх технологічних процесів, що дозволяє оперативно реагувати на зміни та коригувати параметри виготовлення продукції. Вона інтегрована з сенсорами, що збирають інформацію про температуру, вологість та інші критичні показники, що впливають на якість сиру. Особливістю цієї підсистеми є також можливість автоматичного внесення змін до рецептури залежно від якості отриманої сировини, що дозволяє мінімізувати відходи та стабілізувати якість кінцевого продукту.

Лабораторна підсистема контролю якості виконує аналіз якості сировини, проміжних продуктів і готової продукції. Перед початком виробництва молоко проходить перевірку на відповідність заданим параметрам, таким як жирність, білковий склад, рівень кислотності та наявність сторонніх домішок. Готовий сир також піддається аналізу на відповідність нормам за фізико-хімічними характеристиками та мікробіологічними показниками. Лабораторна система використовує автоматизовані методи тестування, що дозволяє отримувати швидкі та точні результати. Вона взаємодіє з іншими підсистемами, передаючи результати перевірок в систему управління виробництвом, що дає змогу оперативно коригувати процеси для досягнення найкращих результатів.

Особливістю цієї підсистеми є можливість прогнозування потенційних відхилень у якості продукції ще на етапі її виготовлення, що дозволяє вчасно коригувати процеси й уникати дефектів. Крім того, всі результати аналізів автоматично зберігаються в цифровій базі даних, що спрощує аудит і забезпечує прозорість виробничих процесів.

Підсистема управління поставаннями та складом забезпечує безперервний контроль за запасами сировини, допоміжних матеріалів, обладнання та готової продукції. Вона веде точний облік залишків, контролює терміни придатності продукції та визначає потребу в нових поставаннях. Автоматизоване управління складом дозволяє відстежувати рух товарів, що мінімізує ризики втрат або псування продукції. Логістична частина підсистеми відповідає за оптимізацію поставок і розподіл ресурсів, що зменшує витрати та забезпечує стабільність виробничих процесів.

Особливість даної підсистеми у вигляді повної автоматизації обліку запасів і руху сировини та готової продукції. Вона використовує Radio-Frequency Identification (RFID)-маркери для відстеження кожної партії товарів, що дозволяє контролювати їхнє місцезнаходження та терміни придатності. Завдяки прогнозним алгоритмам підсистема може автоматично формувати заявки на закупівлю сировини, щоб уникнути перебоїв у виробництві. Окрім цього, вона забезпечує оптимізацію розміщення продукції на складах, що знижує витрати на логістику і скорочує час виконання замовлень.

Підсистема фінансового обліку та бухгалтерії виконує розрахунок витрат і прибутків підприємства. Вона дозволяє аналізувати собівартість продукції, вести контроль витрат на сировину, електроенергію, транспортування та інші статті бюджету. Крім того, підсистема автоматизує облік заробітної плати, податкових платежів та інших фінансових операцій. Завдяки цій системі керівництво підприємства отримує доступ до детальних фінансових звітів і може приймати рішення на основі актуальних економічних показників.

Особливість даної підсистеми у вигляді автоматичного аналізу витрат і доходів підприємства в режимі реального часу. Вона інтегрована з виробничими та постачальними процесами, що дозволяє точніше розраховувати собівартість продукції та прогнозувати фінансові ризики. Завдяки використанню інструментів бізнес-аналітики система допомагає керівництву швидко приймати стратегічні рішення на основі актуальних фінансових показників. Також вона забезпечує інтеграцію з банківськими сервісами та податковими службами, що спрощує проведення платежів і звітності.

Підсистема управління персоналом містить інформацію про співробітників, їхні посади, робочий графік та продуктивність. Вона автоматизує кадровий облік, спрощує ведення документації та дозволяє контролювати робочий час працівників. Біометричні системи входу та електронні перепустки дозволяють фіксувати фактичний час перебування працівників на підприємстві. Також підсистема може містити інструменти для планування навчання та підвищення кваліфікації персоналу, що сприяє зростанню професійного рівня співробітників та покращенню ефективності виробничих процесів.

Особливою функцією є можливість організації дистанційного навчання та атестації працівників без відриву від виробничого процесу. Також система автоматично нараховує зарплати з урахуванням понаднормової роботи, премій та штрафів, що зменшує навантаження на бухгалтерію та знижує ризик помилок у розрахунках.

Підсистема аналітики та звітності збирає дані з усіх інших підсистем і формує комплексні аналітичні звіти. Вона дозволяє керівництву отримувати актуальну інформацію про стан виробництва, якість продукції, рівень запасів та фінансові показники. Інтерактивні панелі керування дають змогу відстежувати ключові показники ефективності підприємства в реальному часі. Завдяки прогнозуванню ризиків та аналізу тенденцій система може допомагати у стратегічному плануванні та ухваленні управлінських рішень.

Особливість даної підсистеми у вигляді використання штучного інтелекту для аналізу великих обсягів даних і формування прогнозів. Вона дозволяє підприємству оцінювати ефективність кожного виробничого циклу, виявляти слабкі місця в процесах і пропонувати шляхи їхньої оптимізації. Інтерактивні аналітичні панелі забезпечують зручний доступ до ключових показників ефективності, а автоматизоване формування звітності скорочує час на підготовку документів і підвищує точність розрахунків.

Комплексна взаємодія всіх підсистем забезпечує безперебійну роботу сирзаводу, підвищує якість продукції та ефективність використання ресурсів.

1.3 Аналіз стандартів харчової галузі

У процесі розробки інформаційних систем для харчової промисловості, особливо таких критично важливих компонентів, як підсистема лабораторного контролю якості, необхідно враховувати відповідність галузевим стандартам. Ці стандарти визначають вимоги до безпеки, стабільності, контролю, простежуваності та відповідності продукції, а отже, напряду впливають на вимоги до функціональності й якості програмного забезпечення. Забезпечення відповідності таким нормам має бути закладене ще на етапі вибору методів забезпечення якості IT-проєкту.

Один із ключових міжнародних стандартів – International Organization for Standardization 9001 (ISO 9001), який регламентує систему управління якістю. Він не є специфічним для харчової галузі, але широко застосовується у створенні IT-продуктів. Його важливість полягає в тому, що він встановлює загальні принципи організації процесів розробки, тестування, впровадження та покращення програмного забезпечення. У проєкті лабораторної підсистеми цей стандарт дозволяє формалізувати контроль якості на всіх етапах життєвого циклу: від збирання вимог до супроводу.

Для харчової галузі критичним є дотримання стандарту ISO 22000, а також принципів Hazard Analysis and Critical Control Points (НАССР), які орієнтовані на забезпечення безпеки харчових продуктів через контроль у критичних точках. У контексті ІТ це означає, що система повинна дозволяти виявляти, фіксувати та аналізувати відхилення, вести журнал змін, зберігати результати аналізів, автоматично повідомляти про критичні значення параметрів. Програмне забезпечення має бути здатне підтримувати повну простежуваність, тобто мати функції логування, звітності та відтворення історії процесів.

Ще одним важливим галузевим стандартом є Good Automated Manufacturing Practice (GAMP 5), який застосовується до автоматизованих систем у регульованих виробничих сферах, включаючи харчову та фармацевтичну промисловість. Цей підхід передбачає обов'язкову валідацію програмного забезпечення, тобто підтвердження його відповідності вимогам замовника та технічному завданню. GAMP 5 вимагає ретельної документації, контрольованого управління змінами, чіткої структури тестування, особливо якщо система взаємодіє з обладнанням або впливає на якість кінцевого продукту. Це особливо актуально для лабораторної підсистеми, де результати аналізів напряму впливають на рішення щодо виробництва.

Окрему увагу слід приділити стандарту International Electrotechnical Commission (IEC) ISO/IEC 25010, який визначає основні характеристики якості програмного забезпечення, такі як функціональна відповідність, надійність, зручність використання, безпека, підтримуваність тощо. Цей стандарт є базовим для оцінки якості будь-якого ПЗ, зокрема й у проєктах, де необхідна висока точність обробки даних, стійкість до збоїв і захищеність інформації.

Забезпечення інформаційної безпеки, зокрема захисту результатів лабораторних аналізів, персональних даних працівників, фінансової інформації, базується на принципах ISO/IEC 27001 – міжнародного стандарту для систем управління інформаційною безпекою. Програмне забезпечення

(ПЗ) має забезпечувати контроль доступу, шифрування, аудит дій користувачів, захист від несанкціонованого втручання.

При виборі методу забезпечення якості ІТ-проєкту важливо враховувати:

- чи підтримує обраний метод валідацію ПЗ згідно GAMP 5;
- чи дозволяє він створити систему простежуваності згідно ISO 22000 / HACCP;
- чи забезпечує він відповідність вимогам якості програмного забезпечення ISO/IEC 25010;
- чи включає заходи інформаційної безпеки відповідно до ISO/IEC 27001.

Таким чином, відповідність галузевим стандартам повинна розглядатися як один із критеріїв вибору методів забезпечення якості, а не лише як завершальний етап перевірки.

1.4 Аналіз існуючих методологій управління ІТ-проєктами в контексті забезпечення якості

Однією з важливих стадій життєвого циклу ІТ-проєкту є забезпечення якості, що супроводжує всі основні фази: від ініціації та планування до розробки, впровадження й супроводу системи. Забезпечення якості програмного забезпечення є критичним аспектом у будь-якому проєкті, а особливо – у галузях, де системи впливають на безпеку, точність і стабільність виробничих процесів. Саме такою є сфера харчової промисловості, до якої належить і система, що досліджується в межах даного проєкту – інформаційна система сирзаводу, зокрема її підсистема «Лабораторія контролю якості».

Проєкти з розробки інформаційних систем для підприємств харчової галузі мають ряд характерних особливостей. Насамперед, вони передбачають роботу з фізичними об'єктами – обладнанням, датчиками, лабораторними

приладами, які мають бути точно інтегровані з програмним забезпеченням. Крім того, ці системи повинні забезпечувати повну простежуваність і зберігання результатів аналізів, відповідність санітарним нормам, вимогам до безпеки харчових продуктів і технічним стандартам. Робота таких систем не допускає помилок у розрахунках, втрати інформації чи нестабільності функціонування, оскільки кожна похибка може мати серйозні виробничі й репутаційні наслідки. Водночас, ці системи часто функціонують у динамічному середовищі, де вимоги змінюються, а функціонал розширюється, що вимагає від проектних команд поєднання стабільності з гнучкістю.

У межах дослідження було розглянуто три підходи – Agile, DevOps і ISTQB – з погляду того, як у кожному з них організовано забезпечення якості, які особливості притаманні кожному етапу цього процесу, а також наскільки ці підходи підходять для використання у випадку проекту розробки лабораторної підсистеми сирзаводу.

У методології Agile якість не розглядається як окрема фаза, а є наскрізним процесом, який супроводжує кожен спринт. Упродовж коротких ітерацій команда розробляє інкремент продукту, що містить нову функцію, яка одразу тестується. Забезпечення якості тут реалізується за рахунок автоматизованого юніт-тестування, тестів прийнятності, рев'ю коду, тісної взаємодії з замовником і постійного зворотного зв'язку. Це дозволяє оперативно виявляти помилки, адаптувати функціонал до змін і скорочувати час виходу на ринок [6]. Проте саме ця гнучкість може виявитися недоліком у контексті лабораторної системи, яка потребує чіткого документування результатів, перевірки відповідності встановленим нормативам, формалізованої системи звітності та регламентованих процедур тестування. В Agile ці компоненти не є обов'язковими, що знижує прозорість і складність відтворення результатів, необхідних для сертифікації чи аудиту.

DevOps як підхід до організації життєвого циклу програмного забезпечення реалізує забезпечення якості через автоматизацію процесів: безперервну інтеграцію, доставку, розгортання, тестування та моніторинг.

Якість тут забезпечується шляхом постійного контролю змін у кодї, негайної перевірки його працездатності в інтегрованому середовищі та відстеження поведінки програми у продуктивному середовищі. Це дозволяє оперативно виявляти технічні помилки, запобігати регресіям, забезпечити стабільність при частих оновленнях [7]. Для проєкту лабораторної підсистеми DevOps є корисним у тих аспектах, де потрібна автоматизація перевірок після змін, швидке оновлення модулів і надійна доставка. Проте, DevOps орієнтований передусім на технічну якість – працездатність, швидкодію, безпеку, але не враховує специфіку бізнес-вимог, наприклад, точність алгоритмів лабораторного аналізу чи дотримання стандартів звітності.

ISTQB – це підхід, який формалізує процес забезпечення якості через системне тестування. Він передбачає чітку структуру: аналіз вимог, планування тестування, проєктування тест-кейсів, виконання тестів, фіксацію результатів і завершення. Тут увага зосереджена не на процесі розробки, а на перевірці того, наскільки кінцевий продукт відповідає вимогам, які до нього висувалися. Це забезпечує високу простежуваність, формалізацію та відповідність регламентам – саме ті параметри, що є критично важливими для лабораторної підсистеми. ISTQB дозволяє забезпечити доказову базу якості продукту, зберегти контрольні точки, звіти, логіку перевірок, що є основою довіри до системи з боку регуляторів або внутрішніх аудитів [8]. Недоліком є складність і тривалість підготовки тестової документації, потреба у кваліфікованих спеціалістах і менша адаптивність до частих змін у проєкті.

З урахуванням вищеописаного можна зробити висновок, що розробка підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу вимагає комплексного підходу. Жодна методологія окремо не здатна повною мірою покрити всі потреби такого проєкту. Agile може бути ефективним у фазах взаємодії з користувачами, розробки нового функціоналу, ітеративного вдосконалення системи. DevOps доцільно застосовувати для автоматизації перевірок, моніторингу та доставки змін. ISTQB є незамінним для формалізованого підтвердження відповідності підсистеми вимогам якості.

Саме поєднання цих підходів дозволяє ефективно реалізувати забезпечення якості, враховуючи особливості ІТ-проєкту, вимоги виробництва та специфіку підсистеми, яка не допускає помилок, потребує прозорості й здатності до сертифікації.

В таблиці 1.1 порівнюються етапи розробки ІТ-проєкту

Таблиця 1.1 - Порівняння етапів розробки ІТ-проєкту

Методологія	Основні етапи розробки	Роль контролю якості	Особливості контролю якості
Agile	-Збір вимог; -планування спринту; -розробка функціоналу; -тестування; -демонстрація результатів.	Контроль якості здійснюється безперервно. Інтегрований у кожен етап розробки. Тестувальники працюють разом із розробниками та кінцевими користувачами.	Гнучке тестування на всіх етапах розробки. Юзабіліті-тестування та приймальне тестування виконуються в кожному спринті. Використовуються швидкі цикли перевірок.
DevOps	-Розробка; -безперервна інтеграція (CI/CD); -тестування в реальному часі; -моніторинг і підтримка.	Контроль якості інтегрується безпосередньо у процес розробки завдяки автоматизованому тестуванню та безперервному моніторингу роботи системи.	Тестування відбувається на кожному етапі розробки. Використання контейнеризації для перевірки стабільності продукту в різних середовищах

Кінець таблиці 1.1

Методологія	Основні етапи розробки	Роль контролю якості	Особливості контролю якості
ISTQB	- Аналіз вимог до тестування; - планування тестування; - проєктування тестів; - виконання тестування; - аналіз результатів.	Контроль якості є процесом, який проходить незалежно від розробки. Використовується суворая система тестування, яка охоплює всі рівні розробки.	Акцент на формалізованому підході до тестування, що включає автоматизоване та ручне тестування. Чітка структура звітності та аналізу помилок.

Схожість підходів до забезпечення контролю якості в розглядаємих методологіях:

Всі три етапи методології застосовують тестування на різних етапах життєвого циклу розробки.

Agile та DevOps інтегрують тестування в процес розробки, тоді як ISTQB розглядає тестування як окремий етап.

DevOps та ISTQB активно використовують автоматизоване тестування, тоді як Agile допускає більше ручного тестування з боку кінцевих користувачів.

ISTQB має сувору систему сертифікації для тестувальників, тоді як Agile та DevOps більше орієнтовані на крос-функціональність команд.

Відмінність підходів до забезпечення контролю якості в розглядаємих методологіях:

ISTQB дотримується суворих стандартів тестування, що підходять для критично важливих IT-рішень.

DevOps робить акцент на автоматизації та безперервному тестуванні в

умовах реального використання.

Agile орієнтований на швидкий зворотний зв'язок від кінцевого користувача та гнучке тестування в рамках кожного спринту.

1.5 Постановка задачі дослідження

Об'єктом дослідження є процес забезпечення якості в ІТ-проєктах, що реалізуються в галузі харчової промисловості, зокрема – під час створення функціональних підсистем у складі комплексних інформаційних систем.

Предметом дослідження виступають методи забезпечення якості програмного забезпечення на всіх етапах життєвого циклу ІТ-проєкту, що стосується розробки підсистеми «Лабораторія контролю якості» сирзаводу.

Метою є дослідження методів забезпечення якості ІТ-проєкту та розробка методу для покращення ефективності контролю якості підсистеми «Лабораторія контролю якості» для інформаційної системи сирзаводу.

Саме тому дослідження методів забезпечення якості ІТ-проєктів є актуальним завданням, що має як теоретичне, так і практичне значення.

У рамках дослідження поставлено такі завдання:

- охарактеризувати особливості інформаційних систем у сфері харчової промисловості та визначити роль процесу забезпечення якості на кожному з його етапів з урахуванням специфіки функціонування підсистеми лабораторного контролю якості;

- виділити специфічні вимоги до підсистем контролю якості;

- оцінити ефективність методів забезпечення якості у процесі розробки та впровадження підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу;

- визначити критерії порівняння підходів забезпечення якості, релевантні до поставлених завдань: рівень формалізації, відповідність стандартам

харчової галузі, гнучкість, автоматизованість, масштабованість, прозорість результатів;

– здійснити порівняльний аналіз підходів забезпечення якості та обґрунтувати доцільність застосування того чи іншого підходу (чи їх комбінації) саме в проєкті розробки підсистеми «Лабораторія контролю якості» ІС сирзаводу.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ІТ-ПРОЄКТУ З РОЗРОБКИ ПІДСИСТЕМИ «ЛАБОРАТОРІЯ КОНТРОЛЮ ЯКОСТІ» ІС СИРЗАВОДУ

2.1 Формування проблем та специфічних вимог до комбінованого методу з розробки підсистеми «Лабораторія контролю якості»

Розробка підсистеми «Лабораторія контролю якості» у складі комплексної інформаційної системи сирзаводу є ІТ-проєктом із підвищеним рівнем відповідальності. Вона безпосередньо впливає на якість кінцевої продукції, а отже, і на репутацію та безпеку підприємства. Управління таким проєктом має будуватися на чіткому дотриманні принципів якості, адаптивності, прозорості та інтегрованості з виробничими процесами.

Однією з основних проблем є необхідність врахування різноманітних вимог. З одного боку – це стандарти якості ISO 22000 та HACCP для пропрограманого забезпечення у сфері харчового виробництва, як складової критичної інфраструктури. З іншого боку – потреби у гнучкості для пристосування до змін у процесах або обладнанні лабораторії. Точна та швидка обробка лабораторних даних також ставить перед системою високі вимоги щодо надійності кожного її компонента [9].

Ще одна проблема полягає у складності координаційних процесів між технічними, організаційними та управлінськими аспектами проєкту. Розробка повинна бути інтегрованою з виробничими процесами, мати змогу швидко адаптуватися до нового обладнання, зберігати повний журнал лабораторних досліджень та бути зручною для персоналу. Це потребуватиме різноманітних підходів у менеджменті якості.

Вимоги:

– галузева специфіка: оскільки проєкт реалізується в харчовій промисловості, управлінський процес повинен враховувати суворі вимоги до якості, простежуваності та безпеки, встановлені стандартами ISO 22000,

НАССР, а також GAMP 5, який регламентує створення автоматизованих систем у критичних сферах [10]. Це означає, що всі рішення, прийняті в межах проєкту, мають бути прозорими, задокументованими та верифікованими;

– інтеграція з виробництвом: розробка підсистеми не може відбуватись ізольовано від технологічного середовища заводу. Тому управління проєктом повинно передбачати постійне погодження технічних рішень із фахівцями лабораторії, інженерами, технологами, а також із фахівцями з ІТ-безпеки. Особливу увагу необхідно приділяти сумісності програмного забезпечення з лабораторними приладами, базами даних та загальною інформаційною інфраструктурою заводу;

– забезпечення якості на всіх етапах: якість не створюється на фінальному етапі тестування – вона закладається ще в момент формування вимог, проєктування, написання коду та інтеграції. Тому важливо вже на етапі ініціації проєкту впроваджувати контрольні точки для перевірки відповідності вимогам, оцінки ризиків і фіксації очікувань замовника. Управління має охоплювати планування тестування, вибір технологій, аналіз навантаження, побудову сценаріїв валідації;

– адаптивність управління: через складність взаємодії різних підсистем сирзаводу та динаміку змін у вимогах користувачів, управлінська модель повинна бути здатною швидко реагувати на нові запити. Це означає, що структура проєкту має підтримувати ітеративність, гнучке планування, короткі цикли оновлень і перевірок, що дозволяє виявляти проблеми на ранніх етапах і мінімізувати витрати на їх виправлення;

– контроль ризиків: проєкт повинен включати системне управління ризиками: ризики несумісності з обладнанням, перевантаження персоналу, втрати даних, кіберзагрози. Для цього необхідно проводити попереднє оцінювання, використовувати практики безпеки DevOps, а також передбачати механізми резервного збереження і відновлення даних;

– вимірюваність ефективності: управління проєктом повинно базуватись на кількісних показниках якості. До таких показників належать: кількість

виявлених дефектів, час на виправлення помилки, відсоток автоматизованих тестів, відгуки користувачів, частота збоїв після оновлення, а також вплив на відхилення параметрів якості сиру після впровадження системи. Це дозволяє не лише управляти проєктом, а й приймати обґрунтовані рішення щодо його подальших етапів;

– навчання та підтримка користувачів: оскільки лабораторія сирзаводу є не ІТ-структурою, до обов'язків управління проєктом входить організація навчання персоналу, створення зрозумілих інструкцій, супровід при впровадженні та збирання зворотного зв'язку для покращень.

Таким чином, ефективне управління ІТ-проєктом із розробки підсистеми «Лабораторія контролю якості» вимагає глибокої інтеграції в галузеві процеси, врахування технологічної взаємодії та застосування методологічно обґрунтованих підходів до якості на всіх етапах проєкту – від аналізу вимог до підтримки готової системи.

2.2 Аналіз ризиків у розробці підсистеми

З метою гарантування надійної, безпечної та ефективної роботи підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу, надзвичайно важливим етапом є проведення розгорнутого аналізу ризиків. Цей процес дозволяє виявити потенційні загрози, які можуть виникнути впродовж усього життєвого циклу системи – від проєктування до впровадження та подальшої експлуатації. Такий підхід базується на принципах НАССР та вимогах стандарту ISO 22000, що адаптовані до ІТ-контексту.

Аналіз розпочинається з побудови детальної карти процесу, в якій описуються всі етапи розробки: формулювання вимог, моделювання, програмування, тестування, інтеграція, розгортання, оновлення та підтримка.

Для кожного етапу визначаються потенційні ризики – технічні (збій ПЗ, некоректна логіка коду, перевантаження сервера), інформаційні (втрата чи пошкодження даних, недоступність лабораторних результатів), організаційні (відсутність зворотного зв'язку, помилки через неправильну взаємодію команд) та безпекові (несанкціонований доступ, маніпуляції з даними).

На основі цього проводиться оцінка ризиків – аналізується ймовірність виникнення кожної небезпеки та потенційна шкода, яку вона може спричинити. Наприклад, помилка в модулі передачі даних з лабораторного обладнання може призвести до спотворення результатів аналізів, що вплине на якість партії сиру. Для подібного ризику серйозність наслідків оцінюється як висока. За допомогою шкали (наприклад, від 1 до 5) для кожного ризику визначаються два параметри – імовірність та критичність. Їхній добуток формує індекс ризику, який дозволяє визначити пріоритетність реагування.

Після цього застосовується дерево рішень НАССР, яке допомагає визначити, чи є етап критичною контрольною точкою (ККТ). Якщо ризик не може бути усунутий на пізнішому етапі, або його наслідки є критичними – впроваджується постійний контроль.

Для кожної ККТ встановлюються критичні межі – тобто чіткі показники, що сигналізують про вихід за межі норми. Це можуть бути:

- максимальний час затримки при деплої (наприклад, не більше 2 хвилин);
- частота збоїв після оновлення (наприклад, не більше 1 на 1000 операцій);
- допустимий рівень дефектів у коді (до 5 на 1000 рядків);
- відхилення параметрів якості продукту (наприклад, $\pm 3\%$ від цільового показника лабораторного аналізу).

Крім того, для кожної контрольної точки розробляються процедури моніторингу: хто, коли і за допомогою яких інструментів буде здійснювати перевірку. У проекті лабораторної підсистеми це можуть бути автоматизовані скрипти перевірки логів, періодичне виконання тест-кейсів, Continuous Integration, Continuous Delivery (CI/CD)-пайплайни з аналітикою або ручний аудит журналів помилок.

Якщо відбувається порушення критичних меж, активуються заздалегідь визначені коригувальні дії. Наприклад, у разі перевищення допустимої кількості дефектів у кодї – команда зобов'язана провести позапланове тестування, відкотити останнє оновлення, або призупинити автоматичне розгортання. У випадку втрати лабораторних даних – активується резервне копіювання.

Уся інформація про ризики, контрольні точки, відхилення та вжиті дії фіксується у спеціальній документації. Це забезпечує повну прозорість процесу управління якістю, дозволяє проводити ретроспективний аналіз та вдосконалювати метод управління надалі.

Таким чином, аналіз ризиків у розробці підсистеми «Лабораторія контролю якості» дозволяє не лише виявити потенційні загрози до моменту їх реалізації, але й створити ефективний механізм реагування та запобігання. Це є важливою умовою для досягнення високої якості системи, відповідності її функціональності нормативам і забезпечення надійного контролю за якістю продукції на сирзаводі.

2.3 Визначення контрольних точок

Контрольні точки – це ключові моменти у процесі розробки, на яких здійснюється ретельний контроль якості з метою виявлення дефектів, порушень, втрат даних або ризиків для безпеки ще до того, як вони почнуть впливати на функціонування системи. Для проєкту у харчовій промисловості, де критично важливими є точність, безпека та відповідність стандартам, впровадження таких точок набуває особливого значення.

Однією з найперших контрольних точок є формалізація вимог до підсистеми. Важливо переконатися, що вимоги користувачів, нормативні стандарти, функціональні потреби лабораторії та технологічні обмеження

чітко задокументовані, узгоджені між усіма учасниками й не містять суперечностей. Помилки на цьому етапі можуть призвести до побудови системи, яка не виконує своє призначення або не відповідає стандартам харчової галузі (наприклад, ISO 22000).

Далі йде аналіз ризиків. Кожен етап життєвого циклу – від проєктування до впровадження – аналізується з точки зору технічної складності, можливих загроз безпеці, збоїв у роботі, втрати даних. Наприклад, під час інтеграції з лабораторним обладнанням може виникнути несумісність протоколів, а при оновленні бази даних (БД) – втрата історичних даних. Оцінюється не лише ймовірність виникнення таких інцидентів, а й масштаб їх наслідків. Для кожного ризику має бути визначено, як його попередити або мінімізувати.

На основі цього формується наступна контрольна точка – перевірка архітектури. Вона охоплює логічну структуру взаємодії модулів, канали обміну даними, захищеність баз даних, можливість масштабування та надійність при високому навантаженні. Також оцінюється можливість подальшої підтримки та оновлення системи без ризику порушення її цілісності.

Важливою точкою контролю є проведення код-рев'ю – систематичний перегляд коду досвідченими розробниками або технічними експертами. Це не просто перевірка на синтаксичні помилки. Аналізується структура коду, його читабельність, дотримання стилістичних правил, відсутність дублікатів, правильне використання алгоритмів, а також – дотримання принципів безпеки при роботі з даними.

Після завершення окремих частин розробки впроваджується автоматизоване тестування. Тут застосовуються кілька типів тестів:

- юніт-тести – перевіряють окремі функції та методи;
- інтеграційні тести – аналізують, чи правильно взаємодіють між собою модулі;
- регресійні тести – перевіряють, що нові зміни не порушили вже працюючий функціонал;

– тести безпеки – шукають вразливості до Structured Query Language (SQL)-ін'єкцій, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) тощо.

Окремою контрольною точкою є CI/CD-процеси, тобто безперервна інтеграція та розгортання. На цьому рівні контролюється стабільність збирання програмного продукту, успішність автоматичних тестів, відповідність збірки очікуваній функціональності, а також середній час, який проходить від внесення змін до появи їх у тестовому середовищі. Невдача будь-якого етапу CI/CD свідчить про необхідність повернення до попереднього етапу.

Контроль якості релізу передбачає комплексну перевірку готовності системи до впровадження: тестову документацію, наявність користувацьких інструкцій, звіти про проходження тестів, акти перевірки з боку лабораторного персоналу. Це – фінальна перевірка відповідності всім вимогам перед запуском.

Моніторинг у продакшн-середовищі – ще одна важлива точка. Після впровадження необхідно відстежувати поведінку системи: швидкість реакції, кількість помилок, стабільність обробки запитів, збої у взаємодії з обладнанням, надходження некоректних даних тощо. Всі ці параметри збираються в реальному часі, а порушення заздалегідь встановлених порогових значень слугує сигналом для негайної реакції.

Важливою також є перевірка резервного копіювання. Потрібно не лише переконатися, що резервні копії створюються, але й регулярно тестувати процес їх відновлення в контрольованому середовищі. Лише це дозволяє гарантувати, що у випадку збою дані дійсно можна буде відновити.

Контроль змін у вимогах – ще одна точка, яку не слід ігнорувати. Якщо під час реалізації надходять нові вимоги або змінюється функціонал, вони повинні бути чітко задокументовані, проаналізовані на вплив, та інтегровані в тест-кейси, архітектуру і технічну документацію.

Останньою і найважливішою точкою є аудит відповідності стандартам,

насамперед ISO 22000. Цей аудит дозволяє оцінити, чи система повністю відповідає галузевим вимогам: безпека продукту, простежуваність дій, надійність обробки, звітність, збереження та шифрування даних. Результати аудиту можуть використовуватись як аргумент для сертифікації системи.

Встановлення критичних меж є ключовим кроком у системі управління якістю на основі принципів HACCP, адаптованої до IT-проєкту з розробки підсистеми «Лабораторія контролю якості». Ці межі дозволяють не лише вчасно виявляти відхилення, а й забезпечувати запобігання помилкам до їх появи. Кожна межа чітко визначає допустимі границі параметрів, відхилення від яких сигналізує про потенційну небезпеку для функціональності, безпеки чи надійності інформаційної системи.

Першою контрольованою межею виступає максимально допустима кількість невдалих тестів, встановлена на рівні нуля. Це означає, що будь-яке тестування – юніт-, інтеграційне, функціональне, навантажувальне або регресійне – має завершуватися без збоїв. Навіть один провалений тест є індикатором ризику, особливо коли йдеться про автоматизовані перевірки мікросервісів, які взаємодіють із лабораторним обладнанням або зберігають чутливу інформацію. Цей підхід гарантує, що всі компоненти програмного забезпечення, перед тим як перейти до наступного етапу (наприклад, інтеграції чи впровадження), функціонують відповідно до вимог і стандартів без будь-яких відхилень.

Наступним параметром є час відгуку системи, який не повинен перевищувати 1 секунду. Цей критерій є особливо важливим в умовах реального лабораторного середовища, де швидкість обробки даних напряду впливає на ефективність прийняття рішень. У випадку, якщо підсистема працює повільно або її відповіді затримуються, це може затримати випуск продукції або створити ризик помилкової інтерпретації результатів аналізів. Досягнення цього критерію забезпечується за рахунок оптимізації запитів до бази даних, балансування навантаження, кешування та застосування ефективних алгоритмів пошуку і фільтрації даних.

Особливу увагу приділено параметрам інформаційної безпеки, що контролюються через результати сканування системи на наявність вразливостей. У цьому випадку критична межа передбачає повну відсутність критичних та високих вразливостей, виявлених інструментами статичного або динамічного аналізу коду. Це включає у себе такі загрози, як SQL-ін'єкції, XSS-атаки, відсутність шифрування паролів, неправильна авторизація користувачів тощо. У сфері харчової промисловості, де підсистема обробляє результати лабораторного контролю сировини, готової продукції або критичних параметрів середовища, недопустимо, щоби ці дані могли бути змінені або втрачені внаслідок кібератаки.

До додаткових критичних меж можна також включити:

- частоту збоїв після оновлень – допустимий рівень встановлюється на рівні менше 1%. Якщо після релізу фіксується понад 1% помилок у роботі системи, це є сигналом до перегляду процесу тестування або процедури оновлення (наприклад, CI/CD pipeline).

- рівень відгуків користувачів – якщо система отримує оцінку нижче ніж 4 із 5 за зручність чи точність у внутрішньому опитуванні персоналу, це свідчить про проблеми з інтерфейсом або логікою взаємодії, які необхідно усунути.

- наявність коректного резервного копіювання – система має проходити щоденну автоматичну перевірку створення резервної копії бази даних. Відсутність хоча б одного актуального бекапу впродовж 24 годин вважається порушенням критичної межі.

- відхилення параметрів продукції (наприклад, сиру) – після впровадження системи допускається лише незначне відхилення фізико-хімічних показників. Якщо показники якості продукту погіршуються після запуску системи, це може свідчити про помилки в логіці розрахунків або взаємодії з лабораторним обладнанням.

Для кожної з цих меж мають бути чітко визначені:

- хто відповідальний за контроль (роль або працівник);

- яким саме способом і як часто відбувається перевірка;
- що вважається відхиленням і які дії потрібно виконати у випадку його виявлення.

Таким чином, встановлення критичних меж у процесі розробки та впровадження підсистеми «Лабораторія контролю якості» дозволяє не лише своєчасно ідентифікувати проблеми, а й підтримувати якість роботи системи на стабільно високому рівні, що є критично важливим для безперервного та безпечного функціонування сирзаводу.

Моніторинг і реагування в системі управління якістю є складним і динамічним процесом, що забезпечує своєчасне виявлення та усунення відхилень від встановлених норм у роботі підсистеми «Лабораторія контролю якості». Його роль особливо критична в умовах харчової промисловості, де навіть незначне відхилення в роботі системи може призвести до недостовірних лабораторних результатів, зниження якості продукції або порушення вимог безпеки харчових стандартів, таких як ISO 22000 чи HACCP.

Перш за все, моніторинг передбачає постійне технічне спостереження за станом системи в режимі реального часу. До цього залучаються як внутрішні інструменти CI/CD-пайплайнів, так і зовнішні сервіси моніторингу, як-от Prometheus, Grafana, Sentry чи Zabbix. Усі ці інструменти контролюють ключові параметри: продуктивність системи, навантаження на сервери, затримки у відповіді, стабільність Application Programming Interface (API), кількість помилок під час обробки даних, частоту появи збоїв, статус резервного копіювання, а також результати автоматизованих тестів. При цьому моніторинг охоплює як серверну частину, так і взаємодію користувача з інтерфейсом.

Особливої уваги потребує відстеження результатів критичних точок – тих етапів, які мають найвищий ризик впливу на якість. Наприклад, помилка під час оновлення бази даних може призвести до втрати або спотворення лабораторних даних. А порушення цілісності під час взаємодії з лабораторним обладнанням – до неправильної інтерпретації результатів. Саме тому після

кожної зміни – нового релізу, оновлення чи інтеграції модуля – вмикаються механізми перевірки заздалегідь встановлених критичних меж. Ці межі можуть бути різними: час відповіді сервісу не має перевищувати 1 секунду, частота збоїв після оновлення повинна дорівнювати нулю, а всі критичні тести – проходити успішно.

Якщо ж хоча б один показник виходить за межі допустимого, система в автоматичному режимі запускає процедуру реагування. Найбільш поширений сценарій – блокування деплою або миттєвий відкат до стабільної версії (rollback). Така дія дозволяє уникнути поширення нестабільного програмного забезпечення на всі підсистеми, що працюють у виробничому середовищі.

Крім цього, реагування може включати й сповіщення відповідальних інженерів або DevOps-команд. Вони отримують детальний лог із описом помилки, скріншотами, трасуванням і посиланнями на код, що її викликав. Це дозволяє оперативно локалізувати проблему, внести виправлення та перевірити нове рішення на окремому середовищі до наступного релізу. Якщо проблема носить системний характер, вона вноситься до списку технічного боргу і ставиться у беклог проекту для глибшого аналізу та усунення.

Моніторинг також має функцію звітності – результати спостереження за якістю періодично аналізуються та зберігаються в базі знань. Це дозволяє у довгостроковій перспективі виявляти тренди, формувати прогнози щодо стабільності, а також виявляти ті ділянки системи, які найчастіше піддаються ризикам.

Отже, система моніторингу та реагування є не просто технічним інструментом, а справжнім «органом імунної системи» проекту, який у реальному часі фіксує відхилення, запобігає їх ескалації, забезпечує прозорість для команди і підтримує стабільність та безпечність роботи підсистеми «Лабораторія контролю якості» у виробничому середовищі сирзаводу.

Документування та аудит є важливими складовими системного забезпечення якості ІТ-проекту, зокрема у розробці підсистеми «Лабораторія

контролю якості» для інформаційної системи сирзаводу [11]. Завдяки належному документуванню забезпечується повна простежуваність процесів, рішень та результатів, що дозволяє контролювати якість не лише поточних дій, а й усього життєвого циклу системи.

Усі ключові етапи розробки повинні супроводжуватись веденням звітності. Це стосується результатів автоматизованих і ручних тестів, які показують, чи пройшла система необхідні перевірки за критеріями відповідності функціональним і нефункціональним вимогам. Також важливо фіксувати висновки код-рев'ю – тобто аналізу якості коду, що здійснюється розробниками один одного перед об'єднанням змін до головної гілки. У таких звітах зазначаються знайдені помилки, рекомендації щодо поліпшення архітектури, стилю чи безпеки, а також дії, які були виконані для виправлення. Це дозволяє створити систему взаємного контролю, де жодна частина коду не потрапляє у виробництво без перевірки.

Журнали змін (логування) відіграють не менш важливу роль. Вони фіксують усі зміни у програмному забезпеченні: що саме було змінено, ким, коли і чому. Це забезпечує не лише зручність для розробників, які можуть відстежувати історію змін, але й критичну прозорість для аудиторів та керівництва. Журнали змін часто поєднуються з системами контролю версій (Git, SVN), що дозволяє не просто бачити зміни, а й при потребі повертатися до попереднього стабільного стану.

Окрім цього, у рамках забезпечення якості необхідно періодично проводити аудит процесів розробки, тестування та супроводу. Аудит може здійснюватися як внутрішніми силами команди (самоаудит), так і зовнішніми фахівцями. Його мета – перевірити, наскільки дотримуються встановлених стандартів, регламентів і процедур, чи є відхилення, які впливають на якість продукту, а також визначити, як ці відхилення можна усунути. Під час аудиту оцінюється, наскільки ефективно працюють механізми контролю якості, наскільки повно реалізовані вимоги стандартів (наприклад, ISO 22000), та які ризики залишаються не покритими.

Таким чином, документування та аудит забезпечують надійне середовище для аналізу, перевірки та вдосконалення процесів, дозволяючи не лише фіксувати, а й управляти якістю на всіх рівнях життєвого циклу підсистеми «Лабораторія контролю якості». Це ключовий механізм не лише для оперативного контролю, а й для стратегічного розвитку та безпеки інформаційної системи сирзаводу.

2.4 Розробка комбінованого методу забезпечення якості підсистеми «Лабораторія контролю якості» ІС сирзаводу

У процесі розробки інформаційної підсистеми «Лабораторія контролю якості» виникає потреба у застосуванні ефективного методу управління забезпеченням якості. Зважаючи на специфіку задач та різноманітність вимог до якості програмного забезпечення в харчовій промисловості, було запропоновано комбінований підхід, що базується на поєднанні трьох методологій: Agile, DevOps та ISTQB.

Цей комбінований підхід дозволяє врахувати потребу в адаптивності (Agile), автоматизації розгортання та тестування (DevOps), а також структурованому підході до тестування (ISTQB). На відміну від класичних методів, запропонована модель забезпечує циклічність, зворотний зв'язок і безперервний контроль якості.

Етап 1. Ініціація. В контексті розробки підсистеми «Лабораторія контролю якості» інформаційної системи сирзаводу є фундаментальним і визначальним для подальшого управління якістю проєкту. Його мета – виявити потребу в зміні, сформуванню бачення майбутньої системи та закласти основу для ефективного управління якістю.

Крок 1.1. Виявлення та обґрунтування потреби. На цьому етапі фіксуються реальні проблеми лабораторного контролю на виробництві:

затримки в обробці результатів, ручне введення даних, дублювання, помилки персоналу тощо. Залучаються представники лабораторії для збору початкових зауважень. У межах гнучкого підходу (Agile) формується попереднє бачення продукту (vision), а DevOps-фахівці оцінюють технічну готовність.

Крок 1.2. Аудит регуляторних та галузевих вимог. Вивчаються стандарти та норми, яких має дотримуватись система. До прикладу, ISO 22000, внутрішні протоколи лабораторії, вимоги з боку державного нагляду. На основі цього ISTQB-тестувальники визначають попередні критерії якості, а також специфікації для майбутнього тестування.

Крок 1.3. Формування початкових вимог до системи. На цьому етапі формуються базові вимоги:

- функціональні (наприклад, автоматичне зчитування з приладів, генерація звітів);
- нефункціональні (наприклад, безперебійна робота, зручність інтерфейсу).

Agile-методологія допомагає створити попередній backlog для подальшої деталізації.

Етап 2. Планування. На цьому етапі формалізуються вимоги до системи, визначається архітектура, структура команди, технологічний стек, інструменти тестування та підходи до забезпечення якості. Саме тут комбінований підхід починає діяти повною мірою: Agile відповідає за гнучкість у плануванні, DevOps – за технічну інтеграцію автоматизації, а ISTQB – за структурованість майбутнього тестування.

Крок 2.1. Формування технічного бачення проєкту. На основі зібраних вимог створюється узагальнене технічне бачення. Визначається структура підсистеми: база даних, інтерфейс користувача, модулі обробки даних, механізми зв'язку з лабораторними приладами. Також відразу враховуються питання масштабованості, надійності та інтеграції.

Крок 2.2. Побудова загальної архітектури системи. Архітектура описує взаємодію всіх компонентів підсистеми: серверна частина, модулі збору

даних, аналітика, звітність, доступ користувачів. Створюються діаграми взаємодії, схеми потоків даних. Це критично важливо для подальшого забезпечення цілісності якості.

Крок 2.3. Планування процесів автоматизації. Описується пайплайн автоматизованої перевірки якості: автоматичне збирання, запуск юніт-тестів, інтеграція з Git, деплой у тестове середовище, моніторинг продуктивності. Також визначається стратегія резервного копіювання та логування.

Крок 2.4. Визначення контрольних точок якості. Формуються ключові показники ефективності (наприклад: відсоток автоматизованих тестів, середній час реагування системи, кількість збоїв на тисячу операцій).

Етап 3. Розробка. Цей етап є центральним у реалізації підсистеми. Тут створюється безпосередньо програмний код, реалізується функціональність, формується логіка взаємодії з лабораторними приладами та іншими модулями системи. Комбінований підхід дозволяє одночасно підтримувати гнучкість, забезпечувати безперервну інтеграцію і формалізовану перевірку якості на всіх рівнях.

Крок 3.1. Реалізація модулів та функціональних блоків. Розробка здійснюється відповідно до архітектури, затвердженої на попередньому етапі. Кожен модуль реалізується незалежно, з урахуванням принципів SOLID, щоб забезпечити майбутнє тестування та повторне використання коду.

Крок 3.2. Внутрішній контроль якості (code review). Перед злиттям кожної зміни в основну гілку проводиться перевірка коду іншими розробниками. Це дозволяє виявити помилки, покращити архітектурні рішення та підтримувати єдиний стиль.

Крок 3.3. Написання юніт- та інтеграційних тестів. Для кожного модуля розробляються тести, що перевіряють поведінку окремих функцій (unit tests), а також взаємодію між модулями (integration tests). Це забезпечує виявлення помилок ще до інтеграції у систему.

Крок 3.4. Побудова зворотного зв'язку з тестуванням. Після реалізації функціоналу забезпечується обов'язкове тестування та збір результатів.

Зворотний зв'язок іде не тільки до тестувальників, але й до розробників, архітектора та власника продукту.

Етап 4. Інтеграція. Етап інтеграції передбачає об'єднання окремих модулів, компонентів і сервісів у єдину функціональну систему. Особливу увагу приділяють перевірки взаємодії між підсистемами, коректності обміну даними, а також відсутності збоїв у точках дотику. Комбінований метод дозволяє поєднати гнучкість адаптації (Agile), інструменти автоматизації перевірок (DevOps) та структурованість тестування (ISTQB) на цьому критичному етапі.

Крок 4.1. Інтеграційне тестування. Проводиться серія перевірок, які мають на меті виявити проблеми у взаємодії між різними частинами системи. Створюються і виконуються сценарії інтеграційного тестування, що охоплюють типові, критичні та нетипові ситуації.

Крок 4.2. Автоматизація процесу інтеграції. Інтеграційні дії автоматизуються за допомогою інструментів DevOps. Налаштовується процес безперервної інтеграції (CI), збірки й розгортання змін, а також підключається моніторинг на випадок збоїв.

Крок 4.3. Верифікація відповідності якості. Оцінюється якість реалізованої інтеграції на основі ключових метрик: надійність передачі даних, відсутність збоїв, швидкість обробки інформації. Приймається рішення про перехід до етапу впровадження або повторну доопрацювання.

Етап 5. Тестування. Етап тестування у процесі розробки підсистеми «Лабораторія контролю якості» є критично важливим для забезпечення надійності, точності та стабільної роботи всієї інформаційної системи. Саме на цьому етапі перевіряється, наскільки створене програмне забезпечення відповідає функціональним вимогам, стандартам якості й очікуванням користувачів.

Крок 5.1. Планування тестування. На цьому етапі формується стратегія тестування: визначаються цілі, обсяг, типи тестів, ресурси та відповідальні особи. Складається тестовий план, який охоплює функціональні,

нефункціональні, інтеграційні, регресійні та користувацькі тести.

Крок 5.2. Проведення ручного та автоматизованого тестування. Виконуються тест-кейси вручну або за допомогою автоматизованих інструментів. При DevOps-компоненті – тестування інтегрується в CI/CD пайплайн. Виконуються юніт-тести, інтеграційні, User Interface (UI), безпекові тести та навантажувальні тести на продуктивність.

Крок 5.3. Аналіз результатів тестування. Усі помилки документуються та класифікуються за критичністю. Порівнюються фактичні результати з очікуваними, ведуться звіти про успішні й невдалі перевірки. На основі цього формується перелік дефектів для усунення та оцінюється рівень готовності системи до релізу.

Етап 6. Впровадження. Етап впровадження підсистеми «Лабораторія контролю якості» є завершальною, але надзвичайно відповідальною фазою життєвого циклу IT-проєкту. Мета цього етапу – перенести систему з середовища розробки та тестування до реального виробничого середовища сирзаводу, забезпечивши її стабільну та безпечну роботу. Впровадження має враховувати специфіку харчової промисловості, де критично важливими є надійність, точність обробки даних, відсутність збоїв і відповідність вимогам безпеки. Комбінований підхід дозволяє на цьому етапі одночасно використовувати автоматизовані інструменти DevOps, адаптивність Agile і контрольні механізми ISTQB.

Крок 6.1. Підготовка середовища впровадження. Перед запуском система має бути розгорнута в ізольованому середовищі для остаточної перевірки. Налаштовуються сервери, мережеві компоненти, бази даних, доступи.

Крок 6.2. Проведення фінального тестування на продуктивність. Здійснюється стрес-тестування, перевіряється швидкість реакції, обсяг оброблених даних, стійкість до пікових навантажень. Особливо контролюються критичні контрольні точки.

Крок 6.3. Післявпроваджувальний моніторинг та підтримка. Система

перебуває під посиленним моніторингом. Фіксуються всі збої, помилки, зворотний зв'язок користувачів. У разі виявлення недоліків – оперативне оновлення, виправлення або відкликання версії.

На рисунку 2.1 відображено схему етапів комбінованого методу забезпечення якості підсистеми «Лабораторія контролю якості» ІС сирзаводу.

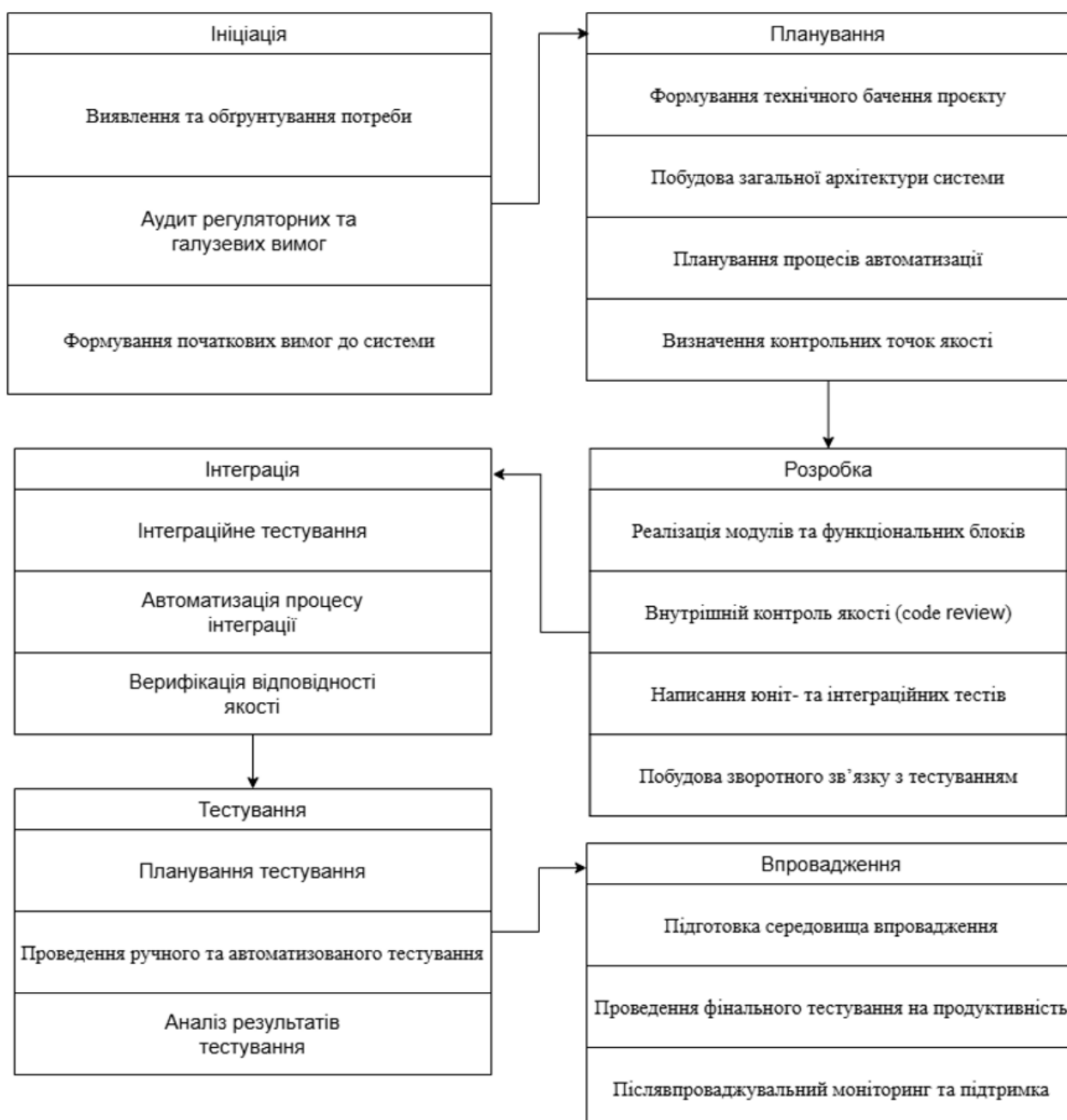


Рисунок 2.1 – Схема етапів комбінованого методу забезпечення якості підсистеми «Лабораторія контролю якості» ІС сирзаводу

У межах впровадженого комбінованого методу забезпечення якості кожен етап розробки підсистеми супроводжується встановленням метрик, які дозволяють об'єктивно оцінити якість виконаних робіт. Для цього використовується набір кількісних індикаторів, що охоплюють різні аспекти функціонування системи. Серед них:

D – кількість виявлених дефектів на 1000 рядків коду;

T – середній час на виправлення помилки;

A – відсоток автоматизованих тестів від загальної кількості тестів;

F – середній бал відгуків користувачів щодо системи (від 1 до 5);

E – частота збоїв після оновлення;

Q – відхилення показників якості готової продукції (наприклад, сиру) до і після впровадження системи.

Такі показники дозволяють виявляти проблемні місця, оперативно реагувати на відхилення та коригувати підхід ще до переходу до наступного етапу розробки. Кожен із показників нормалізується у вигляді індексу якості, що приймає значення від 0 до 1, де 1 означає ідеальний стан. Здійснюється розрахунок ефективності забезпечення якості за такими формулами:

1) Індекс дефектів:

$$IQ1 = 1 - \left(\frac{D}{D_{\max}} \right)$$

де $IQ1$ – індекс дефектів;

D_{\max} – максимальна кількість дефектів;

D – фактична кількість виявлених дефектів на 1000 рядків коду;

1 – ідеальне значення (100% якості);

2) Індекс виправлення:

$$IQ2 = 1 - \left(\frac{T}{T_{\max}} \right)$$

де $IQ2$ – індекс виправлення;

T – середній час на виправлення помилки;

T_{\max} – максимально допустимий час на виправлення, який вважається прийнятним;

1 – ідеальне значення якості;

3) Індекс автоматизації:

$$IQ3 = \frac{A}{100}$$

де $IQ3$ – індекс автоматизації;

A – відсоток автоматизованих тестів від загальної кількості тестів;

100 – значення виражається у %, тому ділимо на 100, щоб отримати нормалізовану оцінку;

4) Індекс користувацького досвіду:

$$IQ4 = \frac{F}{5}$$

де $IQ4$ – індекс користувацького досвіду;

F – середня оцінка користувачів щодо зручності, точності та стабільності роботи системи (5-бальна шкала);

5 – максимальний можливий бал;

5) Індекс стабільності після оновлення:

$$IQ5 = 1 - \left(\frac{E}{E_{\max}} \right)$$

де $IQ5$ – індекс стабільності після оновлення;

E – кількість зафіксованих збоїв після оновлення ПЗ;

E_{\max} – допустиме число збоїв, після якого якість вважається незадовільною;

1 – ідеальне значення якості;

б) Індекс впливу на продукт:

$$IQ6 = 1 - \left(\frac{Q}{Q_{\max}} \right)$$

де $IQ6$ – індекс впливу на продукт;

Q – середнє відхилення параметрів якості продукту після впровадження підсистеми;

Q_{\max} – максимально допустиме відхилення (в межах стандартів харчової якості);

1 – ідеальне значення.

Загальна формула:

$$IQ = \frac{IQ1 + IQ2 + IQ3 + IQ4 + IQ5 + IQ6}{6}$$

де IQ – середній індекс якості.

Цей середній показник дозволяє приймати рішення про доцільність переходу до наступного етапу. Якщо $IQ \geq 0.85$, вважається, що якість задовільна; в іншому випадку необхідно здійснити повторну перевірку, корекцію.

Таким чином, запропонований підхід забезпечує гнучке, але формалізоване управління якістю, з чіткою системою показників і порогових значень, що підтримують стабільність та відповідність вимогам на кожному етапі життєвого циклу проекту.

3 АПРОБАЦІЯ КОМБІНОВАНОГО МЕТОДУ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ІТ-ПРОЄКТУ

3.1 Опис ІТ-проєкту з розробки підсистеми «Лабораторія контролю якості» ІС сирзаводу

У межах апробації розробленого комбінованого методу забезпечення якості було обрано ІТ-проєкт розробки підсистеми «Лабораторія контролю якості» як ключовий елемент цифрової трансформації процесів якості на сучасному молочному виробництві. Підсистема інтегрується до єдиної інформаційної системи сирзаводу та відповідає за автоматизований збір, аналіз і контроль лабораторних даних, що супроводжують виготовлення сирів.

Місія проєкту – забезпечити прозорий, безперервний і об'єктивний контроль якості молочної продукції на всіх етапах її виробництва, знизити людський фактор у процесах перевірки, мінімізувати ризики відхилень і покращити відповідність вітчизняним та міжнародним стандартам (зокрема, ISO 22000, HACCP).

Цільове призначення проєкту – створити повноцінну ІТ-підсистему, яка автоматизує всі дії, пов'язані з лабораторним контролем: від зчитування результатів вимірювань до формування цифрових звітів та інтеграції з виробничими й обліковими модулями. Вона має забезпечити оперативне виявлення відхилень, простежуваність кожної партії продукції та системне покращення якості сировини й готового продукту.

На момент ініціації більшість дій у лабораторії сирзаводу виконувалися вручну або частково в електронних таблицях, що створювало ризики дублювання інформації, затримок у реагуванні на відхилення, складнощів із формуванням звітності та аналізом динаміки якості. Крім того, окремі модулі лабораторного обладнання не були пов'язані між собою і з основною інформаційною системою підприємства.

Проєкт передбачає створення сучасної автоматизованої системи, яка

включає:

- інтеграцію лабораторного обладнання (вологоміри, рН-метри, спектрофотометри, мікробіологічні аналізатори) з цифровими модулями збору даних;
- автоматичну перевірку значень за критеріями якості;
- формування електронних протоколів перевірок;
- моніторинг у реальному часі на контрольних точках згідно з принципами НАССР;
- збереження історії партій, відстеження сировини та готової продукції;
- взаємодію з обліковою системою;
- контроль відповідності показників установленим критичним межам (наприклад, жирність, вологість, кількість бактеріальних клітин тощо).

Правова та регуляторна основа ІТ-проєкту включає дотримання таких норм:

– закон України «Про основні принципи та вимоги до безпечності та якості харчових продуктів» – визначає базові обов’язки оператора ринку щодо контролю за безпечністю продукції на всіх етапах виробництва, включаючи лабораторні перевірки. Відповідність цьому закону є обов’язковою умовою функціонування сирзаводу;

– впровадження системи НАССР – обов’язкова вимога для харчових підприємств згідно з українським законодавством і нормами. Підсистема «Лабораторія контролю якості» має враховувати логіку аналізу ризиків, ідентифікації критичних точок та контролю за дотриманням меж;

– вимоги ISO 22000 – міжнародний стандарт щодо управління безпечністю харчових продуктів, який охоплює як технологічні процеси, так і документування, простежуваність, відслідковування відхилень. Інформаційна підсистема повинна підтримувати всі етапи контролю відповідно до цього стандарту;

– галузеві накази та інструкції Держпродспоживслужби – регламентують перелік параметрів, які підлягають лабораторному контролю (наприклад,

мікробіологічні показники, жирність, вологість, кислотність тощо), а також вимоги до методів і періодичності аналізів;

– регламент внутрішнього контролю якості підприємства – затверджений сирзаводом документ, що встановлює внутрішні контрольні процедури, допустимі відхилення, переліки вимірюваних характеристик та порядок дій у разі виявлення порушень;

– закон України «Про електронні документи та електронний документообіг» – забезпечує легітимність автоматизованої фіксації результатів лабораторного аналізу та електронного підпису відповідальних осіб у системі;

– закон України «Про захист персональних даних» – оскільки підсистема може містити персоніфіковані результати аналізів або дані відповідальних працівників, необхідно забезпечити їх захист і шифрування відповідно до вимог;

– положення Кодексу законів про працю України та внутрішні інструкції з охорони праці – стосуються правил роботи з лабораторним обладнанням, ведення журналів безпеки та підписання протоколів;

– вимоги до інтеграції з державними інформаційними системами (у разі потреби) – наприклад, для подання звітів або участі в системі простежуваності походження сировини й безпеки готової продукції.

Ресурсна база:

– людські ресурси: IT-команда (розробники, тестувальники, DevOps-фахівці), спеціалісти з якості, представники лабораторії;

– фінансові ресурси: бюджет підприємства, залучені інвестиції або гранти на цифровізацію;

– матеріальні ресурси: лабораторне обладнання, сервери, робочі станції;

– нематеріальні ресурси: доступ до даних лабораторних аналізів, внутрішні протоколи перевірки якості, досвід і знання персоналу.

Класифікація підсистеми «Лабораторія контролю якості»:

– тип системи: інформаційно-аналітична;

- архітектура: модульна, з централізованим зберіганням даних;
 - тип взаємодії: інтегрована – обмін даними з іншими модулями ІС (виробництво, складування, логістика);
 - вид діяльності: виробничий контроль у харчовій промисловості;
 - належність до галузі: агропромисловий сектор, молочна промисловість.
- Проєкт передбачає створення таких модулів:
- модуль збору даних: для автоматичного імпорту результатів аналізів із приладів (рН-метри, жироміри, бактеріологічні аналізатори тощо);
 - аналітичний модуль: для розрахунку відхилень, генерації графіків стабільності, визначення трендів якості;
 - модуль звітності: формування звітів у форматах, що відповідають вимогам контролюючих органів (Держпродспоживслужба, внутрішній аудит);
 - інтерфейс контролю якості: панель керування для технологів та аналітиків;
 - модуль архівації: зберігання історії змін, версій і результатів перевірок.
- Очікувана тривалість реалізації проєкту – 8-10 місяців, з бюджетом до 1.5 млн грн.

Команда включає:

- проєктного менеджера;
- бізнес-аналітика;
- 2 розробників;
- тестувальника;
- DevOps-фахівця;
- експерта з лабораторного контролю;
- представника служби якості сирзаводу.

Станом на початок розробки система контролю якості сирзаводу переважно ґрунтується на ручному введенні результатів і не має єдиної централізованої бази даних. Це ускладнює виявлення аномалій, обмежує можливість швидкого реагування та створює ризики недотримання стандартів.

Розроблена підсистема дозволить подолати ці проблеми, зробивши контроль якості точним, своєчасним і технологічно інтегрованим.

3.2 Окремі задачі у контексті розробки проєкту та причини використання реалізації

На етапі ініціації проєкту було реалізовано низку важливих задач, спрямованих на глибоке розуміння поточних проблем лабораторного контролю якості продукції сирзаводу.

Однією з перших задач стало виявлення проблем, пов'язаних з лабораторним контролем якості. Було встановлено, що ручне введення результатів, відсутність уніфікованих протоколів і недосконалі аналітики створюють суттєві перешкоди для ефективного управління. Саме тому було обґрунтовано доцільність розробки цифрової підсистеми. Її реалізація дозволяє автоматизувати введення, виключити помилки, забезпечити простежуваність і підвищити точність обробки даних.

Подальша задача полягала у формуванні концепції цифрової лабораторії якості. Було вирішено застосувати методологію Agile, оскільки вона забезпечує швидке створення первинного продукту (MVP) з можливістю поступового доопрацювання функціоналу відповідно до змін у вимогах. Це стало критично важливим у виробничому середовищі, де специфіка вимірювань і процедур може змінюватися.

Наступною задачею був аудит нормативних вимог (ISO 22000, HACCP, внутрішні протоколи). Для забезпечення відповідності цим стандартам було прийнято рішення про використання підходу ISTQB. Це дозволило ще на ранньому етапі інтегрувати контрольні точки в логіку валідації, що забезпечує прозорість і документованість результатів.

На завершення було сформовано функціональні й нефункціональні

вимоги. Причиною їх конкретизації стало прагнення уникнути двозначностей у розробці, визначити очікування до продуктивності, доступу, сумісності та підтримки. Це дозволило чітко окреслити рамки системи та підготувати основу для її архітектурного проектування.

Після завершення етапу ініціації, на якому було сформульовано проблематику та загальну концепцію майбутньої підсистеми, проєкт перейшов до фази планування.

Першою задачею на цьому етапі стало формування технічного бачення. У ньому було визначено ключові компоненти майбутньої системи: база даних, інтерфейси, механізми інтеграції з приладами, моніторинг показників, API для взаємодії з ІС підприємства. Таке бачення було реалізоване для того, щоб гарантувати комплексну автоматизацію процесу контролю якості з урахуванням вимог регуляторних органів.

Після цього було побудовано архітектуру підсистеми. Причиною вибору саме модульної структури стало прагнення до масштабованості, гнучкості при зміні окремих компонентів та спрощення інтеграції. Чітке розмежування серверної й клієнтської частини дозволило забезпечити стабільну роботу, ізоляцію обробки даних і зручність для користувача.

Окремою задачею стало планування процесів автоматизації. Це включало підготовку методики оцінки рівня автоматизації, продуктивності та стійкості після оновлень. Було прийнято рішення використовувати кількісні індекси, оскільки вони дозволяють об'єктивно вимірювати ефективність і ухвалювати обґрунтовані рішення про доопрацювання.

Ще однією задачею стало формалізоване визначення контрольних точок якості. Причиною такої реалізації стала потреба у створенні універсальної системи оцінювання, яка дозволяє швидко виявляти порушення, незалежно від джерела помилки – технічного, організаційного або людського.

На етапі розробки основною задачею була реалізація модулів згідно з архітектурою, створеною під час планування. Було прийнято рішення використовувати принципи SOLID для забезпечення чіткого розподілу

відповідальностей між компонентами та підтримки зручного розширення. Це дало змогу зробити систему стабільною, логічно структурованою та придатною до подальшого масштабування.

Наступною задачею став внутрішній контроль якості. Було реалізовано перевірку коду іншими учасниками перед інтеграцією, що дало змогу виявляти порушення стандартів і знижувати ймовірність помилок. Такий підхід обрано через його ефективність у виявленні проблем на ранній стадії, що зменшує витрати на виправлення в майбутньому.

Третьою задачею стало створення unit- та integration-тестів для кожного модуля. Було вирішено автоматизувати тестування, щоб пришвидшити перевірку та забезпечити повторюваність результатів. Автоматизація дозволила зменшити ризик людських помилок і підвищити загальну надійність системи.

Останньою задачею стала організація зворотного зв'язку. Було реалізовано механізм збору відгуків користувачів для визначення рівня зручності, розуміння та стабільності роботи системи з точки зору персоналу лабораторії. Причиною цього стала необхідність отримати незалежну оцінку реалізованих функцій, щоб вчасно виявити функціональні недоліки.

Під час інтеграції першочерговою задачею стало тестування взаємодії модулів. Це дало змогу виявити проблеми на стиках між компонентами, які не могли бути виявлені на рівні модульного тестування. Така перевірка була необхідною для збереження цілісності системи після об'єднання.

Далі було реалізовано автоматизацію інтеграційних процесів за допомогою CI-пайплайну. Це рішення було прийнято для зменшення навантаження на команду, пришвидшення перевірок і забезпечення прозорості при внесенні змін. Впровадження DevOps-практик дозволило відслідковувати стабільність системи на кожному етапі злиття.

Ще однією задачею стала перевірка часу реакції на виявлені помилки. Було обрано метод збору статистики усунення проблем, щоб мати змогу оцінювати оперативність реагування. Причиною цього стала потреба

уникнути затримок, які в умовах виробництва можуть впливати на реальний процес випуску продукції.

Завершальною задачею стала оцінка впливу системи на якість готової продукції. Було проведено верифікацію, чи дотримуються показники якості після впровадження змін. Таке рішення було необхідним для того, щоб забезпечити технологічну обґрунтованість результатів розробки і підтвердити практичну ефективність ІТ-рішення у виробничому процесі.

Етап впровадження завершував цикл розробки, забезпечуючи перенесення підсистеми до робочого середовища та її перевірку в умовах, наближених до реальних.

Першою задачею було налаштування ізолюваного середовища впровадження. Причиною реалізації саме в окремому середовищі стала потреба уникнути ризиків впливу на виробничі процеси до остаточної перевірки системи. Це також дозволило адаптувати конфігурацію мережевої взаємодії з приладами без втручання в основну інфраструктуру.

Другою задачею стало проведення фінального тестування продуктивності системи. Його реалізували для перевірки стійкості підсистеми до пікових навантажень. Такий підхід дозволив виявити можливі критичні затримки чи нестабільність, які могли б проявитись лише під час одночасної роботи з кількох приладів.

Третьою задачею був аналіз поведінки системи після впровадження. Причиною включення цього етапу стала потреба перевірити відповідність фактичної поведінки програмного забезпечення очікуваним параметрам стабільності та сумісності з виробничими умовами.

Завершальною задачею стало впровадження післявпроваджувального моніторингу. Його реалізували для оперативного виявлення та усунення потенційних помилок, що могли залишитись поза увагою під час попередніх етапів. Це рішення дозволило забезпечити контроль якості в реальному часі одразу після запуску.

3.3 Апробація комбінованого методу забезпечення якості

Етап 1 Ініціація.

Крок 1.1. Виявлення та обґрунтування потреби.

На початку етапу було проведено діагностику наявних процесів лабораторного контролю якості молочної продукції. Було встановлено, що ключові проблеми полягають у:

- ручному введенні результатів аналізів, що спричиняє затримки, помилки, дублювання;
- неуніфікованості протоколів оцінки якості, що ускладнює порівняння між різними партіями;
- відсутності системної аналітики та візуалізації динаміки змін;
- незадовільному рівні цифрової взаємодії між лабораторією, виробництвом та відділом контролю.

У зв'язку з цим виникла потреба в побудові цифрової підсистеми, яка б дозволила автоматизувати введення даних, підвищити точність, надати засоби контролю, відстеження та аналітики.

У межах гнучкого підходу (Agile) вже на цьому етапі було сформовано первинне бачення продукту – цифрова лабораторія якості, інтегрована з приладами, здатна збирати дані в режимі реального часу, автоматично генерувати звіти та забезпечувати нормативний контроль.

Крок 1.2. Аудит регуляторних та галузевих вимог.

З огляду на те, що галузь належить до харчової промисловості, якість і безпека продукції регламентується численними стандартами. На цьому кроці були опрацьовані:

- ISO 22000 – управління безпечністю харчових продуктів;
- внутрішні лабораторні протоколи підприємства;
- вимоги Держпродспоживслужби та міжнародних сертифікаційних органів;

- рекомендації НАССР для критичних контрольних точок.

Ці нормативи стали основою для формування майбутніх критеріїв якості, які були закладені у систему як контрольні точки для валідації функціоналу. Визначення цих вимог забезпечувалося за ISTQB, яка дозволила структурувати специфікації майбутнього тестування.

Крок 1.3. Формування початкових вимог до системи.

На основі виявлених потреб і нормативних вимог були сформовані базові вимоги до майбутньої підсистеми:

1) функціональні:

- автоматичне зчитування результатів із приладів;
- генерація лабораторних звітів згідно з шаблонами;
- аналітика з динамікою відхилень;
- інтеграція з існуючою обліковою системою сирзаводу.

2) нефункціональні:

- стабільна робота в режимі 24/7;
- мінімальна затримка обробки даних (<5 сек);
- рольова модель доступу;
- сумісність з міжнародними протоколами передачі лабораторних

даних.

На цьому ж етапі ISTQB-фахівці починають формувати початкову тестову документацію, яка включає сценарії перевірки відповідності вимогам і оцінки якості системи. Перші версії критеріїв тестування використовуються для побудови попередніх метрик, які в подальшому ляжуть в основу формул.

На етапі ініціації ще не виконуються самі розрахунки, проте закладаються основні вихідні параметри, які будуть використані у формулах:

- діапазон допустимих значень D , T , A , F , E , Q ;
- критерії, що визначають ідеальний стан ($IQ = 1.0$);
- фреймворк для оцінки відхилень та дефектів;
- методи збору даних для метрик у подальших фазах.

Етап 2 Планування.

Крок 2.1. Формування технічного бачення проєкту.

На основі зібраних у попередньому етапі вимог було сформовано узагальнене технічне бачення майбутньої підсистеми, що включає:

- структуру бази даних для зберігання результатів контролю якості;
- інтерфейси введення даних та перегляду результатів;
- механізми обміну з лабораторними приладами;
- звітність відповідно до ISO 22000 та HACCP;
- інтеграційні API з ІС сирзаводу;
- засоби моніторингу показників продукції в динаміці.

На цьому кроці вже визначаються параметри, які будуть згодом використовуватись у формулах індекс автоматизації, стабільність після оновлення та вплив на продукт.

Крок 2.2. Побудова загальної архітектури системи.

Описано архітектуру підсистеми, що включає:

- серверну частину (обробка запитів, база даних, API);
- клієнтську частину (UI-інтерфейс для лабораторії);
- модулі: збирання даних, аналітика, генерація звітів;
- схеми потоків даних між компонентами;
- інтеграцію з приладами (через драйвери або файлові обмінники).

На рисунку 3.1 відображено схема загальної архітектури системи.

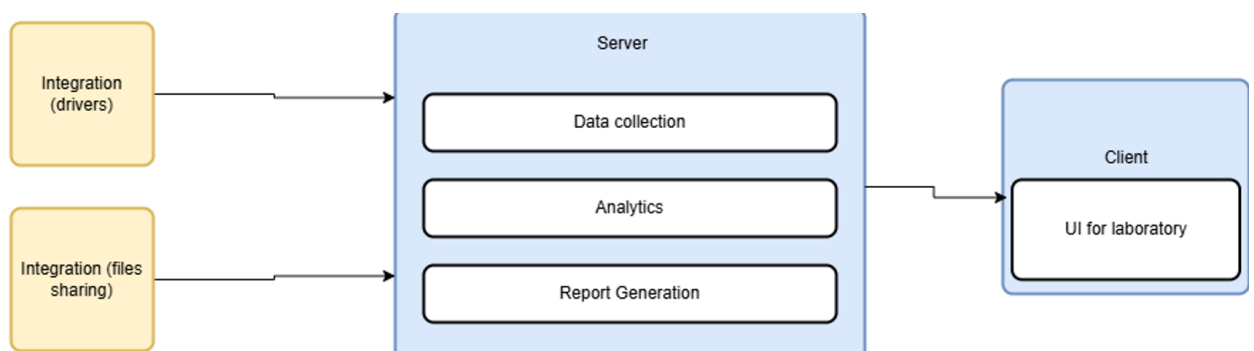


Рисунок 3.1 – Схема архітектури системи

На цьому етапі також фіксуються параметри для оцінки коефіцієнта автоматизації A , який згодом використовується у формулі індексу $IQ3$.

Параметри для оцінки коефіцієнта автоматизації:

$K_{авт}$ – кількість автоматизованих тестів;

$K_{руч}$ – кількість ручних тестів.

Оцінка коефіцієнта автоматизації визначається за формулою:

$$A = \frac{K_{авт}}{K_{руч}}$$

Розрахунок:

Проведено 25 автоматизованих тестів та 15 ручних:

$$A = \frac{25}{15} = 1,66$$

При використанні такої форми формули встановлено порогове значення коефіцієнта, $A \geq 1,5$, як цільовий мінімум для прийняттого рівня автоматизації.

За отриманими даними, значення коефіцієнта автоматизації становить 1,66%, що перевищує встановлений пороговий рівень у 1,5%. Це свідчить про достатній рівень автоматизації.

Крок 2.3. Планування процесів автоматизації.

Створюється паплайн автоматизованого контролю якості, який включає:

- збір даних з приладів у реальному часі;
- запуск автоматичних unit- та інтеграційних тестів при кожному оновленні;
- інтеграцію з Git та CI/CD-сервісами;
- перевірку продуктивності;
- логування, моніторинг помилок, резервне копіювання.

На цьому етапі визначаються майбутні контрольні точки для формул $IQ1$

(дефекти), $IQ2$ (середній час на виправлення), $IQ5$ (стабільність після оновлення).

Крок 2.4. Визначення контрольних точок якості.

Формалізуються ключові показники ефективності системи контролю якості. В таблиці 3.1 відображено ключові показники ефективності системи контролю якості.

Таблиця 3.1 – Ключові показники ефективності системи контролю якості

Показник	Позначення	Призначення
Кількість дефектів на 100 рядків коду	D	Розрахунок індексу дефектів ($IQ1$)
Середній час на виправлення помилки	T	Розрахунок індексу часу усунення ($IQ2$)
Частота збоїв після оновлення	E	Розрахунок індексу стабільності ($IQ5$)
Відсоток автоматизованих збоїв після оновлення	A	Розрахунок індексу автоматизації ($IQ3$)
Середній бал користувача	F	Розрахунок індексу користувацького досвіду ($IQ4$)
Відхилення якості продукції	Q	Розрахунок впливу на продукт ($IQ6$)

Результатом етапу планування є чітке формування базової лінії якості, від якої надалі будуть обчислюватися всі індекси якості. Визначені порогові значення:

D_{\max} – максимально припустима дефектів (наприклад, 30 на 1000 рядків коду);

T_{\max} – критичний час усунення помилки (наприклад, 2 доби);

E_{\max} – припустима кількість збоїв після релізу (наприклад, 10 збоїв);

Q_{\max} – допустиме відхилення параметрів якості (в межах 4%);

A , F , Q – задаються відповідно до очікувань проєкту.

Етап планування дозволив сформувати чітку дорожню карту реалізації підсистеми, структуру команди, технічну архітектуру та контрольні точки якості, які будуть надалі використовуватись для формального оцінювання за допомогою формул $IQ1$ - $IQ6$.

На цьому етапі не виконується розрахунок формул, але вже готуються всі значення, необхідні для розрахунків у фазах розробки, інтеграції, тестування та впровадження.

Етап 3 Розробка.

Крок 3.1. Реалізація модулів та функціональних блоків.

Було створено основні компоненти підсистеми «Лабораторія контролю якості», зокрема: модуль введення лабораторних даних, модуль аналітичної обробки результатів, модуль формування звітів та візуалізації. Реалізація проводилася згідно зі структурою, визначеною на етапі планування, із суворим дотриманням принципів SOLID. Для кожного модуля було впроваджено базову бізнес-логіку, забезпечено відповідність вимогам інтеграції з лабораторними приладами через стандартний протокол зв'язку та реалізовано механізми передачі результатів на серверну частину системи.

Формуються умови для майбутнього контролю якості, зокрема, для розрахунку дефектності коду на наступних кроках.

Крок 3.2. Внутрішній контроль якості.

Цей крок включає ревізію коду іншими учасниками команди перед злиттям змін у основну гілку. Виявляються помилки, порушення стандартів, архітектурні невідповідності.

Формула для оцінки якості:

Індекс дефектів:

Для оцінки якості коду підсистеми було виконано ревізію з використанням ручного перегляду, а також статичного аналізу. У процесі

перевірки виявлено 12 дефектів на 1000 рядків коду, що було зафіксовано під час внутрішнього аудиту після завершення розробки модулів. Допустиме граничне значення для цього показника було встановлено як 30 дефектів на 1000 рядків. На основі цих значень індекс якості розраховано за формулою:

$$IQ1 = 1 - \left(\frac{12}{30}\right) = 1 - 0,4 = 0,6$$

Індекс нижчий за рекомендований поріг (≥ 0.85), отже, код потребує доопрацювання.

Крок 3.3. Написання юніт- та інтеграційних тестів.

З метою підвищення ефективності тестування було впроваджено автоматизовані unit та integration тести. Загальна кількість тестів становила 100, з яких 85 були автоматизовані:

$$IQ3 = \frac{85}{100} = 0,85$$

Аналіз часу реагування на виявлені помилки здійснювався на основі журналів задач, які супроводжували виявлення й усунення дефектів протягом усіх етапів тестування. Було зафіксовано, що середній час від постановки помилки до її усунення становив 6 годин. При цьому допустимим вважався максимум у 16 годин, з урахуванням складності системи та виробничих обмежень:

$$IQ2 = 1 - \left(\frac{6}{16}\right) = 1 - 0,375 = 0,625$$

Автоматизація на високому рівні, іле індекс виправлення свідчить про потребу в оптимізації процесу вирішення дефектів.

Крок 3.4. Побудова зворотного зв'язку з тестуванням.

Після реалізації основного функціоналу підсистеми було організовано зворотний зв'язок із користувачами – фахівцями лабораторії, які працювали з інтерфейсом у тестовому середовищі. З метою кількісної оцінки задоволеності функціоналом було проведено коротке опитування, у якому користувачі оцінили систему за критеріями зручності, швидкодії та інтуїтивності інтерфейсу. Зазначимо, що середній бал відгуків – 4.6:

$$IQ4 = \frac{4,6}{5} = 0,92$$

Висока оцінка досвіду користувачів свідчить про зручність і стабільність реалізованого функціоналу.

Ці показники дозволяють проводити щоденний моніторинг якості, приймати рішення про доопрацювання модулів, коригувати тест-підхід або підсилювати команду. Якщо середнє значення $IQ < 0.85$ – модуль не допускається до інтеграції.

Етап 4 Інтеграція.

Крок 4.1. Інтеграційне тестування.

Для оцінки стабільності системи після внесення змін та виконання інтеграцій було проаналізовано кількість збоїв, що виникли одразу після кожного оновлення. У результаті журналювання інтеграцій за допомогою DevOps-інструментів (CI/CD) було зафіксовано 3 збої. Як порогове значення визначено максимум у 10 збоїв, відповідно до вимог безпеки та надійності системи:

$$IQ5 = 1 - \left(\frac{3}{10}\right) = 1 - 0,3 = 0,7$$

Показник середній – система ще нестабільна після інтеграції, потрібне

доопрацювання окремих точок дотику між модулями.

Крок 4.2. Автоматизація процесу інтеграції.

На етапі інтеграції всі дії в CI-процесі були автоматизовані: виконання тестів, розгортання в тестовому середовищі та логування помилок. У результаті аналізу логів CI/CD-системи було встановлено, що автоматизовано 90% тестів:

$$IQ3 = \frac{90}{100} = 0,9$$

Під час аналізу інтеграційних процесів було зафіксовано, що середній час реагування на критичні помилки становив 4 години. Допустимий поріг для цього індексу встановлено на рівні 12 годин, що обґрунтовано виробничими вимогами до швидкості усунення:

$$IQ2 = 1 - \left(\frac{4}{12}\right) = 1 - 0,3333 = 0,6667$$

Високий рівень автоматизації, але показник виправлення свідчить про затримки в реагуванні – варто оптимізувати логування і оповіщення.

Крок 4.3. Верифікація відповідності якості.

Для визначення реального впливу підсистеми «Лабораторія контролю якості» на кінцеві характеристики молочної продукції було проаналізовано дані, зібрані з лабораторних приладів упродовж контрольного періоду. У результаті було зафіксовано контрольований параметр у межах 1,2%, тоді як згідно з вимогами стандарту ISO 22000, максимально допустиме відхилення не повинно перевищувати 4%:

$$IQ6 = 1 - \left(\frac{1,2}{4}\right) = 1 - 0,3 = 0,7$$

Показник нижчий від бажаного – ефективність впливу ІС на кінцеву якість продукції потребує покращення.

Етап 5 Тестування.

Крок 5.1. Планування тестування.

На початку етапу була сформована детальна стратегія тестування, яка охоплювала всі типи перевірок, необхідних для комплексної оцінки якості підсистеми «Лабораторія контролю якості». Розроблено тестовий план, що включає:

- функціональні тести для перевірки правильності роботи ключових сценаріїв;
- нефункціональні тести для оцінки стабільності, інтерфейсної зручності, сумісності;
- інтеграційні тести для перевірки взаємодії між модулями;
- регресійні тести після внесення змін у код;
- користувацькі тести для оцінки очікувань кінцевих користувачів.

Було розподілено відповідальних фахівців за кожен напрям.

Встановлюються ключові метрики контролю якості, які далі будуть обчислюватись:

- кількість дефектів;
- середній час виправлення помилки;
- автоматизованість перевірок;
- середня оцінка користувачів;
- збої після оновлення;
- відхилення продукції.

Крок 5.2. Проведення ручного та автоматизованого тестування.

Тест-кейси виконуються вручну або за допомогою автоматизованих інструментів (Selenium, Postman, CI/CD скрипти). Виконуються:

- unit-тести – перевірка окремих функцій;
- інтеграційні тести – взаємодія між модулями;
- UI-тести – перевірка зручності інтерфейсу;

- регресійні тести – стабільність після змін;
- навантажувальні тести – продуктивність при великій кількості запитів;
- безпекові тести – вразливість до атак.

Крок 5.3. Аналіз результатів тестів з використанням метрик якості.

Після завершення всіх етапів проєкту та проведення відповідних тестувань було розраховано підсумковий індекс якості системи, що відображає інтегральну оцінку ефективності реалізованої підсистеми. Для цього були використані значення шести показників якості.

В таблиці 3.2 відображено розрахунки оцінки якості системи за формалізованими індексами.

Таблиця 3.2 – Розрахунки оцінки якості системи

Індекс	Виявлено	Максимально допустиме	Розрахунок
<i>IQ1</i>	7	20	0,65
<i>IQ2</i>	4	10	0,6
<i>IQ3</i>	85%	-	0,85
<i>IQ4</i>	4,2	-	0,84
<i>IQ5</i>	3	10	0,7
<i>IQ6</i>	1,2%	5	0,76

Розраховуємо загальний індекс якості:

$$IQ = \frac{0,65 + 0,6 + 0,85 + 0,84 + 0,7 + 0,76}{6} = \frac{4,4}{6} = 0,733$$

Поточне значення $IQ=0.733$ не досягає порогового значення 0.85, що свідчить про незадовільну якість. На підставі цього формується звіт про дефекти, визначаються області з високими відхиленнями (наприклад, час виправлення та стабільність оновлень), і приймається рішення про повторне тестування або перевірку підходів до автоматизації.

Етап 6 Впровадження:

Крок 6.1. Підготовка середовища впровадження.

На цьому етапі система була розгорнута в окремому ізолюваному середовищі, максимально наближеному до умов реального виробництва. Було налаштовано мережеву інфраструктуру, розгорнуто сервери, базу даних, модулі обробки та контролю якості, а також забезпечено доступи для адміністратора та відповідальних користувачів. Проведено перевірку доступності служб, з'єднань з лабораторними приладами та відповідність конфігурації цільовим умовам експлуатації.

Крок 6.2. Проведення фінального тестування на продуктивність.

Після повного розгортання підсистеми було проведено фінальне стрес-тестування з імітацією навантаження, максимально наближеного до умов реального виробництва. У процесі тестування зафіксовано 2 збої при максимальному допустимому рівні в 10. В таблиці 3.3 відображено розрахунки за формулами. У межах післявпроваджувального аналізу було досліджено зміни параметрів якості молочної продукції, які підлягають контролю з боку лабораторної підсистеми. Відхилення склало 1,5% при допустимому порозі у 5%.

Таблиця 3.3 – Розрахунки стабільності після оновлення та вплив на продукт

Індекс	Виявлено	Максимально допустиме	Розрахунок
<i>IQ5</i>	2	10	0.8
<i>IQ6</i>	1,5%	5	0.7

Отримані індекси свідчать про те, що система після впровадження демонструє прийнятний рівень стабільності та впливу на якість, однак дещо не досягає визначеного порогу у 0.85. Зокрема, індекс впливу на продукт вказує на наявність незначних відхилень, які можуть бути критичними в контексті харчової галузі.

У зв'язку з цим рекомендовано:

- провести додатковий аналіз причин відхилень;
- перевірити сценарії пікового навантаження та поведінку системи в нестандартних умовах;
- доопрацювати окремі модулі або механізми обробки даних, які могли викликати збої.

Крок 6.3. Післявпроваджувальний моніторинг та підтримка.

Упродовж 14 днів після впровадження здійснювався активний моніторинг функціонування системи: відслідковувались логі, збирались відгуки операторів лабораторії, DevOps-команда контролювала стабільність серверного середовища. Фіксувалися всі виявлені недоліки або відхилення – для кожного з них було розраховано відповідний індекс якості. У разі перевищення допустимого порогу – застосовувалося відкат версії або оперативні виправлення.

У результаті апробації розробленого комбінованого підходу забезпечення якості, що поєднує підходи Agile, DevOps і стандарт ISTQB, було реалізовано системний процес контролю якості на всіх етапах життєвого циклу підсистеми «Лабораторія контролю якості».

4 ОПИС ПРАКТИЧНОГО ВИКОРИСТАННЯ ОТРИМАНИХ ПРАКТИЧНИХ ТЕОРЕТИЧНИХ РЕЗУЛЬТАТІВ КР

4.1 Перевірка застосованості методу на практичному кейсі

Для перевірки дієвості запропонованого комбінованого методу забезпечення якості була змодельована ситуація застосування підсистеми «Лабораторія контролю якості» в інформаційній системі сирзаводу. Етапи життєвого циклу ІТ-проєкту (від ініціації до впровадження) супроводжувались застосуванням адаптованих кроків комбінованого підходу, що включає принципи Agile, DevOps та ISTQB.

На початковому етапі ініціації була сформована команда, окреслено технічне завдання та визначено потенційні ризики щодо якості майбутнього рішення. Далі на етапі планування відбулося моделювання процесів розробки з урахуванням контрольних точок та критеріїв оцінки. Формалізація задач, створення карти ризиків і попередній вибір метрик контролю дали змогу встановити чіткі межі якості, які потім стали основою для моніторингу та прийняття рішень.

Під час розробки основна увага була зосереджена на збиранні показників – таких як кількість дефектів, середній час на виправлення помилок, відсоток автоматизованих тестів – з подальшим розрахунком індексів якості. Результати розрахунків демонстрували стабільність і контрольованість процесу. Після кожного спринту (розробки окремих функцій) проводилось тестування з автоматичними перевітками й верифікацією відповідності до вимог користувачів лабораторії.

Інтеграція модулів відбувалась із суворим дотриманням CI/CD практик: застосовувались автоматизовані тести, відстеження змін, логування помилок, перевірка резервного копіювання даних. Це дозволило мінімізувати технічні ризики при переході між середовищами (наприклад, із тестового у продуктивне).

Під час впровадження проводилось остаточне оцінювання середнього індексу якості. Якщо його значення перевищувало встановлений поріг (0,85), рішення вважалось придатним до запуску. У разі недосягнення порогових значень активувались механізми коригування – повторне тестування, зміна підходу, уточнення вимог.

Таким чином, метод показав свою ефективність у практичному застосуванні, дозволяючи здійснювати як контроль, так і гнучке адаптування процесів. Він забезпечив повний цикл контролю якості: від оцінки ризиків до прийняття рішення про впровадження підсистеми.

4.2 Аналіз ефективності запропонованого підходу

У результаті апробації комбінованого підходу до забезпечення якості підсистеми «Лабораторія контролю якості» ІС сирзаводу було проведено послідовну оцінку якості на основі визначених у дослідженні формул. Методика оцінювання ґрунтувалася на комплексній системі індексів якості, які враховують ключові метрики – кількість дефектів, час на виправлення помилок, частку автоматизованого тестування, користувацькі відгуки, стабільність після оновлень та відхилення характеристик продукту.

Згідно з розробленим підходом, для кожного індексу була визначена формула, а результати перевірки поетапно узагальнені у вигляді середнього значення – індексу якості. Розрахунок середнього значення здійснено на основі фактичних показників під час реалізації розробки, без перевищення допустимих меж.

Таким чином, комбінований підхід забезпечив інтеграцію гнучких (Agile), автоматизованих (DevOps) та стандартизованих (ISTQB) компонентів контролю якості, які були адаптовані до кожного з етапів життєвого циклу підсистеми. Висока узгодженість дій команди, своєчасне виявлення відхилень

та контроль критичних точок сприяли досягненню стабільних і прозорих результатів.

Формули, використані у моделі, чітко співвіднесені з кроками етапів, а їх розрахунок слугує інструментом прийняття рішень на основі метрик, релевантних для даного ІТ-проєкту. Такий підхід дозволив підтвердити практичну ефективність розробленого методу.

Аналіз отриманих результатів свідчить про здатність методу адаптуватися до змінних умов розробки та масштабуватися відповідно до специфіки процесів конкретного виробничого середовища. Особливе значення мала можливість кількісно відстежувати динаміку якості на кожному етапі – від планування до впровадження – що дозволяло не лише виявляти недоліки, а й прогнозувати ризики ще до моменту їх виникнення.

Підхід виявився ефективним і в контексті взаємодії між технічними та організаційними складовими. Наприклад, зменшення часу на виправлення помилок (T), підвищення частки автоматизованих тестів (A) і стабільна динаміка зворотного зв'язку користувачів (F) продемонстрували злагодженість між етапами тестування, інтеграції та впровадження. Це забезпечило зниження загального ризику зриву проєкту та підвищення довіри до кінцевого продукту.

З огляду на результати апробації, можна стверджувати, що розроблений метод не лише забезпечує формальний контроль якості, а й виступає дієвим управлінським інструментом, який посилює аналітичну основу прийняття рішень у сфері розробки ІС для харчової промисловості.

4.3 Використання методу для навчання персоналу

Розроблений комбінований метод забезпечення якості підсистеми «Лабораторія контролю якості» може ефективно застосовуватись не лише під

час створення та впровадження ІС, але й як інструмент у процесі навчання та підвищення кваліфікації персоналу сирзаводу. Його структурованість, етапність і формалізованість дозволяє використовувати метод як основу для розробки навчальних програм та внутрішніх тренінгів.

Однією з головних переваг методу є його модульність, завдяки якій навчання може бути адаптоване до конкретної ролі працівника: від технічного персоналу лабораторії до ІТ-спеціалістів, відповідальних за супровід і тестування. Наприклад, етап «Розробка» містить елементи роботи з інструментами тестування, оцінки метрик якості та контролю критичних точок, що дозволяє новим працівникам швидко включитися у робочий процес.

Методи, які включають Agile-практики, DevOps-підхід та стандарти ISTQB, можуть бути інтегровані у навчальні модулі та симуляції. Працівники мають змогу ознайомитися з механізмами CI/CD, правилами роботи з тест-кейсами, формулюванням вимог до якості, методами контролю на кожному етапі розробки. Використання формалізованих формул (наприклад, для розрахунку індексу якості або кількості дефектів на тисячу рядків коду) дозволяє працівникам краще розуміти вимірювані показники та відстежувати зміни у продуктивності.

Крім того, метод охоплює й організаційні аспекти контролю, як-от взаємодія команд, аналіз ризиків, прийняття рішень за результатами моніторингу – це важливі компетенції не лише для технічного персоналу, а й для менеджерів. Таким чином, використання методу в освітньому контексті сприяє формуванню системного бачення забезпечення якості, орієнтованого на постійне вдосконалення.

Впровадження комбінованого методу у навчальні процеси дозволяє створити єдину культуру якості, підвищити усвідомленість персоналу щодо важливості контролю на кожному етапі життєвого циклу системи, а також формує сталі навички використання сучасних підходів до управління якістю в харчовій галузі.

4.4 Розширення методу на інші підсистеми ІС підприємства

Розроблений комбінований метод забезпечення якості, впроваджений у підсистемі «Лабораторія контролю якості», має універсальну структуру, що дозволяє масштабувати його застосування на інші підсистеми інформаційної системи сирзаводу без втрати ефективності. Така можливість ґрунтується на тому, що метод базується на поетапному підході управління якістю, який охоплює весь життєвий цикл ІТ-проєкту – від ініціації до впровадження, з чітко визначеними критеріями оцінки, контрольними точками та алгоритмами прийняття рішень.

Завдяки структурі методу, яка поєднує принципи Agile, DevOps та стандарти ISTQB, він може бути адаптований до таких підсистем, як:

- підсистема управління виробництвом, де акцент буде зроблено на безперебійну інтеграцію модулів автоматизації, контроль часу простою обладнання та ефективність технологічного процесу;
- підсистема логістики та складування, де контрольні точки можуть охоплювати швидкість обробки замовлень, точність складського обліку, стабільність обміну даними з іншими модулями;
- підсистема бухгалтерського обліку, де особливу увагу слід приділити точності обчислень, надійності збереження фінансової інформації та відповідності нормативним вимогам;
- підсистема управління персоналом, у якій метод може використовуватись для аналізу якості обробки даних, точності розрахунку заробітної плати та доступності електронного документообігу.

Формули оцінки індексів якості, розроблені в межах підсистеми лабораторного контролю, можуть бути адаптовані до особливостей конкретного модуля. Наприклад, замість кількості дефектів у коді може враховуватись кількість помилок при обміні даними між модулями або середній час реакції системи на запити користувача.

Уніфікація підходу до забезпечення якості забезпечує сумісність і єдині стандарти перевірки, що підвищує надійність всієї системи підприємства загалом. Це також спрощує навчання персоналу та супровід ІС, оскільки одна й та сама методика використовується в різних контекстах, з урахуванням специфіки окремих підсистем.

Таким чином, запропонований метод не лише довів свою ефективність на прикладі лабораторного контролю якості, а й виявив потенціал до широкого застосування в масштабі всієї інформаційної системи підприємства, сприяючи її стабільному розвитку, підвищенню ефективності внутрішніх процесів та якості управлінських рішень.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено повне дослідження процесів забезпечення якості підсистем ІС підприємств харчової промисловості. Визначено актуальність розробки ефективного підходу до управління якістю ПЗ в умовах сучасних виробничих викликів. Проаналізовано існуючі підходи, методи та стандарти контролю якості, обґрунтовано доцільність створення комбінованого методу з урахуванням специфіки харчової галузі.

На основі проведеного аналізу та вивчення методологій Agile, DevOps та стандарту тестування ISTQB було розроблено комбінований підхід до забезпечення якості підсистеми «Лабораторія контролю якості» ІС сирзаводу. Описано всі етапи життєвого циклу підсистеми – від ініціації до впровадження, з деталізацією конкретних кроків, контрольних точок, метрик якості та інструментів тестування.

Під час апробації методу на практичному кейсі розробки підсистеми було впроваджено формалізовану систему оцінки якості, що включає кількісні індекси ефективності.

Окрему увагу приділено можливостям масштабування методу на інші підсистеми ІС підприємства, зокрема, модулі логістики, обліку, управління персоналом тощо. Також обґрунтовано застосування результатів дослідження для навчання персоналу й покращення управлінських рішень у сфері ІТ-розробок для виробничих підприємств.

Поставлені у кваліфікаційній роботі цілі – аналіз, розробка та практична перевірка комбінованого підходу до забезпечення якості, були повністю досягнуті. Результати дослідження мають прикладну цінність і можуть бути використані для покращення процесів впровадження й підтримки високоякісних ІТ-рішень у харчовій промисловості.

Джерела посилання оформлені згідно ДСТУ [12].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методи забезпечення якості продукції на підприємстві. *DSpace* :: *ELAKPI* :: *Репозитарій КПІ ім. Ігоря Сікорського*. URL: <https://ela.kpi.ua/items/79758fe2-9f66-4b0f-870b-7ad4a181d446> (дата звернення: 31.05.2025).
2. Методичні вказівки до науково-дослідної практики для здобувачів вищої освіти усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-наукової програми «Управління проектами в галузі інформаційних технологій». / Упоряд.: К.Е. Петров, Євланов М.В., Цвіркун О.А.. – Харків: ХНУРЕ, 2025. – 19 с (дата звернення: 01.04.2025)
3. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. – Київ: ДП «УкрНДНЦ», 2016. – 31 с. (дата звернення: 01.04.2025)
4. Контроль якості продукції на підприємствах. URL: <https://tourism-book.com/pbooks/book-83/ua/chapter-3323/> (дата звернення: 01.04.2025)
5. Методології управління проектом URL: <https://www.staff.ua/uk/blog/metodologiji-upravlinnya-proektom> (дата звернення 01.04.2025)
6. Гнучка методологія розробки ПЗ Agile. URL: <https://training.qatestlab.com/blog/technical-articles/flexible-software-development-methodology-agile/> (дата звернення: 01.04.2025).
7. DevOps інженер: Навички та інструменти. *FoxmindEd*. URL: <https://foxminded.ua/shcho-maie-znaty-devops-inzhener/> (дата звернення: 02.04.2025)..
8. Сертифікат ISTQB. *QATestLab* | Головна сторінка. URL: https://training.qatestlab.com/blog/technical-articles/what-is-certificate-istqb/?utm_source=chatgpt.com (дата звернення: 02.04.2025).

9. 29-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., – Харків: ХНУРЕ. 2025. – 630 с.
10. HACCP та GMP: основи безпеки та якості харчових продуктів - BM Certification. *BM Certification*. URL: <https://ua.bmcertification.com/nassr-ta-gmp-osnovi-bezpeki-ta-yakosti-harchovih-produktiv/> (дата звернення: 01.06.2025).
11. Quality management in entrepreneurship. *ELAr* :: Головна. URL: <https://openarchive.nure.ua/entities/publication/1e0cd7cc-342d-43ce-a171-1f415baef999> (дата звернення: 30.05.2025).
12. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с. (дата звернення 01.06.2025).