

БАКАЛАВРСЬКА РОБОТА

БР. ІІ - 65.00.00.000 ІЗ

Група ІІ-22-3

Гуменяк Сергій

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Гуменяк Сергій Андрійович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

БАКАЛАВРСЬКА РОБОТА

Розробка програмних продуктів з урахуванням циклу швидкого зворотного зв'язку

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

**Робота містить результати власних досліджень, використання ідей, результатів і
текстів інших авторів мають посилання на відповідне джерело:**

Здобувач освітнього ступеня Гуменяк С.А.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Ксенич Андрій Іванович, к.т.н, доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу
Інститут, факультет інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти бакалавр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою _____ **ІПЗ**
доцент. _____ **В.В. Бандура**

“ _____ ” _____ 2026 р.

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Гуменяк Сергій Андрійович

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) " Розробка програмних продуктів з урахуванням циклу швидкого зворотного зв'язку "

керівник проекту (роботи) Ксеніч Андрій Іванович, к.т.н, доцент

затверджені наказом вищого навчального закладу від “ 21 ” квітня 2026 р. № 179/7

2. Строк подання студентом проекту (роботи) 10 червня 2026 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз вимог до програмного забезпечення

2. Аналіз вимог і постановка задачі

3. Проектування системи

4. Реалізація проекту

5. Тестування та впровадження

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1 Робочі процеси CI/CD (рис. 1.1, ст. 23)

2 Ланцюжок інструментів QFL у Softeam (рис. 2.2, ст. 42)

3 Панель інструментів Q-Rapids (візуалізація стратегічних показників (рис. 2.3, ст. 47)

4. Стан тестування (ліворуч) та цикл зворотного зв'язку щодо якості (праворуч) в історичних переглядах (рис. 2.4, ст. 49)

5 Тематична карта (рис. 3.1, ст. 53)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

2. Дата видачі завдання 2026 р.

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	17.01.2026	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	21.02.2026	виконано
3	Побудова моделі або алгоритму власного рішення	01.03.2026	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	21.04.2026	виконано
5	Оформлення пояснювальної записки бакалаврської роботи	02.05.2026	виконано

Студент

_____ Гуменyak С.А.
(підпис)

Керівник роботи

_____ Ксенич А.І.
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 81 сторінок, 6 рисунків, 3 таблиці, список використаних джерел із 40 найменувань.

Метою роботи є дослідження підходів до розробки програмних продуктів з урахуванням циклу швидкого зворотного зв'язку та аналіз методів підвищення якості програмного забезпечення за рахунок інтеграції механізмів збору, обробки та використання даних від користувачів і системних метрик.

Об'єкт дослідження: процеси розробки програмного забезпечення з використанням підходів швидкого зворотного зв'язку.

Предмет дослідження: методи, моделі та інструменти реалізації циклів швидкого зворотного зв'язку.

Результати дослідження: проведено аналіз сучасних підходів до організації зворотного зв'язку у процесі розробки ПЗ, визначено обмеження існуючих методологій, досліджено методи збору та обробки даних.

У першому розділі розглянуто теоретичні основи швидкого зворотного зв'язку, принципи його застосування в розробці програмного забезпечення, а також проаналізовано обмеження сучасних методологій.

У другому розділі досліджено методи та інструменти забезпечення швидкого зворотного зв'язку, включаючи архітектурні підходи, методи збору даних, аналіз метрик та інструменти моніторингу програмного забезпечення.

У третьому розділі розглянуто практичну реалізацію механізмів швидкого зворотного зв'язку, проведено оцінку їх ефективності.

Висновок: впровадження циклів швидкого зворотного зв'язку дозволяє значно підвищити якість програмного забезпечення, скоротити час розробки, забезпечити швидке реагування на потреби користувачів та покращити процес прийняття рішень у розробці програмних продуктів.

КЛЮЧОВІ СЛОВА: ЗВОРОТНИЙ ЗВ'ЯЗОК, FEEDBACK LOOP, AGILE, DEVOPS, МОНИТОРИНГ, АНАЛІТИКА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ЯКІСТЬ ПЗ, CI/CD.

ANNOTATION

The thesis consists of 81 pages, 6 figures, 3 tables, and a reference list with 40 sources.

The aim of the work is to study approaches to software development considering fast feedback loops and to analyze methods for improving software quality through the integration of data collection, processing, and utilization from users and system metrics.

Research object: software development processes using fast feedback loop approaches.

Research subject: methods, models, and tools for implementing fast feedback loops.

Research results: an analysis of modern approaches to organizing feedback in software development was conducted, limitations of existing methodologies were identified, methods of data collection and tools for software analytics and monitoring.

The first section describes the theoretical foundations of fast feedback loops, principles of their application in software development and limitations of modern methodologies.

The second section analyzes methods and tools for ensuring fast feedback, including architectural approaches, data collection methods, metric analysis, and software monitoring tools.

The third section covers the implementation of fast feedback mechanisms, evaluation of their effectiveness, and outlines future development directions of this approach.

Conclusion: the implementation of fast feedback loops significantly improves software quality, reduces development time, enables rapid response to user needs, and enhances decision-making processes in software development.

KEY WORDS: FEEDBACK LOOP, AGILE, DEVOPS, SOFTWARE MONITORING, ANALYTICS, SOFTWARE DEVELOPMENT, SOFTWARE QUALITY, CI/CD.

ЗМІСТ

ВСТУП	8
 РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ОРГАНІЗАЦІЇ ШВИДКОГО ЗВОРОТНОГО ЗВ'ЯЗКУ В РОЗРОБЦІ ПЗ	 10
1.1 Основні принципи та концепції швидкого зворотного зв'язку у розробці програмного забезпечення	10
1.2 Взаємодія користувача з програмними продуктами як джерело зворотного зв'язку	13
1.3 Аналіз сучасних методологій розробки та їх обмеження щодо швидкого зворотного зв'язку	22
1.4 Висновок по розділу	28
 РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ ЗАБЕЗПЕЧЕННЯ ШВИДКОГО ЗВОРОТНОГО ЗВ'ЯЗКУ	 29
2.1 Архітектурні та організаційні підходи до реалізації циклів зворотного зв'язку	29
2.2 Методи збору та обробки даних зворотного зв'язку	34
2.3 Інструменти аналітики та моніторингу програмного забезпечення	41
2.4 Висновок по розділу	49
 РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЦИКЛІВ ЗВОРОТНОГО ЗВ'ЯЗКУ	 50
3.1 Реалізація механізмів швидкого зворотного зв'язку	50
3.2 Аналіз результатів впровадження та вплив на якість програмного забезпечення	66
3.3 Перспективи розвитку та напрями подальших досліджень	75
3.4 Висновок по розділу	79
 ВИСНОВКИ	 80
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА	82

					БР.ІІ – 65.00.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка програмних продуктів з урахуванням циклу швидкого зворотного зв'язку Пояснювальна записка	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
Розроб.		Гуменяк С.А.						
Перевір.		Ксенич А.І.					7	
Реценз.		Зікратий С.В.				ІФНТУНГ ІІ-22-3		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В. В.						

ВСТУП

У сучасному світі інформаційні технології розвиваються надзвичайно швидкими темпами, що зумовлює постійне зростання вимог до якості програмного забезпечення, швидкості його розробки та здатності адаптуватися до змін потреб користувачів. Одним із ключових чинників успішної розробки програмних продуктів є ефективна організація процесу отримання та обробки зворотного зв'язку від користувачів. Саме впровадження циклу швидкого зворотного зв'язку дозволяє оперативно виявляти недоліки, коригувати функціональність та забезпечувати відповідність продукту очікуванням кінцевих користувачів.

Актуальність теми зумовлена необхідністю підвищення ефективності процесів розробки програмного забезпечення в умовах високої конкуренції та швидких змін ринку. Традиційні підходи до розробки часто не забезпечують достатньої гнучкості та швидкості реагування на зміни вимог, що призводить до збільшення витрат часу та ресурсів. У цьому контексті особливого значення набувають підходи, орієнтовані на безперервний зворотний зв'язок, зокрема Agile, DevOps та DevSecOps, які дозволяють інтегрувати процеси розробки, тестування та експлуатації в єдиний безперервний цикл.

Метою роботи є дослідження та аналіз підходів до розробки програмних продуктів з урахуванням циклу швидкого зворотного зв'язку, а також визначення ефективних методів і інструментів для його реалізації у сучасних умовах.

Завданнями дослідження є: аналіз теоретичних основ організації циклів зворотного зв'язку у процесі розробки програмного забезпечення; дослідження взаємодії користувача з програмними продуктами та її впливу на якість зворотного зв'язку; аналіз сучасних підходів до розробки, зокрема Agile, SSDLC та DevSecOps, і визначення їх обмежень; дослідження методів збору, обробки та аналізу даних зворотного зв'язку; огляд інструментів аналітики програмного забезпечення; реалізація підходів до впровадження циклу швидкого зворотного зв'язку та оцінка їх ефективності.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження є процеси розробки програмних продуктів, що включають етапи аналізу вимог, проектування, реалізації, тестування та супроводу.

Предметом дослідження є методи, моделі та інструменти організації циклу швидкого зворотного зв'язку в процесі розробки програмного забезпечення, а також їх вплив на якість і швидкість створення програмних продуктів.

Методи дослідження включають аналіз наукових джерел і сучасних практик розробки, порівняльний аналіз підходів, моделювання процесів взаємодії користувача з системою, експериментальне впровадження інструментів збору зворотного зв'язку, а також оцінку ефективності запропонованих рішень.

У роботі передбачається дослідження механізмів інтеграції зворотного зв'язку на різних етапах життєвого циклу програмного забезпечення, включаючи етапи розробки, тестування та експлуатації. Особлива увага приділяється використанню аналітичних інструментів, автоматизації збору даних та впровадженню безперервних процесів удосконалення продукту.

Очікується, що результати дослідження дозволять підвищити ефективність процесу розробки програмного забезпечення, скоротити час виведення продукту на ринок, покращити якість кінцевого продукту та забезпечити більш високий рівень задоволеності користувачів.

Бакалаврська робота містить 81 сторінок, 6 рисунків, 3 таблиці, 3 розділи, висновки та список використаних джерел із 40 найменувань.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ОРГАНІЗАЦІЇ ШВИДКОГО ЗВОРОТНОГО ЗВ'ЯЗКУ В РОЗРОБЦІ ПЗ

1.1 Основні принципи та концепції швидкого зворотного зв'язку у розробці програмного забезпечення

У сучасному конкурентному середовищі розробки програмного забезпечення, де користувачі мають широкий вибір, орієнтація на користувача стала критичним фактором успіху. Кілька підходів підкреслюють важливість потреб користувачів, такі як концепція «Системного мислення» та «Дизайнерське мислення». Ці рамки підкреслюють, що процес розробки має бути зосереджений навколо користувачів. Для досягнення цього елементи повинні бути орієнтовані на користувача, забезпечуючи відповідність кожної точки дотику потребам та уподобанням користувача [1]. Такий спосіб мислення означає підтримку постійного діалогу з користувачами для керівництва розробкою продукту та адаптації до їхніх очікувань, що змінюються. Apple, наприклад, демонструє свою відданість орієнтації на користувача через інтуїтивно зрозумілий дизайн та бездоганну зручність використання, а також активно враховуючи відгуки клієнтів. Огляди програм у iTunes Store надають цінну інформацію, яку Apple використовує для покращення функціональності та користувацького досвіду. Цей ітеративний цикл зворотного зв'язку не лише покращує якість продукту, але й сприяє більшій залученості користувачів, що зрештою призводить до більшої кількості завантажень додатків та задоволеності клієнтів.

Такий орієнтований на користувача спосіб мислення тісно узгоджується зі зростаючим зрушенням у методологіях розробки програмного забезпечення. Протягом останніх десятиліть відбувся помітний перехід від традиційної розробки до гнучкої розробки, оскільки гнучкі методи сприяють більш динамічній та гнучкій розробці продуктів [2]. Як підкреслюють, гнучка розробка зосереджується на ключових принципах, таких як «співпраця, адаптивність,

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

постійне вдосконалення та ітеративна розробка», які сприяють ефективному створенню високоякісного програмного забезпечення. Співпраця має вирішальне значення в ітеративній розробці, де регулярний зворотний зв'язок з користувачами відіграє вирішальну роль у постійному вдосконаленні та покращенні продукту. Ці цикли зворотного зв'язку є важливими для забезпечення відповідності продукту потребам користувачів та його відповідності вимогам ринку. Регулярний зворотний зв'язок також сприяє прозорості та допомагає розставляти пріоритети завдань, дозволяючи швидко виявляти зміни та уникати непотрібних затримок. Як зазначають, що «надання цінності клієнтам на ранніх стадіях часто вимагає глибокого розуміння потреб клієнтів» [3]. Тому вкрай важливо ефективно впроваджувати цикли зворотного зв'язку. Затримки або відсутність зворотного зв'язку з користувачами можуть мати негативні наслідки. Тривалі цикли зворотного зв'язку можуть призвести до затримок в отриманні інформації від користувачів, що може призвести до того, що продукт не повністю задовольнятиме потреби користувачів.

Хоча дослідження, обговорюють проблеми, пов'язані з циклами зворотного зв'язку, вони не пропонують рішень для їх подолання [4]. Це дослідження визначає рішення як практичні способи, за допомогою яких організації успішно впроваджують цикли зворотного зв'язку з користувачами в реальних гнучких середовищах. Інша література зосереджується лише на окремих аспектах, таких як покращення залучення зацікавлених сторін або використання UX-дизайну (UX). Крім того, деякі дослідження зосереджуються переважно на безперервному зворотному зв'язку в контексті організаційних змін, а не безпосередньо на конкретних проблемах інтеграції зворотного зв'язку з користувачами в гнучку розробку програмного забезпечення.

Хоча важливість ефективного зворотного зв'язку з користувачами добре відома, а в літературі обговорюються проблеми інтеграції зворотного зв'язку з користувачами, залишається прогалина, оскільки рішення часто або не надаються, або зосереджуються на окремих аспектах. Тому важливо визначити існуючі практичні рішення та надати компаніям всебічний огляд. Систематично

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

аналізуючи ці рішення, компанії можуть приймати обґрунтовані рішення щодо своїх гнучких процесів та повною мірою реалізувати переваги тісної участі користувачів. Це дослідження має на меті заповнити цю прогалину, вивчивши перешкоди для ефективних циклів зворотного зв'язку з користувачами та пропонуючи практичні рішення для їх подолання.

Дослідницька проблема

Дослідження показали, що практичне впровадження циклів зворотного зв'язку стикається з труднощами, і багато компаній мають труднощі з отриманням швидкого та чіткого зворотного зв'язку від користувачів. Цей зворотний зв'язок є критично важливим, оскільки задоволення та перевищення очікувань користувачів дає компаніям конкурентну перевагу [5]. Тривалі або відсутні цикли зворотного зв'язку спричиняють затримки в розробці та невизначеність щодо розробки продукту. Більше того, дослідження, висвітлюють такі проблеми, як затримки у зборі відгуків користувачів, відсутність залучення користувачів та труднощі у встановленні спільного розуміння точки зору користувача. Однак ці дослідження не пропонують практичних рішень для вирішення цих проблем. Як зазначають «існує нестача систематичних підходів та структур для вирішення цих проблем».

Підсумовуючи, існує розрив між теоретичним визнанням зворотного зв'язку з користувачами як важливої складової гнучкої розробки та його практичним впровадженням [6]. Основна проблема полягає в тому, що, незважаючи на визнану важливість, багато організацій намагаються систематично інтегрувати зворотний зв'язок з користувачами у свої гнучкі процеси. Неефективні механізми зворотного зв'язку створюють невідповідність між потребами користувачів та рішеннями щодо продукту, що зрештою впливає на успіх продукту.

Дослідницькі питання

З дослідницької проблеми впливли наступні два дослідницькі питання:

1. Які поширені проблеми впровадження циклів зворотного зв'язку з користувачами були виявлені в літературі?

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Грунтуючись на існуючій літературі, це дослідницьке питання визначає поширені проблеми впровадження циклів зворотного зв'язку з користувачами та забезпечує теоретичну основу для дослідження. Друге дослідницьке питання розглядає, як рішення застосовуються на практиці та як можна мінімізувати проблеми, пов'язані з їх впровадженням.

2. Як на практиці впроваджуються рішення для покращення циклів зворотного зв'язку з користувачами в гнучкій розробці програмного забезпечення, і як пом'якшуються проблеми впровадження?

Цілі та призначення

Дослідження має на меті надати комплексний огляд рішень для покращення циклів зворотного зв'язку з користувачами в гнучкій розробці програмного забезпечення. У ньому окреслено рішення для пом'якшення таких проблем, як затримка зворотного зв'язку, неефективна пріоритезація та недостатня участь користувачів [7]. Аналіз вибраної літератури визначить поширені проблеми, тоді як якісне опитування, підкріплене інтерв'ю, виявить рішення. Зібрані дані будуть проаналізовані за допомогою тематичного аналізу. Ці результати будуть класифіковані, і будуть надані практичні рекомендації для гнучких команд. Рекомендації спрямовані на поглиблення розуміння типових перешкод у процесах зворотного зв'язку та надання рішень для максимізації цінності для клієнтів. Результати будуть особливо актуальними для фахівців в ІТ та програмній галузі, де широко використовуються гнучкі методології.

1.2 Взаємодія користувача з програмними продуктами як джерело зворотного зв'язку

Вивчення взаємодії людини з комп'ютером (HCI) вже давно надає уявлення про найкращі практики проектування для покращення взаємодії користувача з продуктами, програмами та іншими технологічними інтерфейсами [20]. Хоча це значною мірою сприяло еволюції дизайну зручності використання, конкуруюча якість між галузями HCI та кібербезпеки привернула певну увагу.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Зростання обізнаності та подальше вивчення кібербезпеки зіткнулося з багатьма викликами у впровадженні. Найбільше уваги приділяється виклику впровадження безпеки в життєвий цикл розробки програмного забезпечення (SDLC) для створення «безпечного життєвого циклу розробки програмного забезпечення» (SSDLC) або «безпечного життєвого циклу розробки» (SDL) [13]. Основна сфера фокусування, що впливає з ідеї SDL, - це перехід від DevOps до DevSecOps, де гнучка розробка поєднується з процесами безпеки (наприклад, безпека через проектування та огляди безпеки), безперервною інтеграцією, безперервним розвитком (CI/CD), а також IT/операційними завданнями, пов'язаними з розгортанням програмного продукту.

Було безліч пропозицій щодо найкращої інтеграції програмного забезпечення та процесів безпеки, але часто не враховується включення HCI та UX-дизайнерів до цієї нової міжфункціональної SDL. Враховуючи той факт, що застосування принципів UX-дизайну до кінцевого продукту може значно вплинути на ефективність/успішність цих додаткових засобів контролю безпеки для користувача, вкрай важливо посилити цей цикл зворотного зв'язку, включивши дизайнерів до процесу розробки як ще одну зацікавлену сторону. Адаптуючи модель DevSecOps, ми пропонуємо нову модель розробки, яка включає дизайнерів до циклу зворотного зв'язку.

У цьому підрозділі буде розглянуто поточну роботу в цій галузі. Це включає роботу над еволюцією SDLC, яка включає безпеку та гнучку розробку, дослідження/дизайн UX, а також взаємозв'язок між зручністю використання та безпекою.

Еволюція SDLC

Існують різні версії SDLC, адаптовані для визначення, стандартизації та підходу до процесу розробки програмного забезпечення. Моделі SDLC зосереджені на визначенні проблемної області, розумінні її обсягу, визначенні області рішення та окресленні реалізації програмного забезпечення для вирішення цієї проблемної області.

Вступ до SDLC та методологій розробки. SDLC був запроваджений у

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1960-х роках, щоб допомогти стандартизувати підхід до розробки великомасштабних інформаційних систем у часи, коли управління проектами було складним у великих бізнес-конгломератах, де материнські компанії володіли кількома дочірніми компаніями в різних галузях [7]. SDLC – це модель процесу, що описує, як підходити до проектування програмного забезпечення для вирішення певної проблемної області [14].

Ця базова модель SDLC допомогла прокласти шлях та стандартизувати підхід до впровадження програмного забезпечення. Однак базова SDLC не задовольняла всі потреби розробки.

Помітною адаптацією SDLC є модель водоспаду, яка була представлена в 1970 році. Ця модель була розроблена як лінійна та послідовна часова шкала фаз. Кожен етап моделі водоспаду має бути завершений перед переходом до наступного, і вона не призначена для повторного перегляду останнього етапу.

Ключова перевага використання цієї моделі полягає в тому, що її легко дотримуватися та розуміти. Це добре працює, якщо програмне забезпечення, що розробляється, має суворі та зрозумілі вимоги. Однак, оскільки кожен етап має бути завершений перед переходом до наступного, ця модель може призвести до дуже повільної розробки, а також ненавмисно зробити продукт негнучким (з точки зору часу та витрат) до адаптації, якщо пізніше в процесі виникнуть будь-які зміни. Через ці обмеження модель водоспаду підійде не всім.

V-подібна модель – це ще одна адаптація SDLC. Як і модель водоспаду, вона також є послідовною, але вводить два додаткові шари паралельних фаз розробки тестів та фаз тестування. Це підвищує жорсткість моделі, зберігаючи при цьому низьку складність. Введення фаз розробки тестів на кожному рівні також допомагає запровадити додаткове охоплення аналізу ризиків протягом усього процесу розробки, саме в цьому недоліки оригінальної моделі водоспаду [14].

Також, як і модель водоспаду, V-подібна модель може бути повільною та трудомісткою, що не масштабується для довгострокових проектів. Цей процес та всі необхідні для нього розробки/тестування тестів також були б надмірними для

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

проекту з вужчим обсягом [8]. Зі зростанням технологічного ландшафту методології SDLC змінювалися, щоб краще підтримувати швидші, ефективніші та гнучкіші практики розробки.

Гнучка методологія та SDLC. Технологічна галузь сприяла розчаруванню дорогими та трудомісткими методологіями SDLC. Великі проміжки часу між визначенням вимог та їх постачанням часто призводили до збоїв проектів, скасування або дороговартісних переробок, якщо вимоги з часом змінювалися.

Спільнота розробників програмного забезпечення шукала легшу - методологію для стандартизації SDLC, яка б підтримувала ідеалізований швидший, ефективніший та гнучкіший процес розробки. У 2000-2001 роках група з сімнадцяти людей у галузі зустрілася та розробила так званий Agile Manifest. У маніфесті було окреслено чотири ключові цінності цієї нової Agile методології, а також дванадцять допоміжних принципів, які допоможуть розробникам програмного забезпечення впровадити цю нову методологію. Чотири ключові цінності такі:

1. Люди та взаємодія важливіші за процеси та інструменти;
2. Працююче програмне забезпечення важливіше за вичерпну документацію;
3. Співпраця з клієнтами важливіша за укладання контрактів та
4. Реагування на зміни, ніж за планом.

Маніфест Agile [3] мав на меті перерозподілити та змінити пріоритети важливості різних рухомих частин SDLC, надаючи перевагу до дотримання процесів, інструментів, вичерпної документації, контрактів та визначеного плану, за умови, що зміни сприяють місії надання програмного забезпечення, яке успішно та своєчасно відповідає вимогам.

З огляду на ці чотири цінності, процес SDLC було модифіковано. Замість дорогого процесу послідовного проходження фаз розробки, Agile-методологія заохочувала більше поетапної, ітеративної та одночасної розробки фаз SDLC. Такий підхід забезпечував більшу гнучкість для адаптації до змін у вимогах,

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

подолання перешкод та зниження витрат на це. Цей ітеративний та поетапний підхід до розробки значно відійшов від традиційної моделі водоспаду, яка вимірювала прогрес у лінійному послідовному процесі.

Гнучкі команди спираються на інструменти автоматизації, які допомагають забезпечити безперервну інтеграцію та безперервну доставку (CI/CD). Ми детальніше розглянемо CI/CD пізніше.

Хоча Agile-методологія вирішила деякі проблеми, визначені у водоспадній моделі, було виявлено, що є дещо, що Agile-методологія не враховувала. Що було далі після того, як програмне забезпечення було розроблено, протестовано, проведено ітерації та розгорнуто? Був ще один аспект SDLC, на якому не було зосереджено увагу: експлуатація та обслуговування [9]. Це стосується циклу підтримки роботи програмного забезпечення, визначення необхідних покращень (за допомогою функцій постійного моніторингу та спостереження) та впровадження цих покращень. Зустрічайте DevOps.

Agile та DevOps. Так само, як недоліки моделі Waterfall надихнули Agile-методологію, недоліки Agile-методології також надихнули рух DevOps. Хоча ці методології часто використовуються як взаємозамінні, вони не є однаковими [19]. Додатковий фактор, який привносить DevOps, пояснюється його назвою. DevOps стосується об'єднання операцій розробки та ІТ-операцій.

До початку руху DevOps, ІТ-фахівці/фахівці з операційної діяльності зосереджувалися на моніторингу та підтримці різноманітних речей, таких як час безвідмовної роботи, безпека, мережа, інтеграції, надійність та відповідність вимогам [10]. Ці функції вимагають чіткої комунікації та міжфункціонального тестування, коли йдеться про впровадження змін коду або нових функцій до існуючого програмного продукту/системи, щоб гарантувати, що цілісність, функціональність та масштабованість продукту не будуть порушені.

Відсутність такої співпраці між командами розробників та операційними командами призвела до різних непередбачуваних наслідків (вищезгаданих недоліків Agile): створення окремих програмних компонентів, які не могли інтегруватися один з одним, не могли розширюватися, вносили критичні зміни

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

до існуючого програмного забезпечення тощо. Цих речей можна було б уникнути здебільшого (якщо не повністю), маючи ще тісний цикл зворотного зв'язку та структуру співпраці між командами розробників та операційними командами.

DevOps не обов'язково є заміною Agile, а радше іншим фреймворком, який працює разом з методологією Agile. DevOps може дещо погіршити швидкість, компактність та загальну «гнучкість» Agile, але водночас забезпечуючи плавнішу міжфункціональну роботу, збільшуючи частоту та якість комунікації, підтримуючи автоматизацію на вищому рівні та максимізуючи ефективність наскільки це можливо. Додавання DevOps до рівняння вимагає культурних змін, акценту на міжфункціональній співпраці та ролях, а також підвищення загальної оперативності реагування в процесі розробки [17].

DevSecOps та SSDLC. З огляду на те, що фокус галузі зміщується зі швидкої розробки та розгортання на врахування безпеки

наслідки та бажання запобігти використанню можливостей вразливостей безпеки, було проведено багато досліджень, щоб розпочати офіційне впровадження кібербезпеки до SDLC.

Ідея полягала в тому, щоб запровадити безпеку раніше в SDLC. Додавання взаємодії з безпекою в кінці циклу виявилось складним, оскільки всі основні інфраструктурні та проектні рішення вже були прийняті. Якщо були виявлені будь-які серйозні проблеми безпеки, внесення цих змін до релізу ставало дорогим і складним.

Ця ідея «зсуву безпеки ліворуч» у SDLC спонукала до створення - Безпечного життєвого циклу розробки програмного забезпечення (SSDLC) або Безпечного життєвого циклу розробки (SDL).

Разом із цим новим визначенням процесу також відбулася еволюція DevOps у DevSecOps [11]. Це стосується впровадження принципів безпеки та контролю в модель DevOps, яка знову інтегрує розробку та операційну роботу, зосереджуючись на культурних змінах для сприяння успіху нового процесу

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

розробки.

Хоча поява DevSecOps вирішила проблеми, що виникають через відсутність інклюзивного циклу зворотного зв'язку між розробниками, інженерами безпеки та IT/операціями, успішне впровадження DevSecOps у світ корпоративних технологій відбувається повільно та триває. Модель DevSecOps вимагає як культурних, так і інфраструктурних змін у кількох міжфункціональних організаціях у великій корпорації. Ці ж перешкоди передвіщають деякі з викликів, з якими ми неодмінно зіткнемося, впроваджуючи принципи UX-дизайну в модель DevSecOps/SDL [12]. Ми обговоримо ці виклики більш детально пізніше в цій роботі, коли обговорюватимемо запропоноване рішення для подолання розриву між UX-дизайном та безпекою.

UX-дослідження та дизайн. Перш ніж перейти до поєднання розриву між UX та безпекою, нам потрібно дослідити контекст UX-дизайну, UX-дослідження та чому вони важливі.

UX-дизайнери відповідають за те, щоб зробити продукт корисним, - ефективним та інтуїтивно зрозумілим для його користувачів. Обов'язки UX-дизайнера можуть відрізнятися залежно від проекту - вони можуть проводити дослідження користувачів та продукту, створювати персони для цільових користувачів продукту, визначати інформаційну архітектуру продукту, починати створювати макети каркасів для дизайну інтерфейсу користувача продукту, перетворювати каркаси на прототипи, додаючи варіанти взаємодії (наприклад, анімацію, параметри навігації тощо), і, нарешті, проводити тестування зручності використання.

Дослідження UX більш конкретно зосереджені на розумінні ефективних варіантів дизайну, проектуванні з урахуванням доступності, поведінці користувачів, цілях/потребах/больових точках користувачів тощо. Це може включати інтерв'ювання користувачів та зацікавлених сторін, проведення опитувань, проведення фокус-груп або проведення конкурентного аналізу з аналогічними продуктами на ринку [13].

Дослідження та UX-дизайн надзвичайно важливі для визначення успіху

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

програмного продукту. Якщо у вас є функціональний продукт, який надзвичайно складний у використанні, то ваша база користувачів скоротиться та постраждає. До зниження ефективної зручності використання та дизайну додається створення обмежувальних/заборонних функцій у програмному продукті. Зокрема, засобів контролю безпеки. Саме тут вступає в гру поняття «зручність використання проти безпеки».

Зручність використання та безпека. Вивчення UX-комп'ютера зазнало розвитку, і дослідники почали досліджувати, як ми можемо підтримувати зручність використання програмного забезпечення протягом життєвого циклу розробки. Деякі приклади того, як зручність використання та безпека можуть конфліктувати, включають: методи багатфакторної автентифікації, політики паролів та шифрування.

Користувачі скаржилися на необхідність увімкнути двофакторну автентифікацію (2FA) або багатфакторну автентифікацію (MFA) для входу в певні облікові записи. Цитуючи позов проти Apple за примусове використання 2FA в облікових записах Apple, «мільйони . споживачів по всій країні зазнавали та продовжують зазнавати шкоди. через втручання у використання їхніх особистих пристроїв та марнування особистого часу на використання додаткового часу для простого входу» [18]. Хоча 2FA додає додаткову безпеку перевірки облікових записів користувачів, процес має бути розроблений таким чином, щоб забезпечити найменший опір для користувачів.

Аналогічно, політики паролів відомі як проблемна зона для реєстрації нових користувачів на певних платформах. Політики паролів, які мають забагато обмежень, щоб нібито забезпечити надійність пароля користувача, можуть виявитися неприємними для тих, хто намагається створити новий обліковий запис [6]. Іноді компанії можуть вважати, що більше обмежень на створення пароля призводять до сильніших паролів, але це не завжди так. Політики паролів, такі як максимальна довжина символів, обов'язкове використання певної кількості спеціальних символів або непослідовних чисел, не завжди можуть призводити до найвищої ентропії пароля. Занадто велика кількість невдалих

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

спроб створити «прийнятний» пароль, це зниження зручності використання може вплинути на зростання кількості користувачів платформи.

Шифрування, спосіб захисту даних від неавторизованих сторін, є ще одним важливим заходом безпеки. Pretty Good Privacy (PGP) – це спеціальна програма шифрування, яка може використовуватися для підписання, шифрування або розшифрування даних [14]. Хоча PGP є (можливо, однією з найвідоміших) технологій, що використовуються для шифрованого зв'язку, зручність використання програмного забезпечення настільки низька, що його винахідник -навіть не використовує його [8] .

Безпека за замовчуванням/за замовчуванням часто пропонується як рішення цієї проблеми. Безпека за замовчуванням передбачає, що інженери з безпеки беруть участь у розробці продукту. Роблячи це, метою є захист продукту та наявність безпечних режимів за замовчуванням/резервування для подальшого захисту користувачів [11] . Інші йдуть далі, стверджуючи, що нам «просто» потрібно включити безпеку як на етапі проектування програмного забезпечення, так і на етапі UX-дизайну в процесі розробки, причому ідеальний стан – це коли UX-дизайнери ознайомлені з «основами безпеки та автентифікації» [10] .

Легше сказати, ніж зробити. Люди, які працюють у сфері програмного забезпечення, безпеки та дизайну, спеціалізуються у відповідних галузях. Важко надати кожному з цих окремих учасників достатньо ресурсів, навчання та часу, щоб вивчити всі тонкощі всіх трьох галузей, навіть якщо це обмежується лише основами. Значна частина літератури свідчить про те, що нам потрібно провести більше досліджень у цій галузі, включаючи системні дослідження картування та огляди літератури, щоб повністю зрозуміти, як ми можемо підійти до вирішення конкуруючих факторів UX-дизайну та гнучкої розробки, а також безпеки [12] . Ключовий висновок з поточної роботи в цій галузі підтверджує той факт, що UX-дизайнери повинні бути інтегровані в SDL, що спонукає до чергового інфраструктурного та культурного переходу від ідеї «зручність використання проти безпеки» до «зручність використання та безпека». Це допоможе вирішити проблеми, з якими стикаються як домашні, так і організаційні користувачі [4, 15]

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

. Запропоноване в цій роботі рішення має на меті надати фактичні деталі того, як можна реалізувати таку річ, щоб подолати цей розрив між дизайном, програмним забезпеченням та безпекою.

1.3 Аналіз сучасних методологій розробки та їх обмеження щодо швидкого зворотного зв'язку

Відсутність UX у робочому процесі DevSecOps не була головною темою. Досі основним завданням DevSecOps було формалізація та зміцнення циклу зворотного зв'язку між програмним забезпеченням та безпекою. Це було досліджено та запропоновано як перехід від DevOps [2] до DevSecOps та SSDLC/SDL [16].

Цілком логічно, що введення ще однієї сторони в цикл зворотного зв'язку має бути вирішене аналогічно переходу від DevOps до DevSecOps.

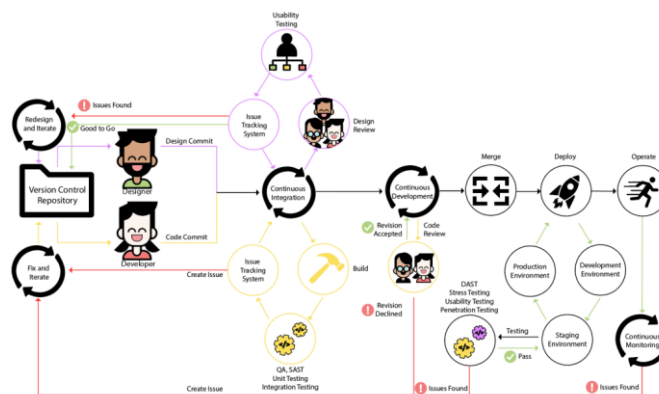


Рисунок 1.1 - Робочі процеси CI/CD

Адаптація SSDLC до TDLC

Те, що ми пропонуємо тут, – це ще одна адаптація SDL та моделі DevSecOps для гнучкої розробки. Ми хочемо включити UX-дизайнерів та дослідників до формалізованої моделі, яку ми пропонуємо як життєвий цикл розробки технологій (TDLC).

У цьому підрозділі роботи буде зосереджено увагу на визначенні того, як

процеси UX-дизайну та дослідження будуть інтегровані в модель DevSecOps, а також на розробці фреймворку CI/CD (автоматизованого набору інструментів для розробки) для забезпечення дотримання TDLC.

Модель «Подвійний діамант». Ми вже розглянули поточну модель DevSecOps, яка ілюструє цикл зворотного зв'язку між програмним забезпеченням, безпекою та IT/операціями. Давайте розглянемо поточний робочий процес UX-дизайну.

Модель «Подвійний діамант» – це широко використовувана модель у процесі UX-дизайну, розроблена Британською радою з дизайну. Чотири ключові кроки, викладені в цій моделі:

1. Відкриття; 2.Визначення; 3. Розробка та 4. Реалізація.

Під час фази «Виявлення» проводиться UX-дослідження для визначення масштабу проекту та його проблемної області. Це дослідження потім аналізується та використовується для синтезу списку больових точок, які необхідно враховувати під час проектування продукту. Це використовується на фазі «Визначення» для повного визначення конкретних проблем, того, що потрібно покращити, та які запропоновані рішення. На фазі «Розробка» UX-дизайнери починають створювати ескізи та макетні схеми для дизайну/редизайну продукту. Вони надають їх у вигляді прототипів [15] .

Ліва половина кожного ромба представляє дивергентне мислення, а права половина кожного ромба представляє конвергентне мислення. Дизайнеру слід практикувати дивергентне мислення під час виконання таких завдань, як дослідження користувачів/продукту/ринку або під час розробки дизайну та макетів. Конвергентне мислення є критично важливим після кожної з дивергентних фаз, оскільки те, що було досліджено, потрібно скоротити та сфокусувати [16]. Якщо дизайнер належним чином дотримувався моделі «Подвійний ромб», він би розпочав процес із загального уявлення про постановку проблеми, визначив би конкретні проблемні області для вирішення на етапі проектування та завершив би процес із конкретними рішеннями.

Модель TDLC. Метою TDLC є об'єднання моделі DevSec-Ops з моделлю

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Double Diamond. Це допоможе нам включити процес UX-дизайну в гнучку модель розробки DevSecOps. Модель TDLC дотримується того ж циклічного стилю, що й модель DevSecOps, щоб показати ітеративний характер цього процесу. TDLC вводить ще одне коло в нескінченний цикл, де чотири фази моделі Double Diamond включені на початку. Фази проектування перетікають у фази розробки, потім у фази операцій, а потім назад до початку. Всі три частини нового нескінченного кільця охоплені безпекою, що відображає включення засобів контролю та інструментів безпеки протягом усього процесу.

Робочий процес TDLC. Існує кілька різних частин робочого процесу TDLC. Ми розіб'ємо їх на зрозумілі розділи, щоб краще проілюструвати кожен процес.

Контроль версій для всіх: проектування, планування та документування важливо підтримувати повну прозорість між усіма сторонами в рамках TDLC. Для цього ми хотіли б запропонувати дизайнерам використання системи контролю версій у робочому процесі.

Під час перевірки файлів у системі контролю версій, все має бути структуровано стандартизованим чином, щоб забезпечити порядок. Якщо для структури проекту завжди дотримується однакового формату, то міжфункціональні команди зможуть щоразу знаходити те, що їм потрібно. Це також полегшить будь-яку автоматизацію в майбутньому.

Дизайн, планування та документація. Робочий процес TDLC починається з фази дослідження UX-дизайнерів. Вони використовують зібрані дослідження для визначення конкретних проблем, їхнього обсягу та пропозиції ефективних дизайнерських рішень. Все це буде зафіксовано в документах, які мають бути рецензовані колегами-дизайнерами, технічним керівником відповідної команди розробників програмного забезпечення та відповідальним за безпеку, призначеним для цього проекту.

Проектні документи будуть передані до спільного репозиторію контролю версій [17]. Потім дизайнери працюватимуть над ітерацією своїх фактичних проектів та прототипів. Ці прототипи передаються команді(ам) розробників, а

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

доступ до них також надається команді безпеки. Щоб команда безпеки бачила всі проекти функцій інтерфейсу користувача, пов'язаних з безпекою, їх можна відстежувати за допомогою додаткового поля метаданих у системі відстеження проблем.

Після того, як прототипи дизайну будуть передані інженерним командам, інженери починають визначати формальні вимоги до самого програмного забезпечення. На цьому етапі вони створять документ з вимогами, спроектують архітектуру та напишуть технічний проектний документ (TDD). Вони працюватимуть разом з командою інженерів безпеки, яка допомагатиме з консультуванням з питань безпеки, вимогами безпеки, моделюванням загроз, дотриманням вимог та переглядом TDD на наявність потенційних проблем безпеки [18]. Ці технічні документи також будуть передані до спільного репозиторію контролю версій.

Безперервна інтеграція. Буде розроблена система безперервної інтеграції, яка допоможе спростити робочий процес подій для проектування, розробки, безпеки та операцій. Детальніше про це в наступному розділі.

Безперервна розробка. Код перейде в цикл перевірки коду. Якщо редакцію відхилено, розробнику доведеться повернутися до виправлення та повторення виявленої проблеми. Якщо редакцію прийнято, це спонукає до об'єднання коду з головним проектом.

Середовища розгортання. Щойно все готово до початку, проєкт можна розгорнути. Існує три різні середовища, в яких можна розгорнути проєкт. Існує середовище розробки (яке використовується розробниками під час створення проєкту), проміжне середовище (де відбувається тестування, наприклад, динамічний аналіз коду, стрес-тестування продуктивності, тестування на проникнення, тестування зручності використання) і, звичайно ж, виробниче середовище (використовується для остаточного випуску продукту цільовому клієнту). Якщо під час цих тестів виявляються будь-які проблеми, створюється відповідний запис про проблему, і дизайнери/розробники повертаються до іншої ітерації свого проєкту/коду.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Безперервний моніторинг. Щоб забезпечити належну роботу програмного забезпечення, необхідно мати систему безперервного моніторингу. В ідеалі це дозволить попередити відповідні сторони про виникнення будь-яких проблем з доступністю, безпекою, функціональністю тощо. Будь-які виявлені проблеми будуть відстежуватися та додаватися до списку завдань, які потребують покращення на наступній ітерації.

Впровадження TDLC як робочого середовища CI/CD. Оскільки ми хотіли б фактично впровадити робоче середовище CI/CD, щоб забезпечити використання цієї нової моделі TDLC, давайте розглянемо деякі логістичні аспекти та робочі процеси.

Як згадувалося, важливо об'єднати всі сторони, заохочуючи - використання спільної системи контролю версій. На рисунку 1 показано цей робочий процес.

Загальний робочий процес буде базуватися на використанні системи контролю версій для відстеження будь-яких змін [19]. Після внесення змін користувач (дизайнер, розробник або спеціаліст із безпеки) буде переміщений через процеси неперервної інтеграції (CI), процеси управління версіями (CD), а потім, зрештою, до операційних завдань, доки не повернеться до початку потоку.

Два основні варіанти використання для висвітлення робочого процесу:

Дизайнер: Затвердження дизайну; Огляд дизайну; Тестування зручності використання, Редизайн ТА Ітерація АБО Передача розробникам.

Розробник: Затвердження коду; Збірка; Тестування: контроль якості, статичний аналіз коду, модульне тестування, інтеграційне тестування, 4. Створення проблеми АБО прийняття/відхилення

Для успішного впровадження цього робочого середовища CI/CD для TDLC ми продумали, які технології та фреймворки можна було б використовувати.

Система контролю версій

Ми вирішили, що git буде найкращим варіантом для контролю версій, оскільки це стало галузевим стандартом. Subversion (svn) – ще один варіант, хоча

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

він дедалі менше використовується в технологічній галузі. Для цього робочого процесу CI/CD ми могли б використовувати набір інструментів, такий як GitLab, який підтримує git і також пропонує багато функцій CI/CD.

Для дизайнерів ми хотіли б інтегрувати щось на кшталт Versions від Symply [20]. Це інтерфейс користувача, який інтегрується з git спеціально для керування версіями графічних файлів та файлів прототипів/каркасів. Цей інструмент відображає відмінності між кожною версією файлів дизайну та підтримує багато плагінів із поширеними інструментами дизайну. Існує кілька способів взаємодії з git, і найпоширеніший спосіб зробити це – через командний рядок, але для дизайнерів, які не мали досвіду роботи з цим, цей інструмент зробить систему контролю версій доступнішою в цьому новому робочому процесі.

Інструменти дизайну

Для початкового впровадження ми хотіли б підтримувати такі поширені інструменти: Adobe Illustrator; . Adobe XD, . Sketch.

Система відстеження проблем

Стандартним інструментом, що використовується для відстеження проблем, є JIRA від Atlassian. Ми б використовували його для створення заявок на проблеми, щоб відстежувати будь-які завдання, які мають виконати залучені сторони в TDLC.

Сервер та система CI/CD

Як згадувалося, GitLab підтримує автоматизацію багатьох завдань CI/CD. GitLab пропонує GitLab Runners для автоматизованого керування збірками. Вони також пропонують функції автоматизованого тестування, розгортання та постійного моніторингу операційного програмного забезпечення після розгортання.

Альтернативи CI/CD

Якщо ми не будемо використовувати GitLab для кожного аспекту автоматизації CI/CD, деякі альтернативні технології, які варто розглянути, включають Apache Ant (збірка), Selenium (тестування) та Jenkins, Puppet або Chef

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

(розгортання).

Успішне впровадження моделі DevSecOps для подолання розриву між програмним забезпеченням та безпекою вже було (і залишається) складним завданням. Програмне забезпечення та безпека – це дві сфери, які мають власні робочі навантаження та пріоритети, що ускладнює визначення пріоритетів міжфункціональної роботи.

Основна проблема полягає в тому, що цей зсув вимагає більше, ніж просто переробки SDLC [21]. Існує багато рушійних факторів, включаючи - культурні зміни, стимулювання міжфункціональної роботи та етапів, а також загальне розуміння того, що вся спільна робота є важливою [9] . Знадобиться більше часу та більша підтримка з боку керівництва/керівництва вищого рівня в компаніях-розробниках програмного забезпечення, щоб дійсно запровадити це як щось, що потребує широкого впровадження в організаціях. Додавання ще одного елемента сюди є викликом, оскільки ми ще не належним чином впровадили рух DevSecOps SDL.

1.4 Висновок по розділу

У першому розділі було розглянуто теоретичні основи організації швидкого зворотного зв'язку в процесі розробки програмного забезпечення. Проаналізовано основні принципи та концепції швидкого зворотного зв'язку, що забезпечують оперативне виявлення проблем і вдосконалення програмних продуктів. Досліджено взаємодію користувачів із програмними системами як важливе джерело інформації для покращення функціональності та якості ПЗ. Також проведено аналіз сучасних методологій розробки та визначено їх обмеження щодо забезпечення швидкого циклу зворотного зв'язку. Отримані результати формують теоретичну основу для подальшого дослідження методів і технологій організації ефективного зворотного зв'язку.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 МЕТОДИ ТА ІНСТРУМЕНТИ ЗАБЕЗПЕЧЕННЯ ШВИДКОГО ЗВОРОТНОГО ЗВ'ЯЗКУ

2.1 Архітектурні та організаційні підходи до реалізації циклів зворотного зв'язку

У цьому підрозділі представлено глибоке дослідження циклів зворотного зв'язку з користувачами в гнучких організаціях. У ньому пояснюються основні принципи гнучкої розробки програмного забезпечення та їх важливість для гнучких, ітеративних процесів розробки. Було проаналізовано відповідну літературу, щоб підкреслити важливість циклів зворотного зв'язку з користувачами. Аналіз визначає проблеми, потенційні рішення та прогалини в дослідженнях, які не були достатньо розглянуті в сучасній літературі.

Принципи гнучкої розробки програмного забезпечення

Гнучка розробка програмного забезпечення (Agile) – це ітеративний та адаптивний підхід до розробки програмного забезпечення, що виник у відповідь на зростаючу потребу в більш гнучких процесах розробки. На відміну від традиційних лінійних моделей розробки, гнучка розробка наголошує на постійному вдосконаленні, швидких циклах зворотного зв'язку та здатності швидко реагувати на вимоги, що змінюються [22]. Однією з переваг гнучкої розробки програмного забезпечення є її здатність допомагати командам динамічно адаптуватися, зберігаючи при цьому сильну спрямованість на забезпечення високої цінності продукту.

У Agile-маніфесті, опублікованому у 2001 році групою досвідчених - розробників програмного забезпечення, визначено чотири основні цінності, які слугують основою для гнучких методологій. Ці цінності підкреслюють:

1. Люди та взаємодія щодо процесів та інструментів,
2. Працююче програмне забезпечення з повною документацією,
3. Співпраця з клієнтом щодо узгодження умов контракту,
4. Реагування на зміни за чітко визначеним планом.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці принципи підкреслюють важливість гнучкості, співпраці та клієнтоорієнтованої розробки, що робить гнучку методологію широко прийнятою в різних галузях.

Роль циклів зворотного зв'язку з користувачем у гнучкій розробці

Гнучка розробка програмного забезпечення зосереджена на зворотному зв'язку та змінах. У розробці програмного забезпечення короткі цикли оцінки та адаптації в поєднанні зі швидкими циклами зворотного зв'язку є важливими для вирішення проблеми непередбачуваного характеру галузі. Ці ітеративні процеси дозволяють гнучким методологіям адаптуватися до постійно мінливих вимог галузі. Цикл зворотного зв'язку – це постійний обмін між розробниками та кінцевими користувачами [23]. Він визначається як часовий проміжок від моменту, коли розробник пише рядок коду, до моменту його виконання та перевірки його функціональності кимось. Цей постійний обмін гарантує, що програмні продукти залишаються відповідними фактичним потребам та очікуванням користувачів, покращуючи зручність використання та функціональність. Інтегруючи цикли зворотного зв'язку в процес розробки, команди можуть виявляти та вирішувати проблеми на ранній стадії, зменшуючи ризик дорогої переробки та підвищуючи загальну якість програмного забезпечення.

Зворотній зв'язок можна збирати за допомогою бета-тестування, досліджень зручності використання або прямих опитувань користувачів. Крім того, регулярні спринт-огляди та ретроспективи забезпечують структуровану платформу для безперервного внеску користувачів. Такий підхід гарантує, що перспектива користувача враховується не лише на початку проекту, а й залишається пріоритетом протягом усієї розробки. Коротші цикли зворотного зв'язку підвищують гнучкість, дозволяючи швидше приймати рішення, сприяючи співпраці та пропагуючи культуру постійного вдосконалення. Дослідження гнучкої інженерії вимог (RE) також підкреслюють важливість циклів зворотного зв'язку. Стверджують, що «найбільш критичними практиками гнучкої інженерії вимог є ті, що підтримують ітеративну розробку з екстремними

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

вимогами та коротким циклом зворотного зв'язку», підкреслюючи їхню роль у забезпеченні гнучкості та адаптивності в програмних проектах.

Прикладом успішного впровадження циклів зворотного зв'язку є LEGO. Регулярно враховуючи відгуки зацікавлених сторін, компанія може швидко модифікувати свої продукти відповідно до потреб користувачів та забезпечити їх відповідність очікуванням клієнтів. Цей приклад демонструє, як ефективні цикли зворотного зв'язку не лише допомагають підтримувати високу якість продукції, але й підвищують задоволеність клієнтів. Зрештою, задоволення очікувань користувачів зміцнює лояльність клієнтів та надає організаціям конкурентну перевагу.

Таблиця 2.1

Проблеми циклу зворотного зв'язку з користувачем

Виклики (Challenges)	Пов'язані проблеми (Related Issues)	Виклики (Challenges)
Затримки та неефективність	Невідповідність продукту потребам користувачів, довгі цикли зворотного зв'язку, затримки в отриманні відгуків, «негнучкі» (non-agile) клієнти.	Затримки та неефективність
Брак залучення користувачів	Складність у забезпеченні співпраці з клієнтом, нечіткий або розмитий зворотний зв'язок, географічні перешкоди, брак комп'ютерних інструментів для підтримки колаборації.	Брак залучення користувачів
Навантаження та брак належних інструментів	Неадекватні методи та інструменти валідації, зниження якості продукту, недостатні механізми аналізу даних, погане розуміння використання функцій, збільшення навантаження через безперервний зворотний зв'язок.	Навантаження та брак належних інструментів
Комунікаційні бар'єри	Непорозуміння, неузгодженість цілей проєкту, зниження залученості стейкхолдерів, брак мотивації користувачів та негативне ставлення.	Комунікаційні бар'єри

Проблеми впровадження циклів зворотного зв'язку

Хоча цикли зворотного зв'язку з користувачами є основоположними для гнучкої розробки програмного забезпечення, їх реалізація створює кілька проблем. Аналіз відповідної літератури виявив кілька повторюваних ключових

проблем у таких сферах, як комунікація, залучення користувачів, своєчасний зворотний зв'язок та інтеграція зворотного зв'язку в цикл розробки. Ці проблеми узагальнено в таблиці 2.1 та детальніше описані в наступному розділі.

Затримки та неефективність

Відсутність або затримка зворотного зв'язку з користувачами може призвести до негативних наслідків, таких як невідповідність між продуктом та потребами користувачів. Такі проблеми, як тривалі цикли зворотного зв'язку, згадуються в багатьох дослідженнях, таких як дослідження, що стосуються різних аспектів процесу гнучкої розробки. Назвали кілька причин тривалих циклів зворотного зв'язку, включаючи затримки через комунікацію із зацікавленими сторонами поза організацією, клієнтів, які не використовують гнучкі методи розробки, та роботу з розгалуженою мережею зацікавлених сторін. Наголошують на важливості підтримки швидких та ефективних циклів зворотного зв'язку для підтримки конкурентоспроможності.

Відсутність залучення користувачів

Ще однією проблемою, що обговорюється в літературі, є відсутність залучення користувачів. Зазначено, що забезпечення достатньої співпраці з клієнтами часто є складним у великих організаціях. Через обмежену участь та нечіткий зворотний зв'язок з клієнтами, командам важко повністю зрозуміти цінність для клієнта та перетворити її на дієві вимоги. Аналогічно, у своєму дослідженні за участю фахівців з гнучкої розробки з Нової Зеландії та Індії виявили, що багато учасників вважають обмежену участь клієнтів однією з найбільших проблем у гнучкій розробці. Ця проблема ще більш помітна в географічно розподілених проектах, де координація участі клієнтів є особливо складною.

Робоче навантаження та брак належних інструментів

Визначено ключову проблему у зворотному зв'язку з клієнтами: відсутність відповідних методів та інструментів валідації, а також проблеми з прототипуванням. Ці недоліки зрештою можуть призвести до зниження якості продукту. Наголошено на проблемі відсутності ефективних механізмів аналізу

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

даних клієнтів. Крім того, вони визначили недостатнє розуміння використання функцій та нездатність проактивно використовувати аналітику клієнтів як ключові проблеми у зборі та використанні даних. Відсутність комп'ютерних інструментів, що підтримують залучення клієнтів [24]. Зазначено, що взаємодія з клієнтом є важливою в гнучкій розробці, оскільки вона генерує цінний зворотний зв'язок. Однак це також збільшує робоче навантаження команди, особливо тому, що гнучка розробка заохочує постійні зміни, вимагаючи частих коригувань та ітерацій, що додає складності та зусиль.

Комунікаційні бар'єри

Ще однією проблемою є комунікація між користувачами та командою розробників. Ефективна комунікація має вирішальне значення для врахування відгуків користувачів у гнучкій розробці. Погана комунікація може призвести до непорозумінь, невідповідності цілям проекту та зниження залучення зацікавлених сторін. Крім того, недостатня мотивація користувачів та негативне ставлення перешкоджають ефективній участі користувачів.

Існуючі рішення для впровадження циклів зворотного зв'язку

Літературна довідка також включає дослідження, що охоплюють різні рішення та стратегії вирішення проблем циклу зворотного зв'язку. У цьому розділі розглядаються різні категорії рішень, представлені в літературі, включаючи інструменти, показники, вдосконалені процеси та структуру команди.

Покращення процесів та структури команди

Виявлено розрив у комунікації між компаніями та їхніми клієнтами. Їхнє рішення зосереджено на покращенні процесів, пов'язаних з розробкою, та оптимізації структури команди. Це включає зменшення кількості передач завдань та використання міжфункціональних команд. Вони припускають, що покращена візуалізація вимог може покращити комунікацію та, згодом, покращити здатність клієнта надавати відповідний зворотний зв'язок [25]. Також рекомендується працювати з міжфункціональними командами для зміцнення співпраці, що може призвести до нового розуміння так званих «больових точок

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

клієнтів». У їхньому дослідженні обговорювалися різні способи інтеграції зворотного зв'язку від клієнтів та команди в процес гнучкої розробки, включаючи послідовні ретроспективи та сеанси тестування користувачами. Ці дослідження не зосереджені безпосередньо на вирішенні проблем, пов'язаних із циклом зворотного зв'язку, і включають рішення, пов'язані з циклом зворотного зв'язку, лише як невелику частину більшої картини, тому потрібні додаткові дослідження.

Інструменти та показники

Приділено багато уваги використанню інструментів, платформ та аналітики. Вони згадали інструменти для безперервної інтеграції та безперервного розгортання (CI/CD), оскільки вони спрощують процес розробки та дозволяють командам швидше впроваджувати зміни, скорочуючи час, необхідний для обробки відгуків клієнтів. Вони також запропонували використовувати інструменти для збору та аналізу задоволеності клієнтів. Цей тип інструментів полегшує клієнтам надання інформації та стандартизує процес оцінювання для компаній [26]. У їхньому дослідженні також розглядаються соціальні мережі як джерело відгуків клієнтів. Збір даних про відгуки клієнтів та показників ефективності може допомогти визначити тенденції та області для покращення.

Запропоновано модель розробки на основі гіпотез та експериментів, що базуються на даних (HYPERX), для вирішення проблеми «розкритого циклу», де рішення щодо вимог спираються на думки, а не на дані. Їхнє тематичне дослідження показало, що модель успішно допомогла компаніям покращити інтеграцію зворотного зв'язку та приймати рішення на основі даних. Модель сприяє вирішенню проблем зворотного зв'язку, але необхідно знайти інші рішення, які вирішують різні типи проблем, пов'язаних зі зворотним зв'язком.

2.2 Методи збору та обробки даних зворотного зв'язку

У цьому підрозділі представлено дизайн дослідження, стратегію та

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

методи, що використовувалися для збору та аналізу даних у цьому дослідженні. Це включає опис обраного методу дослідження, процесу інтерв'ювання для збору даних, створення посібника з інтерв'ювання та того, як дані аналізувалися за допомогою тематичного аналізу.

Дизайн та стратегія дослідження

Обрана стратегія дослідження для цього дослідження – якісне опитування. Це відповідний підхід, оскільки він спрямований на дослідження та розуміння рішень для впровадження циклів зворотного зв'язку з користувачами від практиків у гнучких умовах. Стратегія якісного опитування дає детальний огляд теми для збору даних та виявлення закономірностей. Простими словами, ця стратегія полягає у пошуку необхідної інформації для відповіді на дослідницьке питання. Це корисно, коли дослідник знає, яка інформація потрібна для вирішення конкретної проблеми, а не займається дуже складними або більш абстрактними питаннями.

Наші два дослідницькі питання спрямовані на пошук існуючих рішень для реалізації циклів зворотного зв'язку з користувачами та їх логічну категоризацію. Дослідницькі питання досліджують конкретні проблеми, які є більш конкретними. Оскільки стратегія якісного опитування схильна збирати дані для аналізу з реального світу, щоб відповісти на цей тип дослідницької проблеми, вона дуже добре відповідає меті нашого дослідження. Стратегія підтримує вивчення різних точок зору практиків, що забезпечує унікальне розуміння та комплексний аналіз зібраних даних.

Використання якісних даних дозволяє зрозуміти різний досвід та визначити теми, які можна використовувати для відповіді на дослідницькі питання та надання відповідних рекомендацій. Якісні дані в цьому дослідженні були зібрані за допомогою інтерв'ю і тому представлені у форматі записаної розмови. Було проведено аналіз відповідної літератури, щоб отримати достатні базові знання з обраної нами теми.

Альтернативна дослідницька стратегія

Ще однією дослідницькою стратегією, яку обговорювали для

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

використання для відповіді на наші дослідницькі питання, було тематичне дослідження. Тематичне дослідження зосереджується «лише на одному прикладі речі, що підлягає дослідженню». Воно відоме тим, що пропонує глибоке розуміння конкретного контексту. Ми розглянули цей підхід, оскільки він є якісним і дозволяє глибоко зазірнути в одну компанію, потенційно розкриваючи деталі, які могли б бути пропущені в кількісному підході. Порівняно з традиційним (кількісним) опитуванням, тематичне дослідження пропонує більшу глибину та є цінним якісним методом дослідження [27]. Ми вирішили не використовувати тематичне дослідження, оскільки не хотіли обмежувати обсяг нашого дослідження однією компанією. Натомість ми обрали якісне опитування з використанням напівструктурованих інтерв'ю. Ця стратегія дозволила нам отримати таку ж багату, глибоку інформацію, включаючи точки зору різних організацій та ролей. Цей ширший діапазон вхідних даних допоміг нам краще зрозуміти різноманітність практик та проблем у різних контекстах.

Метод збору даних

Методом збору даних, обраним для цього дослідження, є інтерв'ю. Цей формат дозволяє глибоко дослідити тему, надаючи набагато більше свободи, ніж анкетування. Він надає можливість ставити конкретні запитання про думки, досвід та практику людей, що особливо корисно під час дослідження процесів та проблем впровадження.

Оскільки дослідницьке питання стосується того, як рішення для покращення циклів зворотного зв'язку з користувачами впроваджуються на практиці та як пом'якшуються проблеми, тема є одночасно складною та залежить від контексту. Інтерв'ю дозволяють учасникам поміркувати над своїм особистим та організаційним досвідом у їхніми власними словами, надаючи багаті та нюансовані висновки, які було б важко отримати за допомогою суто кількісних методів.

В індивідуальному форматі було використано напівструктурований підхід до інтерв'ю. Така структура інтерв'ю надає свободу для глибокого дослідження цікавих тем, водночас маючи достатню структуру, щоб гарантувати

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

відповіді на дослідницькі питання та забезпечити кращу узагальнюваність. Формат добре збалансовує структуру та гнучкість. Інтерв'ю проводилися напівформально, намагаючись знайти хороший рівень комфорту для заохочення вільної розмови, зберігаючи при цьому контроль над напрямком інтерв'ю.

Учасників інтерв'ю було відібрано за допомогою цілеспрямованої стратегії вибірки. За цим методом люди обираються на основі їхньої релевантності та знань. Наших учасників для інтерв'ю було знайдено онлайн через професійні мережі, такі як LinkedIn, та професійно профільовані групи на Facebook. Їх було відібрано на основі їхнього досвіду роботи з циклами зворотного зв'язку в гнучкій розробці програмного забезпечення, щоб забезпечити релевантність зібраних даних та їхню реальну інформацію. Щоб запобігти упередженості та врахувати різні точки зору, ми прагнули створити збалансовану групу, яка б включала різні ролі, такі як scrum-майстри, власники продуктів та розробники.

Альтернативним методом, що розглядався, була вибірка для зручності, коли відбираються легкодоступні люди, такі як колеги. Цей метод не був обраний, оскільки він може призвести до обмеженого кола перспектив. Учасники з однаковим досвідом або компанією можуть мати схожий досвід та точки зору, що може не забезпечити необхідної різноманітності для дослідження [28]. Випадкова вибірка також була виключена, оскільки ми проводимо якісне дослідження, яке спирається на учасників, які можуть надати детальні та релевантні висновки. Випадково відібрані особи можуть не мати конкретного досвіду роботи з циклами зворотного зв'язку в гнучких процесах, що є важливим для нашого дослідження, що зменшує ймовірність того, що вони нададуть необхідну глибоку інформацію.

Посібник зі співбесіди

Посібник з інтерв'ю було створено відповідно до структури, запропонованої у її книзі «Основний посібник з проведення дослідницького проекту» [29]. Оскільки було використано напівструктурований підхід, було написано перший варіант тем і питань для включення до інтерв'ю. Наступним

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

кроком було переглянути варіант, щоб оцінити ясність питань та актуальність тем. Потім питання були відредаговані та розміщені логічно, хоча порядок не потребував суворого дотримання. Нарешті, було написано сценарій для вступу було написано та включено до посібника. Кроки повторювалися, доки посібник із інтерв'ю не був завершений, з питаннями, що охоплювали всі відповідні теми.

Проведення та транскрипція інтерв'ю

Інтерв'ю проводилися онлайн, індивідуально, за допомогою Zoom, завдяки його простоті використання та функції запису. Кожне інтерв'ю тривало від 30 до 50 хвилин і починалося зі вступу до дослідження, наших цілей та інформації про анонімізацію та конфіденційність. Ми переконалися, що маємо дозвіл на запис інтерв'ю, і дозволили кожному учаснику поставити запитання перед початком інтерв'ю. Ми використовували сценарій, щоб переконатися, що вся необхідна інформація була включена. Окрім інформування учасника, вступ використовувався для створення розслабленої та комфортної атмосфери, задаючи тон інтерв'ю.

Після завершення кожного інтерв'ю запис оброблявся та завантажувався через Zoom. Потім його обробляла програма Seshat Audio Transcriber, інструмент транскрипції кафедри комп'ютерних та системних наук (DSV) Стокгольмського університету. Інструмент відповідає Загальному регламенту захисту даних (GDPR) [30]. Потім транскрипт було оброблено вручну, щоб видалити слова-заповнювачі та покращити читабельність тексту без зміни тону та значення змісту. Помилки та граматичні помилки також були виправлені вручну, де це було необхідно, а всі імена осіб та організацій були виключені або змінені для збереження анонімності. Учасники мали можливість перевірити остаточний транскрипт інтерв'ю на точність.

Альтернативний метод збору даних

Вибираючи метод збору даних, ми враховували такі фактори, як тип питань, обсяг та часові рамки, щоб найкраще відповісти на наші дослідницькі питання. Перш ніж зупинитися на інтерв'ю, ми розглядали використання анкети з відкритими питаннями. Враховуючи наш час та можливості, перевагою

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

використання анкети було б те, що ми потенційно могли б охопити більшу кількість людей. Інтерв'ю займають більше часу, тому відповіді учасників на питання без нашої участі заощадили б нам час у процесі збору та зняли б тягар транскрибування аудіо. Недоліком підходу з анкетуванням є те, що важче контролювати глибину та точність відповідей. Під час інтерв'ю інтерв'юер може досліджувати та ставити додаткові питання, щоб глибше зануритися в цікаві аспекти, що може покращити якість матеріалу та результати. Тому, на користь отримання більш насичених даних для аналізу, ми вирішили використовувати інтерв'ю для збору даних.

Метод аналізу даних

Тематичний аналіз було обрано, оскільки він добре відповідає другому дослідницькому питанню. Це питання є дослідницьким і спрямоване на розуміння практичних процесів і досвіду з точки зору фахівців, які працюють у реальних гнучких середовищах. Ці питання використовують гнучкий, але структурований метод, який дозволяє виявляти закономірності у відповідях учасників, не покладаючись на заздалегідь визначені аналітичні категорії. Тематичний аналіз підтримує це, залишаючись ґрунтованим на даних. Використовуючи індуктивний підхід, аналіз зосереджується як на рішеннях, так і на спільних перешкодах, чітко описаних учасниками, а також враховує приховані закономірності, які можуть зливатися [31]. Ці характеристики роблять його особливо придатним для відповіді на питання, яке зосереджено на практиці впровадження, а не на абстрактній теорії.

Дані в цьому дослідженні були проаналізовані за допомогою шестиетапного процесу тематичного аналізу. Метод був обраний завдяки його здатності ефективно виявляти закономірності та тенденції в якісних даних. Це гнучкий підхід, який вимагає глибокого занурення в дані, що робить його придатним для такого поглибленого аналізу. Індуктивний підхід був застосований відповідно до рекомендацій Брауна та Кларка, що означає, що кодування даних не спиралося на заздалегідь визначені категорії, а визначалося самими даними. Такий підхід підвищує достовірність та прозорість результатів і

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

гарантує, що висновки ґрунтуються на відповідях учасників, оскільки зміст спрямовує аналіз.

Ми розпочали процес аналізу з ознайомлення з даними [32]. Оскільки інтерв'ю ми проводили окремо, ми вирішили почати з ознайомлення з інтерв'ю, в яких не брали участі, щоб отримати краще уявлення про дані. Під час читання матеріалу ми записували початкові коди як коментарі в Документах Google. Потім ми використовували Скрипти Google Apps, щоб імпортувати коментарі та позначені уривки в Таблицю Google. Коментарі були очищені та звужені до потенційних тем. Потенційні теми були потім переглянуті та обговорені, щоб виділити остаточні теми, які слід було визначити, назвати та включити до звіту.

Альтернативний метод аналізу даних

Обґрунтовану теорію розглядали як альтернативу тематичному аналізу, оскільки це метод аналізу, придатний для якісних даних у невеликих дослідницьких дослідженнях [33]. Основою обґрунтованої теорії є практичний підхід та «відкритий розум», як висловився Денскомб, «дослідник вирушає у подорож відкриттів». Результатом аналізу обґрунтованої теорії має бути нова теоретична основа, що ґрунтується на зібраних даних. Однак це досить складний підхід до аналізу даних, що є однією з причин, чому вибір зрештою припав на тематичний аналіз. Невеликий розмір дослідження не дозволяє робити - узагальнення та формувати нові теорії, тому суто дослідницький характер тематичного аналізу більше підходить для наших цілей.

ІТ-інструменти

У цьому дослідженні для спілкування з учасниками інтерв'ю використовувалися електронна пошта, WhatsApp та LinkedIn, тоді як для проведення інтерв'ю було обрано Zoom. Роботу було написано за допомогою Overleaf, а для конспектування та тематичного аналізу використовувалися Google Docs та Google Sheets. Джерела зберігалися в Zotero, щоб забезпечити спільний доступ та кращу організацію. Розповсюдження форм згоди здійснювалося через Survey&Report, веб-інструмент для опитувань, наданий Стокгольмським університетом.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, інструменти штучного інтелекту (ШІ) підтримували різні етапи дослідницького процесу [34]. Для транскрибування аудіозаписів було використано Seshat Audio Transcriber . Крім того, ChatGPT допоміг порівняти чернетки з необхідними критеріями ясності та релевантності, а також виявити та виправити граматичні помилки. Вихідні дані ШІ керувалися такими підказками, як «Порівняйте цей текст з наступними вимогами» та «Визначте граматичні помилки в наступному тексті». Більше того, DeepL Write і граматики були використані для покращення читабельності письмових текстів шляхом пропонування покращень структури речень та плавності. Ці інструменти служили для вдосконалення кінцевого документа , але не застосовувалися для створення оригінального контенту. Усі інструменти штучного інтелекту використовувалися відповідально, відповідно до політики ШІ DSV .

2.3 Інструменти аналітики та моніторингу програмного забезпечення

У цьому підрозділі описано інструмент аналітики програмного забезпечення Q-Rapids, інтегрований у процес розробки Softeam як частину цього дослідження. Для повного розуміння функціональності вибраного інструменту, у першому підрозділі описано, як дані агрегуються інструментом у трирівневій моделі якості.



Рисунок 2.2 - Панель інструментів Q-Rapids (візуалізація стратегічних показників)

А. Модель якості Q-Rapids

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Оцінка якості, що надається інструментом Q-Rapids, описана в наступному підрозділі, відповідає трирівневій структурі, що накладається моделлю якості Q-Rapids [6]. Три рівні визначають показники якості, фактори якості та стратегічні показники:

- *Метрики якості* – це низькорівневі показники, що вимірюють певну характеристику (наприклад, *коефіцієнт коментарів*), оцінену на основі необроблених даних (наприклад, *проблем*), що зберігаються в інструментах джерел даних (наприклад, *JIRA*).
- Метрики якості об'єднуються у *фактори якості* (наприклад, *якість коду, швидкість виконання завдань*), які визначають важливі поняття стосовно якості.
- Фактори якості об'єднуються для обчислення елементів моделі вищого рівня, які називаються *стратегічними показниками* (наприклад, *якість продукту, ефективність процесу*). Стратегічні показники дозволяють відобразити рівень досягнення тих аспектів, які компанії вважають релевантними для своїх програмних продуктів та процесів прийняття рішень. Таблиця 2.2 містить визначення двох факторів якості, що використовуються Softeam для оцінки стратегічного показника *якості продукту*.

Таблиця 2.2

Показники якості продукції

Назва (QM Name)	Визначення (Definition)
Коефіцієнт коментарів (Comments Ratio)	Відсоток файлів, що знаходяться у визначеному діапазоні щільності коментарів.
Складність (Complexity)	Відсоток файлів, що не перевищують визначену середню цикломатичну складність (базується на кількості шляхів через код).
Щільність дублювання (Duplication Density)	Відсоток файлів, що знаходяться у визначеному діапазоні дубльованих блоків рядків коду.

Інструмент аналітики програмного забезпечення Q-Rapids

Інструмент Q-Rapids було розроблено в результаті європейського дослідницького проекту Q-Rapids. Головною метою інструменту Q-Rapids є

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечення безперервної оцінки якості для підтримки осіб, які приймають рішення, в управлінні якістю в гнучкій розробці програмного забезпечення. Крім того, інструмент надає конкретні функції для підтримки процесів прийняття рішень, пов'язаних з управлінням QR. Далі ми опишемо його основні функції [9].

Оцінка якості. Інструмент Q-Rapids збирає дані з різномірних зовнішніх джерел даних, таких як статичний аналіз коду (наприклад, SonarQube), інструменти безперервної інтеграції (наприклад, Jenkins), репозиторії коду (наприклад, SVN, Git, GitLab), інструменти відстеження проблем (наприклад, Redmine, GitLab, JIRA, Mantis) та журнали використання. Ці дані агрегуються в стратегічні показники відповідно до моделі якості, описаної в попередньому - підрозділі, дані представлені особам, що приймають рішення, через інформаційну панель.

Прогнози та аналіз «що, якщо». Ці функції допомагають особам, що приймають рішення, у прийнятті рішень щодо розвитку продукту. Ряд доступних методів прогнозування (наприклад, ARIMA, ETS, нейронні мережі) прогнозують еволюцію стратегічних показників з часом, враховуючи тенденції джерел даних [10] [11]. Ця функціональність дає можливість реагувати на потенційні проблеми, пов'язані з конкретними аспектами продукту [35]. Коли прогнозування вказує на можливі майбутні проблеми з якістю, аналіз «що, якщо» може бути використаний для оцінки варіантів покращення, показуючи, як ці зміни вплинуть на стратегічні показники, тобто яка оцінка SI реагуватиме на конкретні фактори або значення показників. Фактично, аналіз «що, якщо» може бути використаний як незалежний метод у будь-який момент процесу оцінки якості.

Напіваавтоматичне створення вимог до якості. Коли оцінка якості нижча за деякі визначені користувачем порогові значення для одного або кількох елементів моделі якості, генерується сповіщення про якість. Інструмент Q-Rapids пропонує потенційні QR-коди, пов'язані з цими сповіщеннями, та візуалізує вплив на стратегічні показники впровадження запропонованого QR. Для того, щоб запропонувати QR, інструмент Q-Rapids містить каталог шаблонів QR, який

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

організація повинна вдосконалити для врахування аспектів якості, включених до її моделі якості [12].

Описується, як процес QFL був впроваджений у Softeam. Ця реалізація описується з точки зору того, як Softeam інтегрував QFL та пов'язані з ним інструменти (аналітику програмного забезпечення та планування проектів) у свій процес розробки, а також як ці інструменти використовувалися на етапах QFL.

На рис. 2.3 показано дії, що охоплюються процесом QFL, який охоплює життєвий цикл управління QR-кодами, а саме: виявлення QR-кодів, включаючи визначення каталогу QR-кодів та створення екземплярів QR-кодів у контексті проекту; планування QR-кодів; та розробка QR-кодів та їх перевірка як частина моніторингу зворотного зв'язку щодо QR-кодів.

А. QFL інтегровано в процес розробки Softeam

У контексті процесу розробки Softeam, QFL впливає на фазу розробки (*функції плану, впровадження спринту та оцінка спринту*) та фазу передрелізної валідації (*діяльність інтеграції та валідації*). Процес розробки програмного забезпечення Softeam, що інтегрує процес QFL та інструменти, що підтримують кожну фазу (інструмент Q-Rapids та Open Project). На рисунку також показано інструменти джерел даних, що використовуються Softeam, які створюють дані, що використовуються інструментом Q-Rapids для оцінки якості.

Під час *впровадження спринту* дані розробки генеруються з кількох інструментів (Redmine, Mantis, OpenProject, SVN, Mantis, Jenkins та Modelio Testing System), а також збираються та аналізуються інструментом Q-Rapids для оцінки якості продукту та процесу. Дані використовуються для обчислення показників якості, факторів якості та стратегічних показників на основі моделі якості Softeam.

Оцінка показників якості (моніторинг QR-зворотного зв'язку) використовується *інженерами з якості та керівниками проектів у спринті*. *Оцінювальна діяльність* для виявлення проблем якості у наданих артефактах (виявлення QR-відповідей), які експортуються з інструменту Q-Rapids до

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

інструменту OpenProject. Також аналізується процес розробки, приділяючи особливу увагу діяльності, спрямованій на вирішення проблем якості.

Команда використовує аналітичні звіти, отримані під час оцінювання спринту, разом із питаннями якості, які вже включені до OpenProject, для планування наступної ітерації спринту (*діяльність «Планування функцій»*). Керівник *проекту* вирішує прийняти (чи ні) питання якості, а для прийнятих – найкращий спосіб їх реалізації, перетворивши їх на завдання розробки, включені до майбутнього беклогу спринту [36]. У діяльності *«Інтеграція та валідація»* інструмент Q-Rapids також використовується для перевірки відповідності реліз-кандидата обмеженням якості, визначеним Softeam для кожного поточного продукту, що надається своїм клієнтам. Коли цей аналіз виявляє деякі недоліки якості, керівник проекту може вирішити не включати деякі функції або відкласти реліз.

Під час оцінювання спринту, якщо показники якості та стратегічні індикатори, пов'язані з процесом розробки, виявляють основні проблеми, процес може бути адаптований для наступного спринту (*діяльність з адаптації процесу*). Ці адаптації можуть приймати різні форми, такі як перегляд критеріїв якості, що застосовуються до всіх продуктів (коригування моделі якості), коригування порогових значень, починаючи з яких спрацьовує сповіщення, або оновлення QR-каталогу шляхом введення нових шаблонів або зміни критеріїв застосування існуючих.

В. QFL для виявлення QR та планування QR

Інструмент Q-Rapids надає новий механізм, який дозволяє Softeam контролювати, чи відповідає продукт їхнім критеріям якості; ця функція полягає в напіваавтоматичній генерації QR-кодів. Зображено сумісність інструментів.

Налаштування системи генерації QR-кодів розпочалося з налаштування порогових значень сповіщень (компонент qr-alert; крок 1 на рис. 2.3). Інструмент Q-Rapids періодично перевіряє показники якості, обчислені інструментом, відповідно до налаштовуваних порогових значень, визначених як цілі якості інженерами з якості Softeam. Якщо контрольований показник якості перевищує

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

визначені порогові значення, команда з якості Softeam отримує сповіщення через панель інструментів Q-Rapids (qr-dashboard; крок 2). Інженерам з якості пропонується адаптована відповідь у вигляді параметризованого QR-коду (з каталогу шаблонів QR-кодів), які можуть включити її до беклогу [37]. У цьому випадку в інструменті управління проектами Softeam «Відкрити проект» автоматично буде згенеровано кандидата QR-коду, який пропонує рішення для вирішення проблем з якістю (qr-backlog; крок 3) .

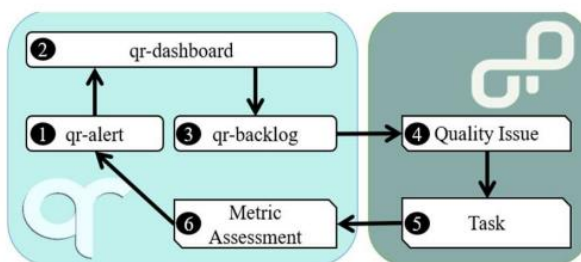


Рисунок 2.3 - Ланцюжок інструментів QFL у Softeam

На стороні OpenProject згенерований QR-кандидат інтегрується в дорожню карту проекту як питання якості. Питання якості реалізуються як новий тип абстрактного завдання Робочого пакету під назвою « *QualityRequirement* ». Softeam вирішив не генерувати конкретні завдання безпосередньо в дорожній карті проекту, що дозволяє керівнику проекту вибрати найкращий підхід до вирішення виявленого питання якості. Керівник проекту вирішує зберегти або відхилити питання якості. Для прийнятих питань якості керівник проекту визначатиме завдання розробки, коли вирішить перенести їх на фазу розробки.

На рис. 8 наведено приклади QR-запитів, згенерованих за допомогою інструменту OpenProject від Softeam. Деякі з них були прийняті та виводять одне або декілька завдань розробки, а деякі – відхилені.

Щоб оцінити зручність використання напівавтоматичної генерації QR-кодів, ми запросили у інженерів з якості та керівників проектів щомісячні звіти з визначенням історій використання, тобто визначенням реальних сценаріїв, де вони використовують цю функцію, та метою використання, згідно з наступним

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

шаблоном:

As a <role>, I have used <tool/tool feature>, to
<purpose>, during <activity>.

OpenProject отримуватиме сповіщення, пов'язані з потенційною проблемою якості в Modelio NG, через нові вимоги до якості, згенеровані в моєму Інструменті управління проектами (пов'язані зі співвідношенням відкритих проблем та співвідношенням належним чином прокоментованих файлів), під час зустрічі на дошці керування. Зустріч на дошці керування відповідає активності функцій плану (WBM).

Доповнюючи використання сценарію, користувач також міг повідомити про сильні сторони та загальну задоволеність. Той самий користувач повідомив про рівень задоволеності як «Задоволений» та наступну сильну сторону: «Я зміг вирішити включити нове завдання на платформі розробки проектів, щоб вирішити проблему «Співвідношення відкритих питань» QR та відхилити «співвідношення належним чином прокоментованих» (ця проблема буде вирішена шляхом нагадування розробнику про правила кодування)».

Розроблений Softeam інструмент Open Project Connector для генератора QR-кодів є загальнодоступним на GitHub. Ми розробили спеціальний розширюваний REST API на основі Spring Boot для перетворення QR-кодів, згенерованих інструментом Q-Rapids, у робочі елементи OpenProject. Інформація про специфікацію служб беклогу доступна на GitHub.

А. QFL для моніторингу зворотного зв'язку QR

Інструмент Q-Rapids забезпечує візуалізацію оцінки якості, яку можна використовувати для моніторингу та перевірки QR-кодів, а також для оцінки процесу розробки QR-кодів.

Щодо моніторингу та перевірки QR, інструмент містить історичний вигляд з інформацією про рішення щодо додавання QR. На рис. 2.4 (ліворуч) рішення щодо додавання QR позначено зеленими хрестиками. Ця візуалізація

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє нам проаналізувати, чи відповідає розробка QR очікуванням на момент отримання QR, тобто чи зростає оцінка відповідного елемента моделі якості після прийняття рішення. На діаграмі показано, як оцінка метрики «Відсоток пройдених тестів» покращується на 20% після додавання QR. «Відсоток пройдених автоматичних тестів має бути щонайменше 0,95» станом на 26 червня.

Щодо процесу розробки QR, використовуючи стратегічний показник Цикл зворотного зв'язку щодо якості (таблиця II.), керівник проєкту може контролювати розвиток релевантності отриманого QR (рис. 2.4, праворуч, помаранчева серія) та, для QR, які він чи вона прийняли, процес закриття завдань розробки, пов'язаних з ними (рис. 2.4, праворуч, синя серія).



Рисунок 2.4 - Стан тестування (ліворуч) та цикл зворотного зв'язку щодо якості (праворуч) в історичних переглядах

Релевантності вимог до якості оцінює якість кандидатів QR-запитів, згенерованих інструментом. Низькі значення цього показника свідчать про погану конфігурацію компонента сповіщення (тобто невідповідні порогові значення). Коефіцієнт *завершення вимог до якості* оцінює прогрес розробки згенерованих QR-запитів, які вже перебувають на стадії розробки (тобто мають похідні завдання розробки в журналі).

На рис. 2.4 (праворуч) показано, як коефіцієнт *релевантності вимог до якості* має деякі коливання на початку періоду, досягаючи стабільної оцінки

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

після налаштування порогових значень, що запускають сповіщення. Коефіцієнт виконання вимог до якості показує, як розробка була завершена протягом періоду, відповідно до очікуваної поведінки, будучи низьким на початку та високим наприкінці.

Візуалізація оцінки якості, доповнена деякими іншими функціями, що надаються інструментом (прогнозування та аналіз «що, якщо»), також допомагає інженеру з якості та керівнику проекту в завданні ручної ідентифікації нових QR-кодів, замикаючи цикл якості зворотного зв'язку.

Протягом періоду з травня по грудень 2019 року інструмент Q-Rapids запропонував сім QR-запитів на основі сповіщень про якість. Шість з них були прийняті інженерами з якості. З цих шести чотири були підтверджені керівниками проектів та перетворені на завдання розробки, а два були відхилені. В результаті 85% QR-запитів, запропонованих Q-Rapids, були прийняті інженерами з якості, з них 66% були прийняті керівниками проектів. Іншими словами, 57% QR-запитів, запропонованих інструментом Q-Rapids, опинилися в беклозі продукту, включеному до дорожньої карти продукту, з планами розробки командою розробників.

2.4 Висновок по розділу

У другому розділі було досліджено методи та інструменти забезпечення швидкого зворотного зв'язку в процесі розробки програмного забезпечення. Розглянуто архітектурні та організаційні підходи до реалізації циклів зворотного зв'язку, що дозволяють підвищити адаптивність програмних систем. Проаналізовано методи збору та обробки даних зворотного зв'язку, які забезпечують отримання актуальної інформації про роботу програмного продукту та поведінку користувачів. Також досліджено сучасні інструменти аналітики та моніторингу програмного забезпечення, що підтримують процес прийняття рішень у реальному часі. У результаті визначено ефективні підходи до організації швидких циклів зворотного зв'язку.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЦИКЛІВ ЗВОРОТНОГО ЗВ'ЯЗКУ

3.1 Реалізація механізмів швидкого зворотного зв'язку у програмному продукті

У цьому підрозділі представлено результати тематичного аналізу. Він - починається з короткого огляду учасників інтерв'ю та пояснюється, як коди були перетворені на теми.

Таблиця 3.1

Огляд учасників інтерв'ю

№	Країна (Country)	Поточна роль (Current Role)	Досвід в Agile (Agile Experience)
1	Швеція	Старший розробник / Провідний інженер	10+ років
2	Швеція	Менеджер проєкту	5+ років
3	Швеція	Agile-коуч	20+ років
4	Індія	Засновник, екс-віце-президент із розробки продукту	15+ років
5	Німеччина	Agile-коуч / Scrum-майстер	10+ років
6	Нова Зеландія	Scrum-майстер	15+ років
7	Швеція	Власник продукту (Product Owner)	10+ років
8	Швейцарія	Власник продукту (Product Owner)	5+ років
9	Німеччина	Agile-коуч / Scrum-майстер	5+ років

Учасники інтерв'ю

Було проведено дев'ять інтерв'ю з фахівцями з Agile з п'яти країн, більшість з яких базуються у Швеції. У таблиці Знаведено огляд учасників, включаючи їхню країну проживання, поточну посаду та роки досвіду роботи з Agile. Усі респонденти мають щонайменше п'ять років досвіду роботи в Agile-середовищі. Їхній рівень досвіду гарантує, що їхні відповіді ґрунтуються на реальному досвіді та хорошому розумінні Agile-роботи.

Інтерв'ю включали питання про їхню поточну роль та компанію, а також про їхні попередні ролі та компанії [38]. Ці деталі допомагають отримати ширшу картину та дозволяють порівнювати різні організації. Оскільки респонденти виконували різні ролі, такі як власник продукту або Agile-коуч, вони також могли поділитися різноманітними точками зору та досвідом.

Коди до тем

Після вилучення кодів, їх фільтрували та сортували в таблицю, де створювали різні категорії. Одним із прикладів того, як коди були розроблені за темами, є коди «важливий швидкий зворотний зв'язок», «надання пріоритету змістовному зворотному зв'язку» та «одержимість клієнтом». Вони були згруповані з іншими кодами, які утворили початкову категорію «Стратегії для ефективнішого зворотного зв'язку». Аналогічно, коди «спостереження за користувачами», «запрошення реальних користувачів до тестування та огляду продукту» та «поставити розробників на місце кінцевого користувача» були згруповані та названі «Залучення користувачів та зворотний зв'язок». Після категоризації кодів ми мали загалом тринадцять категорій у таблиці. Наступним кроком у процесі було групування пов'язаних категорій та формування тем і підтем. Дві категорії, «Стратегії для ефективнішого зворотного зв'язку» та «Залучення користувачів та зворотний зв'язок», були згруповані та утворили підтеми в рамках ширшої теми «Стратегії та впровадження».

Визначені теми

У дослідженні було визначено три основні теми, які були ретельно досліджені та будуть детально обговорені в наступних розділах. Ці теми є «Стратегії та впровадження» охоплюють стратегії зворотного зв'язку, типи циклів зворотного зв'язку, їх час виконання, а також стратегії залучення користувачів, збору зворотного зв'язку та оцінювання. Розуміння цих стратегій допомагає відповісти на дослідницьке питання «Як рішення для покращення циклів зворотного зв'язку з користувачами в гнучкій розробці програмного забезпечення впроваджуються на практиці та як пом'якшуються проблеми впровадження?», розкриваючи практичні підходи до впровадження циклів

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

зворотного зв'язку з користувачами в гнучкій розробці.

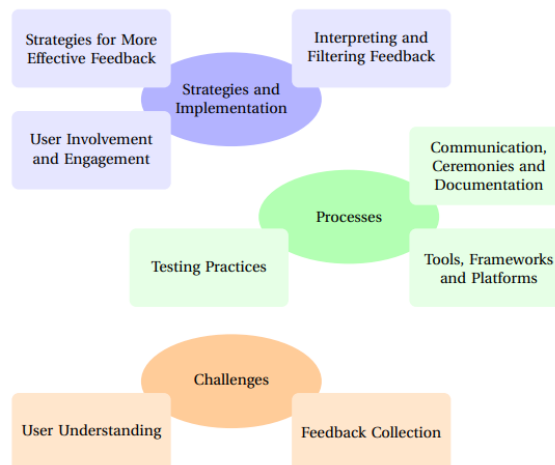


Рисунок 3.1 - Тематична карта

Загальні стратегії для ефективніших циклів зворотного зв'язку

У більшості інтерв'ю поширеною темою було те, що зворотний зв'язок від користувачів має бути швидким, простим і здійснюватися на ранніх етапах процесу розробки. Це важливий фактор для максимізації створення цінності та підтримки гнучкості. Ідея швидкого зворотного зв'язку на ранніх етапах процесу полягає в тому, що чим швидше ви зможете зреагувати та змінити курс, тим дешевшими будуть зміни. Водночас багато хто наголошував на важливості планування процесу зворотного зв'язку та витрачання часу на прототипи, щоб уникнути високих витрат на подальших етапах циклу розробки.

Поширеною стратегією швидкого зворотного зв'язку є збір інформації на етапі вимог, до початку розробки, а не створення функції чи продукту до того, як буде забезпечена його цінність. Респонденти описали такі дії, як схвалення клієнтами окремих функцій та перевірка ідей щодо функцій з користувачами. Використовуючи цей підхід, команди можуть «швидко зазнавати невдач» та виявляти проблеми на ранній стадії, зменшуючи втрачений час розробки.

Як спосіб ранньої перевірки ідей функцій або продуктів, кілька учасників повідомили про використання прототипів різних видів. Одним із прикладів, згаданих респондентом, був ранній прототип системи видачі посилань,

розгорнутої в магазинах для збору відгуків користувачів. Система була погано сприйнята користувачами, тому ця вправа допомогла їм вирішити не повноцінно впроваджувати її, що зрештою заощадило їм витрати на повну розробку системи. Інші учасники повідомили про використання більш легких прототипів, таких як каркаси або ескізи на дошці чи аркуші паперу, як дуже ранніх прототипів для візуалізації своїх ідей.

Зібрані відгуки користувачів зазвичай додавалися до журналу обробки запитів або використовувалися для визначення пріоритетності існуючих елементів. Учасники описували використання відгуків для відкриття квитків на помилки, створення нових елементів журналу обробки запитів та коригування пріоритетів залежно від частоти виникнення або критичності вхідних даних. Більш термінові проблеми могли призвести до виправлення, тоді як незначні пропозиції, пов'язані з UX, накопичувалися та враховувалися в подальшому. Було наголошено на важливості розуміння не лише «що», але й «чому» стоїть за потребою користувача. Такий підхід допоміг забезпечити вирішення правильних основних проблем.

Ще однією поширеною темою як загальної стратегії була підтримка орієнтації на користувача. Респондент 3 висловив це так: «Ми одержимі клієнтом. Це означає, що чого б вони не хотіли, саме цього ми хочемо досягти, щоб зробити їх щасливими». Як необхідність згадувалося наймання правильних людей для просування цієї культури в компанії [39]. Наприклад, наймання розробників, які більше зосереджені на користувачеві, а не лише на технічних аспектах рішення. Ще однією згаданою стратегією було направлення нових співробітників для ознайомлення з продуктом на місці або для роботи в службі підтримки клієнтів протягом певного періоду. Такі практики допомагають подолати розрив між технічними командами та кінцевими користувачами, а також забезпечують узгодженість потреб розробки та реального світу.

Ще однією стратегією підвищення ефективності сесій зворотного зв'язку було завчасне складання порядку денного для демонстраційних сесій. Заздалегідь обговорювані вимоги часто допомагають уникнути невідповідностей

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

та зниження ефективності сесій зворотного зв'язку . Доведено, що краща підготовка значно покращує якість та актуальність обговорень.

Респонденти поділилися різними способами збору відгуків користувачів, включаючи формальні підходи, такі як заявки на підтримку, демонстраційні зустрічі та розсилка повідомлень, а також дуже неформальні підходи, такі як «тестування в коридорі». Цю практику було описано так в інтерв'ю 4: «Те, що я називаю тестуванням у коридорі, означає, що під час створення чогось, будь-хто проходить повз коридором, візьміть цю людину, посадіть її перед вашим монітором і скажіть: «Ось як ми працюємо або ось що ми робили досі»». Різні типи збору відгуків можна використовувати на різних етапах циклу розробки та допомагати збирати більш узагальнені відгуки з різних джерел.

Зрештою, було описано, що метою покращення якості зворотного зв'язку є орієнтація на конкретних партнерів для отримання зворотного зв'язку. Наприклад, шукаючи технічно інноваційні компанії-партнери, перші користувачі могли б надавати якісний та практичний зворотний зв'язок на ранніх етапах процесу розробки.

Залученість та взаємодія користувачів

Для створення ефективного циклу зворотного зв'язку в організації, тобто повного залучення користувачів до процесу розробки, вирішальне значення має. Інтерв'ю виявили кілька стратегій для досягнення цієї мети. Часто згадуваним моментом була важливість тісної інтеграції користувачів у робочий процес, зосереджуючись на залученні реальних кінцевих користувачів. Як зазначив респондент 5: «запрошуйте реальних користувачів, а не лише клієнтів». Було виділено кілька стратегій для цього, таких як залучення користувачів до двотижневих оглядів або демонстрацій для швидкого збору відгуків [40]. Такий підхід може допомогти оцінити, чи рухається команда в правильному напрямку, та забезпечити узгодженість дій між усіма зацікавленими сторонами, включаючи команду розробників та операторів.

Інший підхід - це «shadowing» користувачів шляхом спостереження за їхньою взаємодією з продуктом, щоб виявити будь-які проблеми, які у них

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

виникають. Цей метод дозволяє командам визначати часто використовувані функції та проблемні області, не вимагаючи від користувачів безпосереднього зворотного зв'язку. Крім того, вони можуть спостерігати, як користувачі взаємодіють з продуктом, без попереднього знайомства, що забезпечує більш достовірне уявлення про їхнє фактичне використання.

Третя згадана стратегія полягає в отриманні думок безпосередньо від користувачів, наприклад, шляхом поїздки для особистих зустрічей з користувачами для отримання відгуків або проведення інтерв'ю. Одним із прикладів, наведених в інтерв'ю, було те, що під час обговорення нової функції команда вирішила відвідати користувачів безпосередньо в магазині, щоб негайно зібрати відгуки. Цей метод заощадив час на тривалі обговорення та дозволив їм зібрати прямі відгуки, які були враховані у фінальній версії.

Інтерв'ю також показали, що пропонування певних стимулів для заохочення користувачів до надання відгуків є важливим. Участь користувачів можна заохочувати за допомогою невеликих винагород, таких як непрямі платежі, пропонування безкоштовного місяця їхнього продукту або надання їжі та напоїв, коли користувачів особисто запрошують. Респондент в інтерв'ю 7 також зазначив, що користувачі більш мотивовані, коли відчують себе «залученими, і їм, як правило, більше подобається система».

У деяких інтерв'ю також зазначалося, що зворотний зв'язок слід збирати всередині компанії, дозволяючи людям бачити продукт з точки зору користувачів. Однією зі стратегій, згаданих в інтерв'ю №1, було «договірне тестування», яке виявилось корисним для отримання першого зворотного зв'язку всередині компанії. Це допомагає переконатися, що людям всередині компанії дійсно подобається продукт, як зазначив респондент у тому ж інтерв'ю: «Якщо ви не змогли змусити своїх співробітників сподобатися продукту, який ви створюєте, то вам важко очікувати, що іншим сподобається ця функція». Якщо внутрішні користувачі вже мають проблеми, у клієнтів, ймовірно, їх буде ще більше. Це також дозволяє командам виправляти ранні перешкоди та погані функції, перш ніж показувати продукт клієнтам. Іншим зазначеним методом

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

було перетворення розробників на користувачів. Розробники повинні почати думати з точки зору користувача. Один з опитаних пояснив, що це можна зробити, дозволивши їм протягом одного-двох тижнів випробувати щоденну роботу користувача, щоб вони могли краще зрозуміти, яке життя користувача.

Створення міцного та стабільного зв'язку між двома сторонами є важливим для побудови успішного циклу зворотного зв'язку між розробниками та користувачами. Однак, припустимо, що самі розробники виступають прямим контактом для користувачів. У такому випадку це може призвести до проблем, оскільки вони втрачають дорогоцінний час і не можуть повністю зосередитися на своїх основних завданнях: «якщо користувач знає, що це розробник, то він зв'яжеться з ним безпосередньо, і тоді розробники ніколи не матимуть часу, щоб реально працювати та розробляти речі» (респондент 9). Деякі компанії з інтерв'ю вирішили цю проблему, запровадивши спеціальні ролі, які діють як комунікаційні мости між командами розробників та користувачами. Однією з цих ролей є бізнес-аналітик (БА). Респондент 2 вказав на важливість цієї ролі, сказавши: «А також з точки зору зворотного зв'язку, БА дуже важливі для нашого бізнесу, тому нам потрібно мати їх більше».

Бізнес-аналітики відповідають за збір, аналіз та чітке донесення вимог між бізнес-сторonoю та командами розробників. Така схема гарантує, що клієнти завжди мають надійну контактну особу, а розробники можуть зосередитися на своїй роботі. Також згадувалися експерти з предметної області (SME). У деяких випадках одна людина виконує обидві ролі, поєднуючи глибокі знання продукту з сильними навичками розуміння та донесення вимог. В іншій компанії власник продукту в основному займається спілкуванням з користувачами. Один з опитаних сказав, що ідеальна схема полягає в тому, щоб одна або дві людини постійно контактували з користувачами.

Важливість гарної комунікації з клієнтами згадувалася кілька разів. Один з опитаних наголосив, що внутрішні команди слід навчити краще комунікувати, ставити правильні запитання, збирати чіткі відгуки та ввічливо казати «ні», коли запит клієнта неможливий або виходить за межі доступного бюджету. Інший

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

респондент наголосив, що важливо залишатися приземленим, залишатися відкритим до зворотного зв'язку, приймати конструктивну критику та підтримувати професійну дистанцію від продукту.

Інтерпретація та фільтрація зворотного зв'язку

Після збору відгуків користувачів так само важливо їх правильно інтерпретувати та фільтрувати. Навіть попри те, що від клієнтів очікується чесний, відкритий та чіткий відгук, користувачі часто мають інші пріоритети та обмежений час для надання обдуманих відповідей. Як зазначив респондент 6: «Хоча відгук важливий, він не завжди відповідає дійсності». Також кілька разів зазначалося, що користувачі іноді висувають нереалістичні запити. Респонденти 6 та 8 посилалися на відому цитату, яку часто приписують Генрі Форду, засновнику Ford Motor Company: «Якби я запитав людей, чого вони хочуть, вони б сказали швидших коней».

У коментарях підкреслюється важливість ретельного аналізу та оцінки - відгуків. Компанії опитаних учасників застосовують різні стратегії для кращого розуміння та інтерпретації відгуків користувачів. Один із підходів полягає в оцінці того, чи стосується відгук ширшої групи користувачів, чи він відображає лише індивідуальну думку. Один з опитаних пояснив, що вони відтворюють сценарії користувачів, щоб краще визначити, чи є відгук помилкою, запитом на функцію чи іншим типом покращення, і чи його впровадження створить цінність для більшості користувачів. Інша стратегія полягає в тому, щоб спочатку перевірити, чи запитувана функція вже існує, але, можливо, була пропущена користувачем. Пропозиція додається до списку невиконаних завдань, якщо відгук має реальну цінність.

Зрештою, не всі відгуки будуть впроваджені. Однак важливо ретельно переглянути та зрозуміти отримані відгуки й приймати обґрунтовані рішення на їх основі.

Процеси

Ще однією визначеною темою є «Процеси», яка зосереджена на різних практиках, інструментах та структурах, що використовуються для впровадження

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

циклів зворотного зв'язку. Розуміння цих процесів допомагає відповісти на дослідницьке питання, надаючи уявлення про те, як різні організаційні практики та інструменти сприяють успішному впровадженню циклів зворотного зв'язку з користувачами.

Комунікація, церемонії та документація

Ефективні практики комунікації, чітка документація та чітко визначені вимоги стали вирішальними механізмами для забезпечення роботи циклів зворотного зв'язку. Різні гнучкі фреймворки та їхні вбудовані або запропоновані церемонії створені для підтримки цих процесів, але вони можуть сильно відрізнятися між компаніями та окремими командами. Практики, описані учасниками, є поєднанням формалізованих стратегій та більш адаптивних, ситуативно-специфічних практик, і часто очевидно, що для успішного збору та впровадження зворотного зв'язку потрібно більше, ніж просто церемонії. Учасники інтерв'ю 3 та 6 зазначили, що державні установи та компанії, що спеціалізуються на апаратному забезпеченні, можуть мати труднощі з повним переходом на гнучкі методи. Одна з них сказала наступне про свій минулий досвід: «Я думаю, що це частина апаратного забезпечення, але я думаю, що вони роблять те саме в органах влади та інших компаніях, що вони завжди перебувають у поточному проекті. Тож це не просто гнучке середовище. І це дещо ускладнює ситуацію» (інтерв'юований 3).

Як згадувалося, церемонії створені для підтримки інтеграції зворотного зв'язку, а їхня ітеративна природа є однією з речей, що робить їх гнучкими. Церемонії можуть відрізнятися між командами залежно від таких факторів, як культура, розмір команди та розмір проекту. Це залежить від потреби, що проілюстрував учасник інтерв'ю 7, який описав конкретний тип церемонії, яку вони використовували: «У великих проектах ми можемо проводити інші види сесій та запрошувати людей на дизайн-спринт чи щось подібне. У нас є приблизно день, коли ми збираємо всю команду та деяких користувачів і намагаємося отримати відгуки про те, що ми робимо». Ця практика адаптована до їхнього поточного проекту та дозволяє їм швидко перебирати ідеї та відгуки,

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

іноді навіть розробляти нові ідеї на місці. Один з опитаних також згадав про наявність внутрішніх сесій, де відгуки та завдання були закріплені за командою: «Потім у нас є остаточні огляди та зустріч, де ми фактично переконуємося, що кожен знає, що він має робити» (інтерв'юований 2).

Комунікація є вирішальною частиною збору та змістовного реагування на зворотний зв'язок. Кілька учасників обговорили проактивні стратегії для сприяння відкритому та ефективному спілкуванню. Респондент 5 розповів, як він діє, коли члени команди вагаються ставити запитання: «Якщо я бачу, що на їхніх обличчях є якісь запитання, але вони не достатньо сміливі, щоб дати свій відгук або поставити запитання, які в них є наразі, я пишу безпосередньо в чаті: «Не соромтеся ставити запитання, саме для цього і присвячена ця зустріч»». Такий підхід допомагає зменшити ризик того, що невисловлені занепокоєння підірвуть узгодженість проекту з потребами, які він має задовольнити. Він також сприяє культурі відкритих, інтерактивних зустрічей.

Інша, більш формалізована комунікаційна стратегія – це проведення зовнішніх конференцій, де команда збирається разом із клієнтом, щоб досягти згоди щодо проекту. Учасник інтерв'ю 1 описав конференцію так: «А потім у нас є ці зовнішні конференції, де ми досягаємо згоди із замовником. Це ті документи Word з інструменту вимог, про які замовники домовляються. Відбувається пінг-понг, і тільки після того, як вони домовляються про це на зовнішній конференції, тільки тоді справа переходить на більш детальний рівень». Комунікаційні стратегії варіюються від формальних до формальних, щоб допомогти подолати розрив між користувачами, зацікавленими сторонами та розробниками на різних рівнях.

Документація також відіграє вирішальну роль у забезпеченні узгодженості очікувань та мінімізації непорозумінь. Кілька учасників наголосили на важливості спільного, стандартизованого способу документування вимог користувачів: «У нас є спільний спосіб документування наших вимог, який вирішує багато проблем. Для цього визначені правила, і ці правила розуміють розробники, тестувальники, всі, хто бере участь» (у -

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

респондента 2). Чіткі правила щодо документування допомагають уникнути суперечливих інтерпретацій, особливо у великих організаціях.

Практика тестування

В інтерв'ю згадувалися різні типи тестування користувачів, кожне з яких стосується різних етапів розробки продукту. На етапі розробки одним із обговорюваних методів було тестування з реальними клієнтами. Під час цих сесій користувачам надається продукт і їх просять виконати певні дії, що дозволяє команді побачити, як користувачі природно взаємодіють з продуктом. Цей підхід добре працює для простіших продуктів, оскільки користувачі можуть їх протестувати.

без вступу. Однак для складніших продуктів рекомендується спочатку надати вступ, оскільки інакше «люди розчаруються ще до того, як спробують» (респондент 9).

Ще одним методом, який з'явився на етапі розробки, було А/В - тестування. Цей підхід передбачає створення двох версій продукту для двох різних груп та порівняння їхньої ефективності, щоб побачити, яка з них працює краще.

Це особливо корисно для оцінки змін у користувацькому досвіді, оскільки потребує менше ресурсів, ніж інші методи тестування.

Тестування прийняття користувачами часто використовується перед випуском продукту, що допомагає переконатися, що програма працює належним чином та відповідає необхідним стандартам. Для складніших продуктів однією зі згаданих стратегій було надання користувачам демонстрації або проведення сесії запитань і відповідей, перш ніж їм дозволять тестувати продукт протягом двох тижнів. Це допомагає користувачеві зрозуміти, як орієнтуватися в продукті та чого очікувати.

Бета-тестування також є стандартним перед релізом, особливо для великих запусків. Замість того, щоб робити так званий «великий вибуховий реліз» для всіх користувачів, що може призвести до «хаосу», компанії часто спочатку випускають продукт для обраної групи. Щоб керувати цим,

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

використовується позначення функцій, щоб розгорнути певні функції для меншої групи. У деяких випадках користувачі, які добровільно беруть участь у бета-програмах, отримують ранній доступ до нових функцій. Ці користувачі надають відгуки, і після врахування їхніх пропозицій повний реліз стає доступним для всіх. Цікавим моментом, про який йшлося в інтерв'ю, було те, що одна компанія використовує людей, які не платять за продукт, для тестування нових функцій. Ці користувачі, як правило, більш відкриті до випробування нових функцій і, як правило, більш толерантні до помилок порівняно з платними клієнтами.

Після релізу автоматизовані системи зворотного зв'язку були згадані як цінний інструмент для продуктів, які вже доступні в Інтернеті. Наявність системи, яка дозволяє користувачам швидко та легко залишати відгуки, є важливою для активних додатків та веб-сайтів. Цей безперервний цикл зворотного зв'язку допомагає компаніям з часом покращувати свої продукти та послуги, вирішуючи проблеми та враховуючи пропозиції в міру їх виникнення.

Інструменти, фреймворки та платформи

Окрім методів тестування, в інтерв'ю було визначено різні інструменти, фреймворки та платформи для збору та структурування відгуків користувачів. Їх можна розділити на канали комунікації, інструменти UX/користувацького інтерфейсу (UI) та автоматизовані методи.

Канали комунікації включають такі інструменти, як Zoom, які дозволяють компаніям підтримувати зв'язок з клієнтами або користувачами навіть на відстані.

Доступні прямі канали комунікації, індивідуальні або групові. Один учасник зазначив, що його компанія має доступ до облікового запису Teams клієнта, що дозволяє пряму взаємодію. Ці прямі зв'язки пропонують кілька переваг, таких як пошук користувачів для тестування продуктів. Крім того, компанії постійно контактують з клієнтами та користувачами, а проблеми або відгуки можна швидко вирішити. Системи відстеження проблем, такі як Jira, також згадувалися як ефективні канали комунікації.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Інструменти UX/UI корисні на етапі дослідження. Кілька респондентів описали візуалізацію ідей як дуже важливу. Як зазначив респондент 7: «Ви повинні візуалізувати це, тому що неможливо пояснити комусь, як все працюватиме, бо на вас впливає власний досвід, ви упереджені». Для підтримки візуалізації компанії часто використовують прототипи, створені на основі вайрфреймів. Ці прототипи мають перевагу в тому, що вимагають мінімальних інвестицій, водночас забезпечуючи ранній зворотний зв'язок. Цей процес також створює цикл зворотного зв'язку, оскільки вайрфрейми переглядаються та коригуються на основі відгуків користувачів. Для цього використовуються такі інструменти, як Figma. Інший підхід – це карта шляху, яка допомагає компаніям візуалізувати користувацький досвід, визначати «больові точки» та досліджувати шляхи покращення загального процесу. Іншим згаданим візуальним інструментом є дошка, де люди можуть ділитися відгуками та ідеями, за які потім можна голосувати.

Деякі компанії використовують автоматизовані інструменти для збору відгуків. Одна компанія згадала про сервіс, який генерує хмару слів з коментарів у магазині додатків, які можна фільтрувати за країною та версією. Цей інструмент допомагає їм збирати та візуалізувати відгуки з різних регіонів. Інший інструмент аналізує стабільність продукту та відстежує такі показники, як час завантаження та показники відмов. Ці автоматизовані методи надають цінну інформацію, не вимагаючи прямого зворотного зв'язку від користувачів, формуючи невербальний цикл зворотного зв'язку, який доповнює внесок користувача.

Один учасник згадав, що вони використовують інструмент управління вимогами. Інструмент доступний для всіх в організації, що дозволяє їм відстежувати деталі продукту та вимоги. Він включає механізм зворотного зв'язку, щоб команда могла надавати інформацію перед початком розробки. Крім того, компанія почала інтегрувати штучний інтелект для оптимізації репозиторію. Співробітники можуть ставити запитання штучному інтелекту, на які він автоматично відповідає. Використання штучного інтелекту для пошуку у

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

вимогах є корисним для великих проектів, оскільки штучний інтелект може виявляти закономірності, які можуть бути не одразу очевидними для окремих осіб, заощаджуючи час і покращуючи загальне розуміння проекту.

Виклики

Третя визначена тема – «Виклики». Ця тема описує поширені перешкоди, з якими стикаються учасники. Розуміння цих викликів допомагає відповісти на дослідницьке питання, визначаючи перешкоди для впровадження ефективних циклів зворотного зв'язку з користувачами. Крім того, вона надає інформацію про потенційні рішення для подолання цих викликів, тим самим покращуючи інтеграцію зворотного зв'язку в процеси гнучкої розробки.

Проблеми розуміння користувачами

В інтерв'ю було виявлено кілька проблем, пов'язаних з розумінням користувачів та інтерпретацією їхніх відгуків. Значною проблемою є те, що користувачі можуть не розуміти нових функцій або не використовувати їх, оскільки вони ставлять на перше місце інші потреби. Навіть добре розроблені функції можуть зазнати невдачі, якщо користувачі їх не приймають: «у вас може бути чудова функція, але якщо користувачі нею не користуються, немає сенсу витрачати на неї багато часу» (інтерв'юований 8).

Одна з проблем збору відгуків користувачів полягає в тому, що вони часто ґрунтуються на особистому досвіді, який може не відображати ширшу базу користувачів. Один з опитаних зазначив, що відгуки іноді можуть бути вузькими або упередженими, зосереджуючись на індивідуальних поглядах, а не на колективному розумінні того, що працює найкраще. Важливо враховувати, хто надає відгуки. Новатори, або ті, хто першими впроваджує продукт, як правило, більш поблажливі до помилок і прагнуть допомогти покращити його. Навпаки, ті, хто пізно впроваджує продукт, менш терплячі та частіше скаржаться, якщо щось не працює одразу. Тому вкрай важливо пріоритезувати відгуки від потрібних користувачів та оцінити, наскільки їхній внесок репрезентативний для всієї групи користувачів. Не всі відгуки можна впровадити, оскільки це було б дорого та неефективно. Фільтрація найбільш релевантних та практичних

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

висновків є важливою, щоб уникнути марнування ресурсів.

Практичні приклади з інтерв'ю показують, наскільки важко правильно визначити потреби користувачів. В одному випадку, незважаючи на численні демонстрації та спринт-огляди, кінцевий продукт не виправдав очікувань користувачів. Основна проблема полягала в тому, що зворотний зв'язок не був повністю зрозумілий, і бракувало розуміння того, як користувачі насправді взаємодіятимуть із функціональністю. Як зазначив учасник інтерв'ю №3: «Якщо ви насправді не ставите правильних запитань, а лише слухаєте, ви можете багато чого втратити».

Інший приклад стосувався пілотного проєкту, де тестувальники отримали забагато підтримки під час фази тестування. Ця додаткова допомога зрештою приховала проблеми, які в іншому випадку були б виявлені.

Більше того, було зазначено, що навіть висококваліфікованим розробникам може бути важко повністю зрозуміти потреби користувачів. Технічна експертиза не перекладається в інтуїтивне розуміння поведінки користувачів, що підкреслює важливість тісної співпраці між командами розробників та експертами з дослідження користувачів.

Переешкоди для збору відгуків

Існує кілька труднощів, коли йдеться про збір змістовних відгуків. Одним із питань, що було підкреслено в інтерв'ю, є різниця між платними та неплатними користувачами. Як зазначив респондент 6, платні користувачі, як правило, більш чесні у своїх відгуках: «Вони говорять правду лише тоді, коли вкладають свої гроші». Це показує важливість створення стимулів, які заохочують користувачів ділитися більш чесними та цінними думками.

Ще однією проблемою є те, що користувачі походять з різних країн. Через культурні відмінності функції чи програми можуть використовуватися по-різному, що може призвести до різних відгуків. Як наслідок, продукти, можливо, доведеться адаптувати для певних регіонів. Різниця в часових поясах також ускладнює збір відгуків, а тести іноді проводяться лише в одній країні, що ускладнює узагальнення результатів. Мовні бар'єри ще більше ускладнюють

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

ситуацію, деякі компанії навіть кажуть: «О, тоді ми просто не будемо тестувати це з жодною з них» (інтерв'юований 9). Таке ставлення може заважати компаніям по-справжньому зрозуміти потреби користувачів, покладаючись виключно на внутрішнє тестування.

Деякі розробники також надають перевагу тому, щоб зворотний зв'язок залишався всередині команди. Або тому, що вони стурбовані технічними аспектами, які можуть не до кінця розуміти, або тому, що їм незручно щодо того, як вони виглядатимуть на зустрічах. Респондент 9 також зазначив: «Їм було дуже важко застосувати ідею та менталітет Scrum». Ще однією виявленою проблемою є те, що більшість розробників більше зосереджуються на демонстрації виконаної роботи, ніж на отриманні зворотного зв'язку.

Пошук потрібних тестувальників також може бути проблемою. Як зазначив респондент 7: «Люди, які хочуть долучитися, зазвичай не є представниками звичайних користувачів». Ця проблема демонструє проблему надмірної залежності від знайомих контактів або членів внутрішньої команди. Хоча деякі клієнти відкриті до демонстрацій та змін, вони також можуть чинити опір наданню зворотного зв'язку. Таким чином, критично важливо переконатися, що клієнти готові витратити час, необхідний для надання змістовної інформації.

Постійною причиною ускладнення процесів зворотного зв'язку є обмеженість ресурсів. Кілька учасників зазначили, що тиск часу та зміна пріоритетів заважають їм витрачати достатньо часу на взаємодію з користувачами та підтримку циклів зворотного зв'язку. Респондент 8 пояснив, що їхнє робоче навантаження включає усунення відкладених запитів, підтримку розробників, допомогу користувачам там, де команди підтримки недостатньо, і, крім того, процеси зворотного зв'язку. Він пояснює, що так було не завжди: «У нас були юніти, які використовувалися в минулому, це були внутрішні менеджери з успіху клієнтів, які повинні були стимулювати впровадження та покращувати це. Але ці посади були скорочені під час реструктуризації минулого року, тому це також відображає роботу організації».

Серед кількох запропонованих причин невивільнення ресурсів для

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

взаємодії з користувачами є складність визначення цінності, яку надають посади, призначені для користувачів. Той самий учасник висловив це так: «Проблема, яку я бачу, полягає в тому, що важко кількісно оцінити та важко встановити ціну наданої цінності, що, ймовірно, є однією з причин, чому цих посад більше не існує». Учасник інтерв'ю № 6 стверджує, що компанії повинні припинити використовувати співробітників на межі своїх можливостей, щоб звільнити їх для важливих видів діяльності, таких як обмін інформацією: «Якщо у вас є хтось, хто доступний. І немає затримок, то це надзвичайно цінно, тому що ви дозволяєте цьому експерту оптимізувати ваші великі витрати, а саме вашу команду розробників». Занадто багато рівнів комунікації також можуть уповільнити процес зворотного зв'язку. Важливо забезпечити доступність і мати когось, хто відповідає за управління комунікацією без надмірного обміну даними, щоб забезпечити своєчасний збір та реагування на відгуки, що підвищить ефективність.

3.2 Аналіз результатів впровадження та вплив на якість програмного забезпечення

У цьому розділі обговорюються основні висновки дослідження стосовно дослідницького питання «Як рішення для покращення циклів зворотного зв'язку з користувачами в гнучкій розробці програмного забезпечення впроваджуються на практиці, і як пом'якшуються проблеми впровадження?». Обговорення структуровано навколо основних тем, визначених у тематичному аналізі, проведеному на основі інтерв'ю. Результати інтерв'ю також будуть пов'язані та порівняні з висновками з існуючої літератури.

Впровадження циклів зворотного зв'язку з користувачем

Різні практичні стратегії, які Agile-команди використовують для посилення своїх циклів зворотного зв'язку. Найголовнішими з них, що вплинули з інтерв'ю, були: необхідність швидкого та раннього зворотного зв'язку для мінімізації втрат та перевірки ідей, цінність глибокої участі користувачів та

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

культури компанії для її підтримки, а також важливість інтерпретації та фільтрації зворотного зв'язку. Практики, описані в цих темах, повинні бути збалансовані відповідно до потреб вашої команди та зацікавлених сторін під час інтеграції зворотного зв'язку з користувачами.

Швидкий та ранній зворотний зв'язок

Значний акцент було зроблено на зборі відгуків на ранніх етапах циклу розробки з використанням недорогих методів, таких як ескізи, вайрфрейми та інші прототипи. Ідеї слід перевіряти на ранніх етапах, щоб уникнути непотрібних зусиль з розробки та дозволити командам «швидко зазнавати невдач і мінімізувати втрати». Такий підхід добре узгоджується з ітеративною природою гнучких фреймворків та зосередження на постійному вдосконаленні. Натомість учасники також наголосили на необхідності планувати сесії зворотного зв'язку задовго до їх проведення, щоб забезпечити їхню цілеспрямованість та максимально ефективність. Хоча гнучкість сприяє швидкості, також необхідно планувати стратегічно, щоб зворотний зв'язок залишався актуальним і його можна було перетворити на щось практичне. Командам необхідно знайти баланс, щоб надавати високоякісний зворотний зв'язок, залишаючись при цьому - гнучкими.

Глибока залученість користувачів та культура

Ще однією важливою темою була важливість прямої участі користувачів. Більшість учасників доклали зусиль, щоб залучити кінцевих користувачів до своїх процесів зворотного зв'язку, а не зосереджуватися лише на бізнес-зацікавлених сторонах, де це можливо. Для подолання цього розриву використовувалися такі практики, як спостереження за користувачами та направлення розробників для роботи з продуктом на місці. Однак багато учасників також заявили, що вони хочуть більш прямої участі користувачів у своєму циклі розробки, щоб скоротити лінію комунікації та покращити розуміння потреб користувачів. Хоча деякі учасники хотіли прямого контакту між розробниками та користувачами, інші стверджували, що важливо не обтяжувати розробників без потреби та використовувати натомість

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

посередників, таких як бізнес-аналітики та власники продуктів. Це гарантувало б, що розробники не будуть перевантажені надмірними комунікаційними завданнями та зможуть вільно зосередитися на розробці. Багато хто вважав посередницькі ролі вирішальними для взаємодії з користувачами та отримання максимальної цінності з їхніх відгуків. Тим не менш, ресурси є постійною проблемою, пов'язаною з наявністю цих виділених ролей.

Інтерпретація та фільтрація зворотного зв'язку

Збір відгуків, звичайно, є критично важливою частиною циклу зворотного зв'язку, але інтерв'ю показали, наскільки важливо вміти правильно інтерпретувати та фільтрувати відгуки. Не всі відгуки слід враховувати, і може бути складно витягти з них практичні висновки - необхідно використовувати структуровану інтерпретацію та критерії фільтрації. Потрібно переконатися, що відгуки представляють групу користувачів, і виявити помилки, нові функції або непорозуміння, можливо, у формі недостатньо використаної існуючої функції. Цей крок процесу має вирішальне значення для правильного розподілу ресурсів розробки.

Проблеми впровадження

Під час інтерв'ю було виявлено кілька ширших проблем, серед яких були затримки та неефективність, спричинені організаційними обмеженнями, недостатня залученість користувачів через брак мотивації, високе робоче навантаження та проблеми з комунікацією, часто спричинені занадто великою кількістю рівнів між розробниками та користувачами.

Затримки та неефективність

Деякі учасники визначили довгі цикли зворотного зв'язку або їх відсутність як проблему, переважно в менш гнучких організаціях, з різних причин. Прикладами були урядові установи та організації, орієнтовані на апаратне забезпечення, яким важко керувати переходом від традиційної до гнучкої розробки. Один учасник поділився своєю думкою, що такі організації можуть мати труднощі з повним переходом на гнучку методологію, припустивши, що існує особливе середовище, окрім гнучкого, що може

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

ускладнити їх об'єднання. Проблеми з впровадженням гнучкої методології в цих середовищах підтверджуються дослідженнями щодо гнучкої трансформації в урядових організаціях та організаціях, орієнтованих на апаратне забезпечення, відповідно. Деякі методи пом'якшення цієї проблеми включають раннє залучення зворотного зв'язку та спроби зробити процес зворотного зв'язку максимально простим, щоб знизити бар'єр. Конкретні стратегії, такі як «тестування коридору», були згадані як легкі та швидкі варіанти зворотного зв'язку.

Відсутність залучення користувачів. Труднощі з безпосередньою взаємодією з кінцевими користувачами були постійною проблемою для багатьох учасників. Особливо складним було мотивування користувачів надавати цінніші відгуки. Легше залучити внутрішніх зацікавлених сторін або отримати відгуки від колег, але зазвичай це не те місце, де можна знайти найцінніший відгук. Методи пом'якшення включають орієнтацію на певні групи, які є ранніми користувачами та мають внутрішню мотивацію мати доступ до нових технологій та допомагати їх удосконалювати. Однак проблема, що виникає тут, полягає в тому, що цей тип користувачів може не бути репрезентативним для ширшої групи користувачів. Більш прості методи включають подарункові сертифікати або інші бонуси для тих, хто бере участь.

Робоче навантаження та використання інструментів. Високе робоче навантаження було поширеною проблемою, яку порушували багато учасників, головним чином через брак спеціалізованих ролей для інтеграції відгуків користувачів. Більшість учасників висловили задоволення своїми поточними інструментами або не мали конкретної думки. Найпоширенішими типами згаданих інструментів були інструменти для комунікації, UX/UI або автоматизації. Один учасник зазначив, що компанія намагається інтегрувати інструменти штучного інтелекту, які досі виявилися корисними. Вони також висловили побажання та сподівання на подальшу інтеграцію та майбутні технології штучного інтелекту, щоб допомогти знаходити закономірності та сортувати велику кількість відгуків та вимог.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

Комунікаційні бар'єри. Занадто багато рівнів комунікації та непорозуміння були названі поширеними перешкодами для ефективного спілкування з користувачами. Нездатність ставити правильні запитання була названа джерелом непорозуміння. Інтерпретація зворотного зв'язку може бути складною, але вкрай важливо зробити зворотний зв'язок дієвим. Використання спеціалізованих посередників, які можуть побудувати стосунки з користувачами та глибше розуміння, було названо ефективною стратегією пом'якшення. Також було відзначено ефективність спроби зменшити перешкоди для реєстрації та поставлення запитань.

Проблеми, виявлені під час співбесід. Деякі проблеми, що виникли під час інтерв'ю, не були відображені в літературі. Пошук репрезентативних користувачів для зворотного зв'язку був проблемою, про яку згадували багато учасників. Існує упередженість відбору, яку необхідно враховувати під час аналізу зворотного зв'язку. Команда повинна знати, звідки надходить зворотний зв'язок, що не завжди так. Інші питання включають те, наскільки безпосередньою має бути взаємодія між розробниками та користувачами, а також визначення окупності інвестицій у витрачений час та ресурси на процеси зворотного зв'язку. Ці проблеми були виявлені під час інтерв'ю та варто зазначити як потенційний внесок для майбутніх досліджень.

Рольові перспективи щодо відгуків користувачів. Інтерв'ю показують, що хоча врахування відгуків користувачів загалом вважається важливим у гнучких організаціях, конкретні уявлення про те, як це має відбуватися, різняться залежно від ролі в компанії.

Розробник. Розробники часто хочуть отримувати ранній та прямий зворотний зв'язок від користувачів. Вони згадують такі речі, як «прототипи», «договірне тестування» або спілкування з реальними користувачами. Їхня головна мета - якомога раніше зрозуміти, що насправді потрібно користувачам, щоб вони могли швидко створювати потрібні речі. Але вони також кажуть, що зворотний зв'язок часто змінюється або неправильно розуміється в процесі. Респондент 1 стверджує: «вони можуть говорити речі, які насправді не мали на

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

увазі». Як результат, розробники надають перевагу прямому спілкуванню з користувачами, щоб уникнути неправильного тлумачення.

Scrum-майстер та Agile-коуч. Scrum-майстри та Agile-коучі загалом погоджуються, що пряме спілкування між розробниками та користувачами може бути корисним. Однак вони також вказують на потенційні проблеми. Неструктурований контакт, стверджують вони, може призвести до втрати часу або опору в організації. Як зазначив респондент 9: «Часто цього неможливо зробити, тому що якщо користувач знає: «О, це розробник», то він зв'яжеться з ним безпосередньо, і тоді розробники ніколи не матимуть часу фактично працювати та розробляти речі». З цієї причини вони віддають перевагу більш структурованим форматам, таким як залучення спеціальної особи з комунікацій або використання таких процесів, як системи заявок. Респондент 3 також наголосив: «Проблема полягає в тому, що не можна бути хорошим у всьому». Це підкреслює поширену думку, що хоча розробники є висококваліфікованими у виконанні технічних завдань, вони не завжди можуть бути однаково сильними у спілкуванні з користувачами або в точному тлумаченні потреб користувачів.

Власник продукту. Власники продукту вважають, що користувачі повинні бути в центрі розробки продукту, як висловився респондент 7: «все має бути зосереджено навколо користувача». Водночас вони часто відповідають за пріоритезацію, економічну ефективність та підтримку стратегічної перспективи. У цій ролі їм потрібно фільтрувати, узгоджувати та оцінювати відгуки, що іноді може суперечити бажанню розробників щодо швидкого впровадження. Хоча розробники сподіваються на чіткі та прямі вимоги, власники продукту усвідомлюють, що користувачі часто не знають точно, що їм потрібно. Тому вони покладаються на такі інструменти, як карти подорожей та дизайн-спринти, перевіряють ідеї на ранній стадії та прагнуть збирати відгуки поетапно. Як пояснює респондент 7: «ви не можете зробити все, що вони хочуть, тому що це коштуватиме вам занадто дорого». Респондент 8 додає: «одна сторона думає про те, що ми хочемо зробити, і приділяє час плануванню наперед, щоб уникнути затримок та зміни напрямку».

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Внесок у дослідження. Розглядаючи практичні стратегії та проблеми, виявлені в інтерв'ю, це дослідження пропонує нове розуміння того, як цикли зворотного зв'язку з користувачами реалізуються та оптимізуються в гнучкій розробці програмного забезпечення. Ключовим внеском цього дослідження є всебічне вивчення конкретних методів, які гнучкі команди використовують для інтеграції зворотного зв'язку з користувачами на практиці. Попередні дослідження визнавали важливість зворотного зв'язку з користувачами, але часто зосереджувалися переважно на проблемах або надавали лише вузькі чи поодинокі рекомендації.

Це дослідження виходить за рамки загальних та теоретичних підходів, включаючи висновки з інтерв'ю, зосереджуючись на реальних практиках та стратегіях. Воно надає детальний огляд практик та визначає дієві стратегії, такі як зосередження на швидкому та ранньому зворотному зв'язку, глибоке залучення користувачів та практична інтерпретація зворотного зв'язку. Ці висновки є цінними для того, щоб допомогти гнучким командам створювати ефективні цикли зворотного зв'язку, що відповідають гнучким принципам - ітеративної розробки та постійного вдосконалення.

Це дослідження не лише визначає рішення та стратегії, але й виклики, з якими стикаються Agile-команди під час впровадження цих циклів зворотного зв'язку. Виявлені виклики узгоджуються з висновками, наведеними в літературі. Крім того, це дослідження пропонує конкретні стратегії пом'якшення цих викликів, такі як раннє залучення, орієнтація на певні групи користувачів та використання посередників, таких як менеджери продуктів або бізнес-аналітики, для пом'якшення цих викликів.

Якість та обмеження досліджень. Щоб забезпечити високу якість дослідження в нашому дослідженні, ми дотримувалися найважливіших аспектів роботи Денскомба (2010). Центральним моментом є достовірність (валідність).

Основна увага приділялася узгодженню дослідницьких питань з цілями дослідження та забезпеченню того, щоб питання інтерв'ю фіксували відповідну інформацію, що стосується дослідницької проблеми. Напівструктурований

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

формат дозволив нам дослідити нові теми, що виникали під час інтерв'ю, гарантуючи, що результати є достовірними та відображають досвід учасників. Узгоджений та стандартизований посібник з інтерв'ю також допомагає підтримувати надійність (достовірність). Такий підхід допоміг забезпечити узгодженість інтерв'ю, залишаючи при цьому достатню гнучкість для індивідуальних відповідей.

Переносність (узагальнюваність) є потенційним обмеженням дослідження, оскільки розмір вибірки становив дев'ять учасників, а учасники були обмежені Швецією, Німеччиною, Швейцарією, Індією та Новою Зеландією. Хоча результати можуть бути статистично неузагальнюваними, ми обов'язково включили фахівців з різних ролей у гнучкій розробці та різні статі, щоб збільшити різноманітність точок зору. Ці перспективи дали ширше розуміння культурного та організаційного контексту цих країн.

Щодо підтверджуваності (об'єктивності), співпраця двох дослідників підвищила об'єктивність дослідження. Обидва дослідники брали активну участь у відборі учасників, зборі даних та процесі аналізу. Працюючи разом, ми забезпечили врахування різних точок зору, а регулярні обговорення дозволили нам поміркувати над потенційними упередженнями та мінімізувати суб'єктивний вплив. Цей процес забезпечив об'єктивність результатів дослідження та їхню основу на документації та прозорому аналізі. Хоча сувора відтворюваність неможлива при використанні напівструктурованих інтерв'ю, це не впливає на якість дослідження.

Етичні та соціальні наслідки

Етичні міркування були ретельно враховані, забезпечивши інформовану згоду, добровільну участь та сувору анонімізацію всіх зібраних даних. Інструменти штучного інтелекту та інші ІТ-інструменти використовувалися обережно та відповідально відповідно до політики ШІ Стокгольмського університету. Щодо суспільних наслідків, дослідження сприяє покращенню гнучких практик, визначаючи практичні рішення для інтеграції відгуків користувачів. Хоча існують ризики, пов'язані з обробкою персональних даних,

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

було вжито всіх необхідних запобіжних заходів, щоб максимально мінімізувати ризики. В результаті цього дослідження не було виявлено жодних суттєвих суспільних ризиків.

Використання ІТ та інструментів штучного інтелекту

Цифрові інструменти були ключовими для підвищення ефективності та організованості дослідження. Комунікаційні платформи, такі як електронна пошта, WhatsApp, LinkedIn та Zoom, дозволяють проводити інтерв'ю онлайн. Такий підхід заощадив час і забезпечив більшу гнучкість. Однак, оскільки інтерв'ю не проводилися віч-на-віч, було важливо створити дружню та відкриту атмосферу на початку. Також виникають деякі труднощі, зокрема, періодичні проблеми з підключенням до Інтернету під час одного з інтерв'ю. Крім того, спілкування через чат здавалося менш особистим, що іноді призводило до скасування інтерв'ю.

Такі інструменти, як Google Docs, Google Sheets та Zotero, були безцінними для організації та аналізу даних. Ці інструменти покращують ведення нотаток та управління джерелами, спрощуючи співпрацю завдяки хмарному сховищу та запобігаючи втраті даних. Водночас захист особистої інформації учасників був надзвичайно важливим. Тому дані були анонімізовані перед аналізом, а доступ до документів був обмежений.

Інструменти штучного інтелекту також були корисними на певних етапах процесу. Seshat Audio Transcriber заощадив час під час транскрипції, але тексти все одно потрібно було перевіряти вручну, щоб уникнути помилок і забезпечити правильне передавання запланованого значення. Такі інструменти, як ChatGPT, DeepL Write та Grammarly, покращили чіткість і читабельність тексту. Однак, використовуючи ці інструменти, слід бути обережним, щоб зберегти оригінальність роботи та тексту. Вони використовувалися виключно як допоміжні засоби для письма, а не як генератори контенту, щоб зберегти автентичність та дотримуватися рекомендацій DSV.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Перспективи розвитку та напрями подальших досліджень

Це дослідження забезпечує міцну основу для подальших досліджень - циклів зворотного зв'язку з користувачами в гнучких організаціях. Для поглиблення та розширення поточних висновків можна було б дослідити кілька напрямків.

Кількісний підхід. Кількісне дослідження, що охоплює більшу та різноманітнішу вибірку, може допомогти перевірити та розширити результати цього дослідження. Такий підхід дозволить детальніше дослідити різні точки зору в різних ролях і може виявити закономірності або суперечності, які не видно в якісних інтерв'ю. Крім того, це надасть можливість вивчити потенційні відмінності між країнами та регіонами.

Тематичне дослідження. Поглиблене дослідження конкретного випадку в реальній організації може запропонувати цінну інформацію про те, як стратегії зворотного зв'язку з користувачами впроваджуються на практиці. Таке дослідження може допомогти зрозуміти, як ці стратегії функціонують у різних організаційних умовах, зокрема порівнюючи невеликі та великомасштабні гнучкі фреймворки.

Порівняння фреймворків. Майбутні дослідження також можуть дослідити, як різні гнучкі фреймворки, такі як Scrum, Kanban або Scaled Agile Framework (SAFe), впливають на використання та ефективність циклів зворотного зв'язку. Це може допомогти зрозуміти, як конкретні методології впливають на інтеграцію зворотного зв'язку від користувачів у процеси розробки.

Новітні технології. Інший можливий напрямок досліджень включає вивчення ролі нових технологій, таких як інструменти на основі штучного інтелекту для автоматизації або аналізу відгуків користувачів. Ці технології можуть покращити процеси зворотного зв'язку та підвищити ефективність і точність циклів зворотного зв'язку в гнучких середовищах.

У дослідженні розглянуто поширені проблеми впровадження циклів зворотного зв'язку з користувачами в гнучкій розробці програмного

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечення шляхом аналізу відповідної літератури. Крім того, було використано якісний дослідницький підхід, проведений з дев'ятьма інтерв'ю з досвідченими практиками гнучких методологій. Інтерв'ю дозволили дослідженню отримати розуміння того, як зворотний зв'язок з користувачами успішно інтегрується в реальні умови та як долаються труднощі. Результати мають на меті допомогти компаніям та практикам гнучких методологій у вдосконаленні їхніх процесів гнучкої розробки шляхом ефективного впровадження циклів зворотного зв'язку з користувачами, що зрештою забезпечує більш ефективну та результативну роботу. Хоча деякі перспективи окреслені за рольовими групами, дослідження недостатньо вичерпне, щоб зробити будь-які висновки щодо відмінностей, специфічних для конкретних ролей. На підтримку цього надається практичний посібник, який пропонує конкретні рекомендації щодо інтеграції зворотного зв'язку з користувачами в гнучкі робочі процеси.

Короткий виклад результатів дослідження, пов'язаних з дослідницькими питаннями

Результати цього дослідження безпосередньо відповідають на поставлені дослідницькі питання.

1. Які поширені проблеми впровадження циклів зворотного зв'язку з користувачами були виявлені в літературі?

Аналіз літератури з розширеного вивчення питання виявив кілька проблем, пов'язаних із впровадженням циклів зворотного зв'язку. До них належать затримки, неефективність та невідповідність між продуктом та потребами користувачів.

Тривалі цикли зворотного зв'язку, повільне надання відгуків та негнучкі клієнти також створюють проблеми в процесі зворотного зв'язку. Інші труднощі включають заохочення користувачів до співпраці, реагування на нечіткі або розпливчасті відгуки та подолання організаційних бар'єрів. Високе робоче навантаження та неадекватні інструменти також були названі як проблеми.

Крім того, проблеми з методами та інструментами валідації можуть

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

призвести до зниження якості продукту, слабкого аналізу даних, поганого розуміння використання функцій та збільшення робочого навантаження через необхідність постійного зворотного зв'язку. Ще однією значною проблемою є комунікація, яка може призвести до непорозумінь, нечітких цілей проекту, низького рівня залученості зацікавлених сторін, недостатньої мотивації користувачів та негативного ставлення. Ці висновки з літератури стосуються дослідницького питання, визначаючи ключові труднощі у впровадженні циклів зворотного зв'язку з користувачами. Інтерв'ю додатково підтвердили та підтримали ці висновки, забезпечивши міцну основу для розуміння основних перешкод у процесі зворотного зв'язку.

2. Як на практиці впроваджуються рішення для покращення циклів зворотного зв'язку з користувачами в гнучкій розробці програмного забезпечення, і як пом'якшуються проблеми впровадження?

Це питання було розглянуто шляхом проведення інтерв'ю з фахівцями з Agile. Ключові результати виявили кілька важливих висновків. Одним із висновків було те, що команди впроваджують цикли зворотного зв'язку з користувачами, наголошуючи на швидкому та ранньому зворотному зв'язку для перевірки ідей та швидкого зменшення витрачених зусиль. Для досягнення цієї мети вони використовують прості, недорогі методи, такі як ескізи та прототипи, забезпечуючи належне планування, щоб сесії зворотного зв'язку були цілеспрямованими та цінними. Крім того, глибока участь користувачів виявилася важливою.

Практичний посібник

Цей практичний посібник розроблений для підтримки та покращення зворотного зв'язку з користувачами.

цикл у гнучкій розробці.

1. Збирайте ранні та недорогі відгуки

- Використовуйте ескізи, прототипи та макетні схеми для ранньої перевірки ідей.

- Плануйте сесії зворотного зв'язку свідомо та з чіткою метою.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

- Покладайтеся на ітерації та швидке навчання, щоб «швидко зазнавати невдач» та уникати непотрібної розробки .
- 2. Глибоко залучайте реальних кінцевих користувачів
 - Залучайте кінцевих користувачів до процесу зворотного зв'язку, а не лише зацікавлених сторін.
 - Використовуйте такі методи, як спостереження за користувачами або семінари.
 - Використовуйте посередницькі ролі, такі як власник продукту або бізнес-аналітик, коли пряме спілкування неможливе.
- 3. Стратегічно інтерпретуйте та фільтруйте зворотний зв'язок
 - Не всі відгуки однаково цінні; фільтруйте їх вибірково.
 - Класифікуйте відгуки як помилки, запити на нові функції або непорозуміння.
 - Переконайтеся, що відгуки є репрезентативними та не обмежуються невеликою групою досвідчених користувачів.
- 4. Вирішення поширених проблем
 - Зробіть процеси зворотного зв'язку простими та легкими, щоб уникнути організаційних бар'єрів.
 - Заохочуйте користувачів залишати відгуки, пропонуючи винагороди, але переконайтеся, що відгуки є репрезентативними.
 - Розподіліть завдання зворотного зв'язку між різними ролями та інструментами для обробки великих робочих навантажень.
 - Підтримуйте коротке та чітке спілкування, а також користуйтеся помічниками, щоб уникнути проблем у спілкуванні.
- 5. Використовуйте інструменти з чіткою метою
 - Автоматизація, інструменти UX та штучний інтелект можуть допомогти з аналізом відгуків.

Інструменти самі по собі не вирішують проблеми зворотного зв'язку. Їх правильне використання має вирішальне значення.

Хоча деякі команди наголошували на прямому контакті між

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

розробниками та кінцевими користувачами, щоб краще зрозуміти їхні потреби, більшість покладається на посередницькі ролі, такі як власники продуктів або бізнес-аналітики, для управління комунікацією та запобігання перевантаженню розробників. Інший висновок полягає в тому, що ефективна інтерпретація та фільтрація зворотного зв'язку є вирішальними. Командам необхідно визначити, який зворотний зв'язок є дієвим та репрезентативним, щоб розумно розподіляти ресурси розробки. Проблеми, з якими часто стикаються під час впровадження, включають організаційні затримки, обмежену мотивацію користувачів, високе робоче навантаження та комунікаційні бар'єри, спричинені кількома рівнями між користувачами та розробниками. Стратегії пом'якшення включають залучення користувачів на ранній стадії, орієнтацію на мотивовані групи користувачів, призначення спеціальних ролей для сприяння комунікації та використання доступних інструментів для управління робочими навантаженнями. Ці висновки стосуються того, як рішення для покращення циклів зворотного зв'язку з користувачами впроваджуються на практиці та як пом'якшуються проблеми під час гнучкої розробки програмного забезпечення.

3.4 Висновок по розділу

У третьому розділі було розглянуто практичну реалізацію та оцінку ефективності циклів швидкого зворотного зв'язку в програмних продуктах. Проведено реалізацію механізмів швидкого зворотного зв'язку, що забезпечують оперативне реагування на зміни та потреби користувачів. Проаналізовано результати впровадження таких механізмів та їх вплив на якість програмного забезпечення, зокрема на стабільність, продуктивність і зручність використання системи. Також визначено перспективи розвитку та напрями подальших досліджень у сфері організації швидкого зворотного зв'язку. Отримані результати підтверджують доцільність використання швидких циклів зворотного зв'язку для підвищення ефективності процесів розробки програмного забезпечення.

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання бакалаврської роботи було досліджено підходи до розробки програмних продуктів із врахуванням циклу швидкого зворотного зв'язку, що дозволяє суттєво підвищити ефективність процесу створення програмного забезпечення, скоротити час реакції на зміни вимог користувачів та покращити якість кінцевого продукту. У роботі проаналізовано сучасні підходи до організації розробки, зокрема Agile, SSDLC та DevSecOps, а також їх роль у забезпеченні безперервної взаємодії між користувачем і командою розробки. Окрему увагу приділено інструментам збору та аналізу зворотного зв'язку, що дозволяють приймати обґрунтовані рішення на основі реальних даних використання системи.

У процесі дослідження було розглянуто особливості взаємодії людини з комп'ютером як ключового джерела зворотного зв'язку. Це дозволило визначити, що саме поведінкові дані користувачів, їхні дії в системі та результати використання функціоналу формують основу для безперервного вдосконалення програмного продукту. Було також проаналізовано обмеження традиційних підходів до розробки, які не забезпечують достатньо швидкої реакції на зміни вимог та часто призводять до накопичення технічного боргу.

Особливу увагу в роботі приділено методам збору даних зворотного зв'язку та їх автоматизації. Було розглянуто підходи до інтеграції аналітичних інструментів у програмні продукти, що дозволяє в реальному часі відстежувати поведінку користувачів, виявляти проблемні сценарії використання та формувати інсайти для подальших покращень. Використання сучасних засобів аналітики програмного забезпечення дозволяє значно підвищити точність прийняття рішень у процесі розвитку продукту.

У роботі також було запропоновано підхід до впровадження циклу швидкого зворотного зв'язку в процес розробки, який передбачає інтеграцію етапів аналізу, розробки, тестування та збору даних у єдиний безперервний процес. Це забезпечує можливість швидкого внесення змін у функціональність

					БР.ІІІ - 65.00.00.000 ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

системи та оперативного реагування на потреби користувачів без значних витрат часу та ресурсів.

Проведене дослідження показало, що застосування циклу швидкого зворотного зв'язку дозволяє суттєво підвищити гнучкість процесу розробки, покращити якість програмних продуктів та забезпечити більш тісну взаємодію між розробниками і кінцевими користувачами. Водночас було виявлено, що ефективність такого підходу значною мірою залежить від рівня автоматизації процесів збору та аналізу даних, а також від зрілості використовуваних інструментів.

Загалом, результати роботи підтверджують доцільність використання підходів, орієнтованих на швидкий зворотний зв'язок, як одного з ключових факторів підвищення якості та конкурентоспроможності сучасних програмних продуктів. У перспективі подальші дослідження можуть бути спрямовані на розширення методів автоматичного аналізу поведінки користувачів та інтеграцію штучного інтелекту для прогнозування потреб користувачів і автоматичного вдосконалення програмних систем.

					БР.ІІ - 65.00.00.000 ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Angular Documentation. (2025). <https://angular.io/docs>
2. Azure Application Insights Overview. (2025). <https://learn.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview>
3. Azure DevOps Documentation. (2025). <https://learn.microsoft.com/en-us/azure/devops/>
4. Azure Functions Documentation. (2025). <https://learn.microsoft.com/en-us/azure/azure-functions/>
5. Beck, K. (2001). *Manifesto for Agile Software Development*. <https://agilemanifesto.org/>
6. Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice*. Addison-Wesley.
7. Bootstrap Documentation. (2025). <https://getbootstrap.com/docs/>
8. Cockburn, A. (2002). *Agile Software Development*. Addison-Wesley.
9. DORA. (2024). *Accelerate State of DevOps Report*. <https://dora.dev/>
10. Fowler, M. (2019). *Refactoring: Improving the Design of Existing Code*. <https://martinfowler.com/>
11. Freeman, E., & Robson, E. (2021). *Head First Design Patterns*. O'Reilly Media.
12. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns*. Pearson.
13. GitHub Actions Documentation. (2025). <https://docs.github.com/actions>
14. Google Lighthouse Documentation. (2025). <https://developer.chrome.com/docs/lighthouse/>
15. Google UX Research. (2024). <https://design.google/library/ux-research/>
16. Green Software Foundation. (2024). <https://greensoftware.foundation/>
17. Hassan, M., & Khan, A. (2023). Real-time feedback systems in web applications. *IEEE Access*, 11, 11234–11250.
18. IBM DevOps Guide. (2024). <https://www.ibm.com/cloud/learn/devops>

					БР.ІІІ - 65.00.00.000 ІІЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

19. ISO/IEC 25010:2011. Systems and software quality models.
20. ISO/IEC 12207:2017. Software lifecycle processes.
21. Jira Software Documentation. (2025).
<https://www.atlassian.com/software/jira>
22. Kubernetes Documentation. (2025). <https://kubernetes.io/docs/>
23. Martin, R. C. (2017). *Clean Architecture*. Pearson.
24. Microsoft Azure Documentation. (2025). <https://learn.microsoft.com/en-us/azure/>
25. MongoDB Documentation. (2025). <https://www.mongodb.com/docs/>
26. .NET Best Practices. (2023). <https://learn.microsoft.com/en-us/dotnet/>
27. Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann.
28. OWASP Top Ten. (2024). <https://owasp.org/www-project-top-ten/>
29. Postman API Testing Guide. (2025). <https://learning.postman.com/docs/>
30. React Documentation. (2025). <https://react.dev/>
31. Scrum Guide. (2020). <https://scrumguides.org/>
32. Selenium Documentation. (2025).
<https://www.selenium.dev/documentation/>
33. Stack Overflow Developer Survey. (2025).
<https://survey.stackoverflow.co/>
34. Sutherland, J. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*.
35. Thomas, D., & Hunt, A. (2019). *The Pragmatic Programmer*. Addison-Wesley.
36. Visual Studio Dev Tools. (2025). <https://visualstudio.microsoft.com/>
37. Wang, L., & Zhang, Y. (2025). Continuous feedback in web systems. *ACM Computing Surveys*, 57(1), 1–28.
38. Web Vitals Guide. (2024). <https://web.dev/vitals/>
39. World Wide Web Consortium (W3C). (2025). <https://www.w3.org/>
40. YouTrack Agile Management. (2025).
<https://www.jetbrains.com/youtrack/>

					БР.ІІІ - 65.00.00.000 ІІЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: " Розробка програмних продуктів з
урахуванням циклу швидкого зворотного зв'язку "

Обсяг пояснювальної записки: 81 аркушів

Дата закінчення бакалаврської роботи 10 червня 2026р.

Підпис студента _____