

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Проект метрик Agile-процесів для малих компаній
з розробки програмного забезпечення.

Виконав(ла): студент(ка) 4 курсу, групи СНС-41
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Туріч В.А.

(прізвище та ініціали)

Керівник

(підпис)

Липак Г.І.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Хоміцький Б.В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Боднарчук І.О.
(підпис) (прізвище та ініціали)

«22» червня 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

студенту Туріч Владислав Андрійович
(прізвище, ім'я, по батькові)

1. Тема роботи Проект метрик Agile-процесів для малих компаній з розробки програмного забезпечення

Керівник роботи к.т.н., доц. Липак Г.І.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 22 червня 2023 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

ВСТУП РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ МОНІТОРИНКУ AGILE-ПРОЕКТІВ 1.1 Формулювання проблеми 1.2 Проект Q-Rapids 1.3 Аналіз публікацій на тематику кваліфікаційної роботи 1.4 Методологія дослідження РОЗДІЛ 2. ФОРМУВАННЯ СИСТЕМИ МЕТРИК ПРОЕКТУ 2.1 Визначення метрик процесу 2.2 Оцінка метрик процесу 2.3 Оцінка завдань (тасків) 2.4 Вибір показників 2.5 Загальна оцінка метрик процесу РОЗДІЛ 3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 3.1 Аналіз небезпеки і шкідливості при розробці програмного забезпечення 3.2 Інформаційно-психологічні небезпеки ВИСНОВОК ПЕРЕЛІК ПОСИЛАНЬ

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

1.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., к.т.н., доц.		

7. Дата видачі завдання 24 січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	24.01.23-27.01.23	<i>Виконано</i>
2.	Підбір джерел по темі роботи	28.01.23 – 01.04.23	<i>Виконано</i>
3.	Оформлення першого розділу	15.04.2023	<i>Виконано</i>
4.	Оформлення другого розділу	30.04.2023	<i>Виконано</i>
5.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	15.05.2023	<i>Виконано</i>
6.	Оформлення кваліфікаційної роботи	07.06.2023	<i>Виконано</i>
7.	Перевірка на плагіат	07.06.2023	<i>Виконано</i>
8.	Нормоконтроль	09.06.2023	<i>Виконано</i>
9.	Попередній захист кваліфікаційної роботи	11.06.2023	<i>Виконано</i>
10.	Захист кваліфікаційної роботи	22.06.2023	

Студент

(підпис)

Туріч В.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Липак Г.І.

(прізвище та ініціали)

АНОТАЦІЯ

Проект метрик Agile-процесів для малих компаній з розробки програмного забезпечення // Кваліфікаційна робота освітнього рівня "Бакалавр" // Туріч Владислав Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // с. – 49, рис. – 10, табл. – 2, кресл. – 10, бібліогр. – 29.

Ключові слова: програмна інженерія, швидка розробка програмного забезпечення, гнучка розробка програмного забезпечення, показники процесів, якість програмного забезпечення, МСП.

Гнучка розробка стала загальноприйнятою практикою для компаній, що займаються розробкою програмного забезпечення, завдяки її численним перевагам. Однак, виконання гнучких проектів вимагає від малих і середніх підприємств (МСП) особливої уваги, оскільки вони мають бути швидкими, але при цьому задовольняти вимоги клієнтів до якості.

Тому метою цієї роботи є аналіз практичного досвіду використання метрик, пов'язаних з процесом розробки програмного забезпечення, як засобу підтримки МСП в гнучкій розробці за методологією Agile. Було розроблено та проведено дослідницький протокол, який відповідає потребам компанії, використовуючи пілотний проект. Результатом є множина метрик для гнучкої розробки за методологією Agile, які були практично перевірені в контексті невеликої компанії, що займається розробкою програмного забезпечення, і прийняті компанією для використання у своїх Agile-проектах.

Отже, інші розробники можуть використовувати цей каталог метрик у своїх Agile-проектах, зокрема якщо вони працюють у сфері МСП, і адаптувати

їх до своїх потреб та інструментів. Отримані дані можуть бути використані як основа для подальших досліджень, включаючи емпіричні дослідження.

ANNOTATION

Agile-Processes Metrics Project for Small Software Development Companies // Qualification work of the educational level "Bachelor" // Vladyslav Turich // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group CHc-41 // Ternopil, 2023 // p. – 49 , fig. – 10, tables – 2, references – 29, posters – 10.

Keywords: software engineering, rapid software development, Agile software development, process metrics, software quality, SMEs

Agile development has become an accepted practice for software development companies due to its many benefits. However, the implementation of flexible projects requires special attention from small and medium enterprises (SMEs), as they must be fast, but at the same time meet the quality requirements of the customers.

Therefore, the purpose of this work is to analyze the practical experience of using metrics related to the software development process as a means of supporting SMEs in flexible development according to the Agile methodology. A research protocol tailored to the needs of the company was developed and conducted using a pilot project. The result is a set of metrics for flexible Agile development that have been field-tested in the context of a small software development company and adopted by the company for use in its Agile projects.

So, other developers can use this catalog of metrics in their Agile projects, especially if they work in the SME field, and adapt them to their needs and tools. The obtained data can be used as a basis for further research, including empirical ones.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ МОНІТОРИНКУ AGILE-ПРОЕКТІВ	9
1.1 Формулювання проблеми.....	9
1.2 Проект Q-Rapids	10
1.3 Аналіз публікацій на тематику кваліфікаційної роботи	12
1.4 Методологія дослідження	16
1.1.1 Мета та питання дослідження.....	18
1.1.2 Інструментальне забезпечення дослідження.....	18
РОЗДІЛ 2. ФОРМУВАННЯ СИСТЕМИ МЕТРИК ПРОЕКТУ	22
2.1 Визначення метрик процесу	22
2.2 Оцінка метрик процесу.....	25
2.2.1 Продуктивність процесу розробки.....	25
2.2.2 Якість продукту	28
2.3 Оцінка завдань (тасків).....	28
2.4 Вибір показників	30
2.5 Загальна оцінка метрик процесу.....	31
РОЗДІЛ 3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	36
3.1 Аналіз небезпеки і шкідливості при розробці програмного забезпечення	36
3.2 Інформаційно-психологічні небезпеки	38
ВИСНОВОК.....	43
ПЕРЕЛІК ПОСИЛАНЬ	45

ВСТУП

Методології гнучкої розробки сьогодні широко застосовуються компаніями, які розробляють програмне забезпечення [1]. Галузеві опитування показують, що практично всі організації певною мірою використовують методи Agile, і більше половини з них використовують Agile як звичайний підхід до розробки програмного забезпечення. Практики повідомляють про багато переваг, починаючи від скорочення часу виходу на ринок до підвищення задоволеності клієнтів і зниження витрат на розробку, серед іншого. Однак управління гнучкими проектами може бути складним [2], особливо у випадку малих і середніх підприємств (МСП). Завдання для власника продукту та Scrum-майстра полягає щонайменше в двох аспектах: забезпечити якість програмного продукту та сприяти ефективності команди та процесу.

Слід зазначити, що в ТНТУ також займаються проблемами управління Agile-проектів. Наприклад в роботах [3, 4, 5] обговорюються проблеми управління якістю в гнучких технологіях розробки та для розподілених команд. В роботі [6] виконано огляд проблем в гнучких проектах для великих розподілених команд. В роботі [7] обговорюються задачі переходу на цифрові технології ведення бізнесу для малих та середніх підприємств.

Зараз у багатьох компаніях, які займаються розробкою програмного забезпечення, команди використовують різні спеціальні інструменти (такі як Jira, GitLab і SonarQube), щоб підтримувати процес розробки та якість коду та продуктів. Зазвичай це відбувається на регулярній ретроспективній зустрічі, в якій бере участь уся команда. Що стосується якості коду, ці інструменти надають достатньо інформації для команди Scrum. Проте все ще існує прогалина та потреба в додаткових рішеннях, які відображатимуть ефективність команди та якість процесу. Можна сказати, що в даний час заходи з удосконалення процесів в основному базуються на уявленнях розробників і мало підтримки надається для прийняття мудрих рішень на основі даних.

Основний результат цієї роботи полягає у формі набору показників, які вимірюють процес розробки програмного забезпечення Agile (який ми надалі називаємо показниками процесу) у компанії малого та середнього бізнесу, а також обговорення того, як ці показники допомогли команді Scrum у розробці комерційного продукту.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ МОНІТОРИНКУ AGILE-ПРОЕКТІВ

1.1 Формулювання проблеми

Наукова література показує, що вимірювання є невід'ємною частиною розуміння, прогнозування та оцінки проектів розробки програмного забезпечення [8]. Розробка програмного забезпечення включає в себе багато процесів, і вимірювання дозволяє нам характеризувати, контролювати, прогнозувати та вдосконалювати ці процеси. Будучи діяльністю, зосередженою на людині, процеси програмного забезпечення схильні до проблем [9], що додає додаткової впевненості в тому, чому їх слід постійно оцінювати та вдосконалювати, щоб відповідати очікуванням клієнтів і зацікавлених сторін організації.

Вимірювання процесу програмного забезпечення може допомогти в досягненні бажаного рівня продуктивності, можливостей і якості. Крім того, вимірювання програмних процесів також дозволяє дізнатися про якість програмного продукту.

Зростаюча популярність Agile розробки програмного забезпечення (ASD) робить розуміння показників програмного забезпечення в контексті Agile більш актуальним. Дослідження визнають необхідність гнучких організацій використовувати метрики, але емпіричні дослідження метрик у промисловій ASD залишаються рідкісними [10]. Зокрема, мало уваги приділялося обґрунтуванню метрик, згаданих у літературі (наприклад, діаграми «згорання», показники проходження тесту та відповідний темп) і тому, як вони фактично використовуються на практиці.

Крім того, незважаючи на те, що мета вимірювання в ASD подібні до традиційних підходів (тобто планування та відстеження гнучких спринтів або циклів, моніторинг якості продукції, а також виявлення та усунення проблем, пов'язаних з процесом), програми вимірювання досить різні на практиці.

Орієнтація Agile на легкі практики, безперервне постачання робочого програмного забезпечення, гнучкі фази розробки та мінімальну документацію роблять необхідні для того, щоб програми вимірювань були добре узгоджені з мисленням Agile та принципом простоти. У конкретному випадку метрики процесу розробки є складними та нематеріальними, що робить таке вимірювання складним завданням на практиці [11]. Крім того, через часові, бюджетні та ресурсні обмеження вимірювання програмного забезпечення рясніє проблемами, особливо в малих і середніх підприємствах [12].

З огляду на ресурси [12], методи вибору показників [13], інфраструктурні засоби, розмір команди і добре сплановану програму вимірювання програмного забезпечення, показники процесу можуть допомогти малим і середнім підприємствам у вимірюванні та покращенні ефективності їх процесу.

1.2 Проект Q-Rapids

Q-Rapids був спільним проектом виробництва та академічної спільноти (фінансувався Європейською комісією в рамках H2020 Framework), в якому брали участь три дослідницькі партнери та чотири компанії. Він запропонував інноваційні методи та інструменти для підтримки індустрії розробки програмного забезпечення у покращенні рівня якості (програмного забезпечення та процесу) при використанні Agile та Rapid розробки програмного забезпечення [14]. Усі партнери працювали разом за стратегією спільної розробки. Крім того, кожна компанія адаптувала результати в міру їх отримання до своїх конкретних потреб.

Підхід Q-Rapids заснований на зборі та аналізі даних з кількох джерел (репозиторії програмного забезпечення, інструменти управління проектами, використання системи та якість обслуговування). Дані агрегуються в індикатори якості, які надаються різним зацікавленим сторонам за допомогою інструменту Q-Rapids.

Інструмент Q-Rapids, як результат проекту, забезпечує постійну оцінку пов'язаних з якістю стратегічних показників для осіб, які приймають рішення. На рисунку 1.1 показано фрагмент концептуальної архітектури інструменту. Основними модулями є збір даних, моделювання й аналіз даних і прийняття стратегічних рішень.

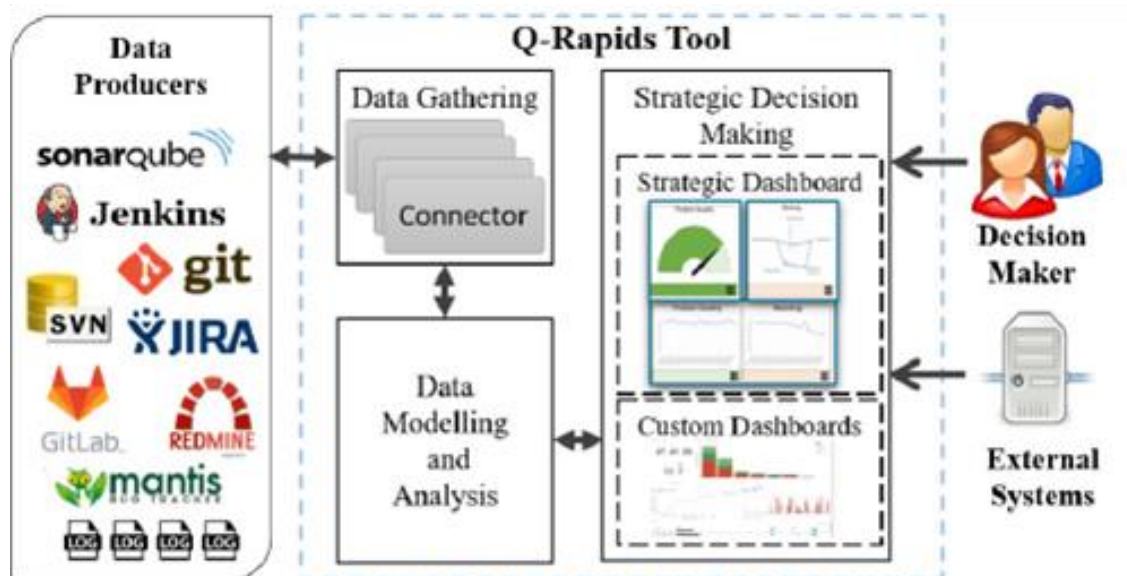


Рисунок 1.1 – Концептуальна архітектура інструменту Q-Rapids

Модуль збору даних складається з різних конекторів Apache Kafka, які дозволяють збирати дані з різномірних зовнішніх джерел даних, таких як статичний аналіз коду (наприклад, SonarQube), інструменти постійної інтеграції (наприклад, Jenkins), сховища коду (наприклад, SVN, Git, GitLab), інструменти відстеження проблем (наприклад, Redmine, GitLab, JIRA, Mantis) і журнали використання.

Модуль моделювання та аналізу даних обробляє дані для оцінки якості програмного забезпечення (продукту та процесу). Якщо говорити конкретно, показники обчислюються на основі зібраних даних. Ці показники об'єднуються в фактори якості, пов'язані з розробкою та використанням. Нарешті, фактори якості агрегуються в показники високого рівня, які називаються стратегічними показниками, які можна узгодити зі стратегічними цілями організації.

Дані оцінки, оброблені модулем моделювання та аналізу даних, візуалізуються на інформаційних панелях, включених до модуля стратегічного прийняття рішень. Дані візуалізуються як зведені для кінцевого користувача через веб-графічний інтерфейс, який також називається Strategic Dashboard. Стратегічна інформаційна панель також містить посилання на налаштовані інформаційні панелі, які можна розробити для візуалізації спеціальних діаграм, включаючи дані, отримані з інструментів виробників даних.

Ця інформаційна панель дозволяє користувачеві відображати ці дані, розраховані для поточного етапу проекту, а також еволюцію метрик, факторів і стратегічних показників з часом. Ще одна конфігурована властивість візуалізації даних – це можливість налаштувати сітку часових діаграм відповідно до поточних потреб і представити еволюцію даних із деталізацією від днів до місяців. Інформаційна панель також дозволяє переміщатися між різними елементами, що забезпечує відстеження та забезпечує розуміння оцінки.

Інші інструменти аналітики програмного забезпечення, подібні до Q-Rapids, нещодавно з'явилися у сфері розробки програмного забезпечення. Деякі з них залежать від домену, наприклад, метрична структура Європейського співробітництва з космічної стандартизації (ECSS) для підвищення прозорості розробки програмного забезпечення у відносинах клієнт-постачальник космічних місій [15]. Крім того, на ринку доступні деякі комерційні інструменти з такими ж характеристиками, як інструмент Q-Rapids. Наприклад, Square надає інформаційну панель, подібну до Q-Rapids, і включає кілька метрик програмного забезпечення та індикаторів, які вимірюють якість програмного забезпечення, хоча вони не порівнюються з набором показників, проаналізованих у цій роботі.

1.3 Аналіз публікацій на тематику кваліфікаційної роботи

Існує довга історія досліджень метричних програм (МП) і велика кількість літератури, що рекомендує фактори успіху для їх впровадження [16], [17]. Однак

літератури про використання МСП у контексті Agile досить мало. Більше того, літератури щодо вимірювання процесів програмного забезпечення та їхньої ролі в покращенні процесів МСП ще менше.

Вимірювання програмних процесів за допомогою процесних метрик дозволяє об'єктивно та кількісно оцінювати програмні процеси, що може призвести до постійного вдосконалення та навчання. Однак вимірювання метрик процесу є проблемою [18]. Процеси програмного забезпечення за своєю суттю є складними та нематеріальними, що робить їх вимірювання складнішим, ніж їх аналоги – матеріальні продукти.

В ідеалі заходи з вимірювання повинні споживати мало зусиль і часу, але бути достатньо адекватними, щоб задовольнити вимоги організації до вимірювань. Організаціям, які розробляють програмне забезпечення, необхідно зважити економічну ефективність, визначаючи пріоритети для цілей і цілей вимірювання. МСП мають додаткові обмеження у вигляді обмеженого бюджету, амбітних термінів виконання та короткострокової стратегії. Через ці причини вимірювання процесів програмного забезпечення, особливо в МСП, стає більш складним завданням.

Автори в роботі [19] провели систематичний огляд використання та впливу програмних метрик на ASD у промисловості. Вони повідомили, що програмні показники в основному використовуються для планування спринту, відстеження прогресу, покращення якості програмного забезпечення, виправлення програмного процесу та мотивації людей. Такі показники, як швидкість, оцінка зусиль, задоволеність клієнтів, кількість дефектів і збірка, використовуються в ASD.

Показники кількості дефектів, затрачених зусиль, кількості ресурсів, тривалості проекту, продуктивності і задоволеності клієнтів зазвичай подаються, як витвір мистецтва.

Інше використання метрик у ASD, яке обговорюється в літературі, стосується планування та відстеження розробки програмного забезпечення [19],

розуміння продуктивності розробки та якості продукту [20], вимірювання якості процесу [21], оцінки зусиль та звітування про прогрес і якість зацікавленим сторонам, не залучені до реальної розробки [22].

У сукупності метрики процесу (МП), спрямовані на планування спринту, процес виправлення програмного забезпечення, оцінку зусиль, продуктивність розробки та дефекти програмного забезпечення, можна використовувати для вимірювання продуктивності процесів організації. Однак ця мета прямо не вказана в жодному зі згаданих вище оглядів. Навпаки, [19] зауважив, що необхідні додаткові дослідження, щоб дослідити обґрунтування використання показників, які автори знайшли у своєму огляді.

Більшість досліджень представляють початкові результати впровадження метрик в організаціях, які не були оцінені в більш широкому промисловому контексті. Використання показників для вимірювання кількості, якості, темпу та статусу роботи може призвести до більш точного та професійного прийняття рішень. Однією з ключових переваг, за словами авторів, були кращі цілі вимірювання, які узгоджувалися зі зрілістю компанії. Що стосується метрик процесів, то більшість досліджень зосереджені на використанні метрик процесу головним чином для прогнозування збоїв/дефектів програмного забезпечення.

Були проведені дослідження, що оцінювали МП у малих і середніх підприємствах, але обсяг був обмежений окремим регіоном, що ускладнює узагальнення їхніх висновків. Наприклад, з метою оцінки членів парламенту в індустрії програмного забезпечення Пакистану [23] провели систематичне картографічне дослідження в поєднанні з опитуванням серед 200 практиків, щоб підкреслити стан практики вимірювання. Сорок два відсотки організацій, які відповіли на опитування, були МСП. Загалом МСП виявилися гіршими, ніж їхні великі колеги.

Наприклад, МСП мають найнижчу частку серед організацій, які мають будь-який визначений процес вимірювання, стандарти вимірювання та використання моделей вимірювання та інструментів. Крім того, 65% малих і

середніх підприємств, як правило, використовують МП в основному на рівні проекту, і лише 13% малих і середніх підприємств впроваджують їх у всій організації.

Одним із позитивних результатів щодо вимірювання процесів програмного забезпечення було те, що 70% малих і середніх підприємств повідомили, що вони або зосереджуються на вимірюванні процесу, або поєднують процес і дві інші сутності. Однак у відповідних первинних дослідженнях був неясний контекст, у якому проводилося вимірювання процесу, і не було зосереджено увагу на показниках процесу для вдосконалення процесу.

Існує досвід іспанського МСП у впровадженні МП. Тепер практики можуть об'єктивно оцінити компроміс між своєчасними релізами та надійністю програмного продукту. Автори [24] співпрацювали з турецьким МСП у галузі охорони здоров'я, щоб інституціоналізувати практику вдосконалення процесів, і повідомили про покращення у розподілі часу організації для вимог, кодування та етапів тестування. Крім того, автори виявили, що рівень дефектів, а також оцінка зусиль тестування зменшилися.

Один із найбільш цікавих підходів до вдосконалення процесу МСП був задокументований в експериментальному дослідженні [25]. Автори представили Scrum, щоб підвищити продуктивність процесу без впливу на якість продукції на дуже малому підприємстві. Автори стверджують, що Scrum може виявитися гарною альтернативою для вдосконалення процесів в організації з дуже обмеженими ресурсами, що протягом тривалого часу викликає занепокоєння при впровадженні МП в МСП. З цих досліджень стає очевидним, що оцінки МП у малих і середніх підприємствах стосуються головним чином загального вдосконалення процесу програмного забезпечення, де роль метрик процесу для досягнення цієї мети або мається на увазі, або взагалі відсутня.

Згідно з сучасним рівнем техніки, існує велика залежність від експертів із вимірювання та досвіду, і організації, як правило, віддають перевагу сприйняттю співробітників об'єктивним процесам вимірювання для вдосконалення процесу.

Навпаки, пточне дослідження надає емпіричні докази використання метрик процесу для покращення продуктивності процесу та навіть полегшення прийняття рішень. Емпірична валідація є особливо відмінним аспектом нашого дослідження, оскільки вона була визначена як прогалина в дослідженні.

Крім того, слід зазначити, що рішення Q-Rapids, яке втілює МП, об'єднує такі основні функції, як автоматичний збір даних, підтримку різноманітних джерел даних, визначення експертних показників і візуалізація, про що не йдеться в літературі, описаній у депутатах.

1.4 Методологія дослідження

Для дослідження були використані дані з відкритих джерел, опубліковані за результатами моніторингу девелоперської компанії ІТТІ. Це компанія з розробки програмного забезпечення та консалтингу у Польщі. Зараз у там працює понад 7000 співробітників. Компанія надає послуги з розробки програмних продуктів в ряді прикладних областей (адміністрування, комунальні послуги, електронна охорона здоров'я та управління кризовими ситуаціями). У цій роботі описується дослідження, зосереджене на одному конкретному програмному продукті CONTRA. Це інтегрована система програмного забезпечення корпоративного класу для управління складом (WMS) і виробництвом (MES), розгорнута у формі веб-додатку.

В проєктах компанії використовується Scrum для розробки програмного забезпечення, включаючи і згаданий продукт CONTRA. Як правило, від 7 до 10 розробників щодня працюють над конкретними розгортаннями або над новими функціями для вдосконалення продукту. Команда Scrum проводить щотижневі зустрічі зі спринту в останній день спринту. Кожна зустріч команди Scrum складається з наступних традиційних частин: огляд, ретроспектива та планування наступного спринту.

Для проведення цього дослідження ми застосували цикл «Дія-дослідження»: діагностика, планування та проектування дій, виконання дій, оцінка та специфікація навчання [31], [38]. Учасники ІТТІ в проекті Q-Rapids виконували в компанії подвійну роль дослідників і лідерів проекту.

Дія розпочалася у вересні 2018 року з діагностики галузевих потреб у формі вдосконалення процесів, які ІТТІ хотів розглянути. Це породило нашу дослідницьку мету та дослідницькі питання (задокументовані в Розділі IV-C).

Будучи учасником проекту Q-Rapids H2020, щоб впоратися з цими вдосконаленими, ІТТІ вирішив налаштувати підхід та інструмент Q-Rapids за допомогою розширених показників процесу. У жовтні 2018 року як команда дослідників, так і команда CONTRA Scrum вирішили, які репозиторії компаній використовуватимуться та як зробити ці дані придатними для дії з результатами проекту Q-Rapids (планування та дизайн дій).

Далі та ж об'єднана команда дослідників і практиків спільно виявила 25 метрик процесу-кандидата, доданих до вибраних даних компанії. Отримані метрики процесу були використані командою CONTRA Scrum під час їхніх зустрічей для аналізу ефективності процесу та якості продукту, а також для оцінки завдань (виконання дій). Ця акція проходила з листопада 2018 року по травень 2019 року.

Показники процесу були оцінені під час ретроспективної сесії з Scrum-командою CONTRA в червні 2019 року (оцінка).

Нарешті, команда вивчила підмножину показників процесу, які є більш ефективними для діагностованих проблем (визначаючи навчання). З липня 2019 року до теперішнього часу така підмножина метрик процесу використовується в інших проектах ІТТІ.

1.1.1 Мета та питання дослідження

На регулярних зустрічах Scrum-команди діагностували необхідність:

- a) відстежувати продуктивність процесу команди;
- b) підтримувати стабільний рівень якості продукту, додаючи нові функції;
- c) покращувати оцінку завдань під час спринтів.

Беручи до уваги ці виробничі потреби, керівництво розглянуло можливість налаштування метрик процесу, концепцію, яка вже реалізована в проекті Q-Rapids, і застосувало її до CONTRA, яка виступає в якості пілотного випадку. Дотримуючись підходу Goal-Question-Metric (GQM) [26], ми можемо означити результуючу дослідницьку мету цього дослідження наступним чином: Аналіз метрик процесу розробки з метою їх оцінки щодо моніторингу та оцінки ефективності процесу Agile з точки зору команди Scrum у контексті МСП.

Ми розбиваємо цю загальну дослідницьку мету на три дослідницькі питання, узгоджені з потребами, передбаченими вище:

- RQ1. Чи допомагають метрики процесу Scrum-команді МСП контролювати ефективність власного процесу?
- RQ2. Чи допомагають метрики процесу команди Scrum малого та середнього бізнесу підтримувати стабільний рівень якості продукту, додаючи нові функції?
- RQ3. Чи допомагають метрики процесу Scrum-команді МСП покращити оцінку завдань під час спринтів?

1.1.2 Інструментальне забезпечення дослідження

На етапі планування дій і проектування циклу дослідження ми вирішили використовувати GitLab як джерело даних для пілотного проекту. GitLab – це веб-інструмент із відкритим кодом, який забезпечує підтримку повного життєвого циклу розробки програмного забезпечення, зосереджуючись на

управлінні репозиторієм і відстеженні проблем серед інших можливостей. GitLab широко використовується в усіх проектах компанії, і, зокрема, Scrum-команда CONTRA підтвердила, що це інструмент, який може найкраще відображати процес, якого дотримується команда під час розробки.

Компанія збирила дані з GitLab приблизно за 12 місяців розробки проекту, щоб показники процесу можна було оцінювати протягом тривалого періоду життєвого циклу. За цей час загалом 31 розробник відкрив до 2975 питань, з яких вони закрили до 2651, і відбулося 40947 подій, що описують зміни статусу завдань або проблем.

Таблиця 1.1 містить зібрані дані для кожної проблеми з GitLab. Ці дані зберігаються в спеціальному індексі в системі Elasticsearch. З цих даних протягом досліджуваного періоду інструмент Q-Rapids надав загалом 1830 метрик, 1098 факторів і 732 точки оцінки стратегічних показників.

Таблиця 1.1 – Інформація про види даних, зібраних з GitLab

Атрибут	Опис	Тип
assignee	призначений розробник	текст
author	учасник команди для створення артефакту	текст
dstart	дата коли артефакт з'явився в беклозі перший раз	дата
dfinish	дата коли артефакт був закритий	дата
elapse	час життя артефакту (в днях)	ціле число
estimate	оціночний час життя артефакту (в днях)	ціле число
created	час створення артефакту	дата
issueid	ідентифікатор артефакту	ціле число
isactive	вказує чи присутній артефакт на дошці спринту	текст
colid	ідентифікатор стовпця на дошці спринта	текст
returned	кількість повернень артефакту в стовпець дошки спринта	ціле число
spent	час, витрачений на закриття артефакту	ціле число

Щоб оцінити корисність метрик процесу, визначених у проекті, команда Scrum діяла наступним чином:

- реалізувалися конектори для GitLab, які дозволили ефективно збирати дані для початку процесу вимірювання;

- враховуючи наявність двох стратегічних індикаторів, наданих проектом Q-Rapids, пов'язаних із двома першими дослідницькими питаннями (продуктивність процесу та якість продукту), використовували інформаційну панель Q-Rapids для метрик процесу, пов'язаних із цими запитаннями.
- для третього досліджуваного питання віддали перевагу розгортанню інформаційних панелей Kibana, щоб оцінити оцінку завдання. Щоб забезпечити комплексне рішення, ці інформаційні панелі Kibana були інтегровані в інформаційну панель Q-Rapids.

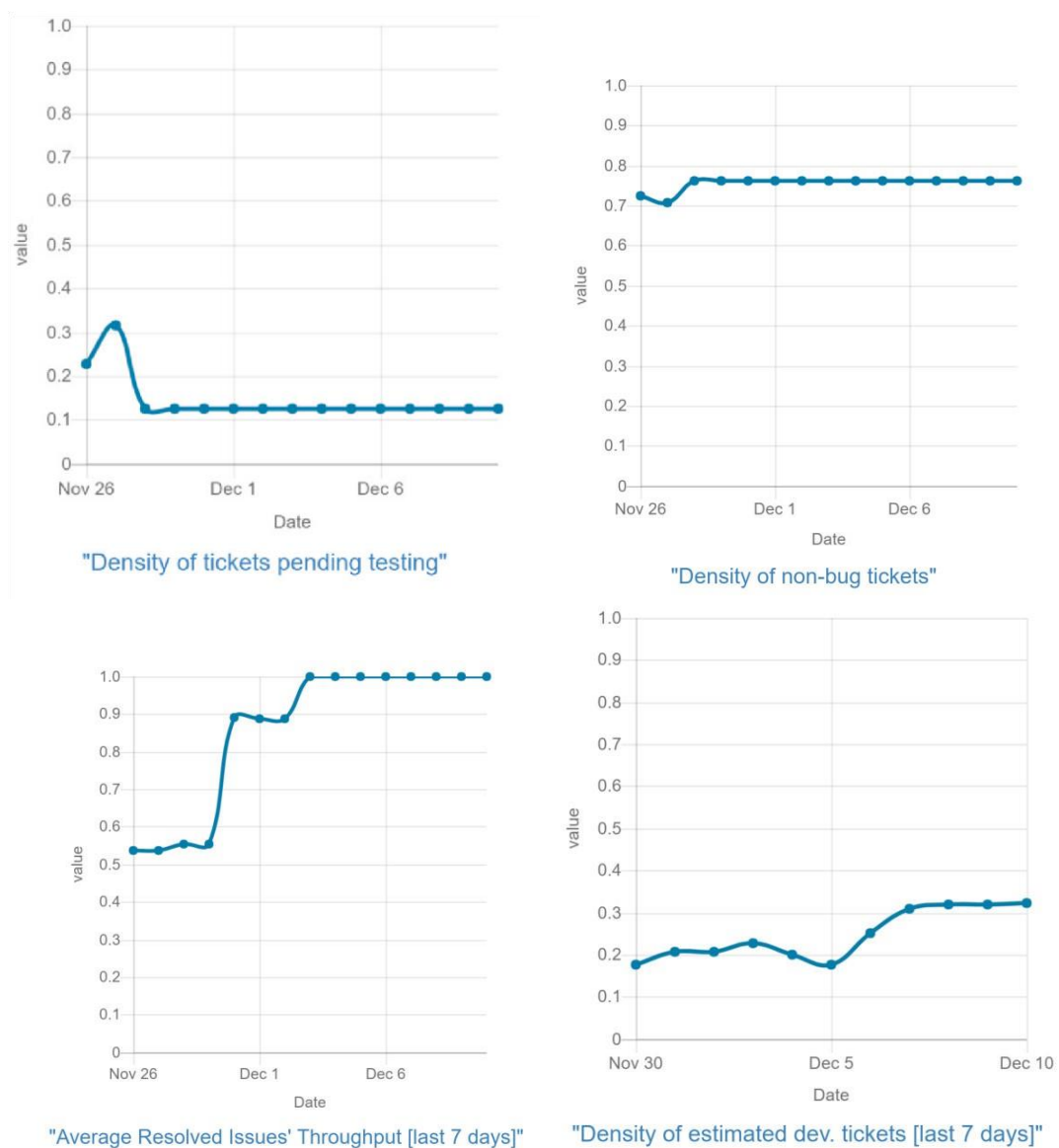


Рисунок 1.2 – Графіки щільності карток артефактів на дошці спринта

На рисунку 1.2 показана інформаційна панель Q-Rapids із прикладом перегляду історичних даних. На цих діаграмах ми бачимо, що щільність карток, що очікують на тестування, і щільність помилок були стабільними у значеннях низької та високої якості відповідно протягом п'ятнадцяти днів. З іншого боку, за той самий період спостерігається явне покращення середньої пропускну здатності вирішених проблем (останні 7 днів) і деяке покращення щільності очікуючих карток розробки (останні 7 днів).

На рисунку 1.3 показано приклад використання панелей Kibana.

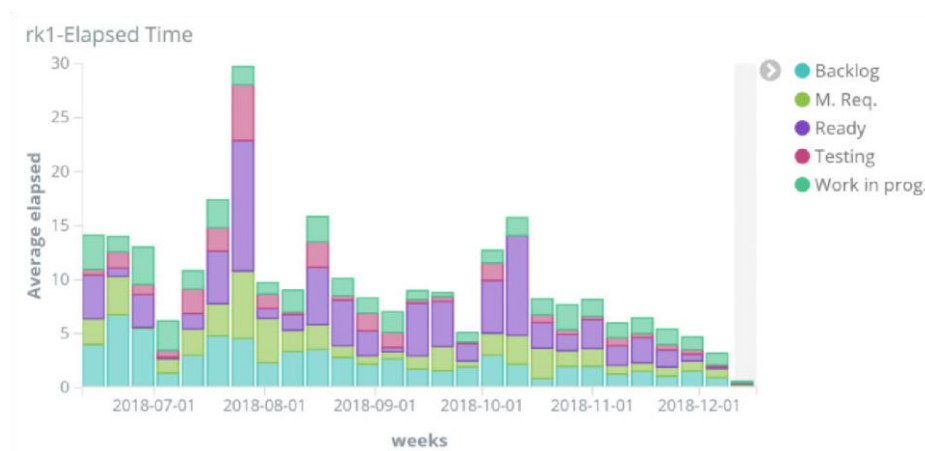


Рисунок 1.3 – Приклад використання панелей Kibana

У цьому випадку представлено подання об'єднує кілька показників, пов'язаних із середнім часом виконання завдань (відповідно до їх стану).

Для окремих етапів процесу розробки ми виділили метрики, пов'язані з виконанням завдань, виправленням помилок і тестуванням.

РОЗДІЛ 2. ФОРМУВАННЯ СИСТЕМИ МЕТРИК ПРОЕКТУ

2.1 Визначення метрик процесу

Команда Scrum проекту та дослідницька група обговорили й проаналізували, який тип показників процесів на основі GitLab вони вважають кандидатами для оцінки процесів Agile-розробки.

Команда розробників надала відповідні спостереження, які стали основою для розробки набору показників:

- Три основні концепції, які вони використовують у своїй щоденній практиці для моніторингу прогресу:
 - завдання;
 - проблема/помилка;
 - трудомісткість.
- Перехід станів між завданнями розробки (відкрито → завершено → закрито) є особливо важливим для аналізу прогресу.
- Трудомісткість особливо цікава щодо її оцінки, оскільки планування ресурсів у команді (наприклад, розподіл розробників) значною мірою залежить від її точності.
- Наведені вище концепції можна проаналізувати в основному з двох точок зору: числового (наприклад, числа, накопиченої суми або середнього) і часового виміру.
- Показники мають бути вимірюваними за допомогою даних GitLab.

Беручи до уваги ці принципи, команда об'єднала пропозиції з 25 потенційних показників. Незважаючи на те, що багато інших показників виявилися цікавими, команда віддала перевагу зберегти пропозицію керованою на цій першій ітерації, таким чином зосередившись на тих показниках, які, як погодилися розробники, є найбільш визначальними. Показники можна розділити на кілька категорій, як показано в таблиці 2.1.

Таблиця 2.1 – Вибрані показники для характеристики Agile-проекту

No	Category	Metric	Value
1	General	Number of development tasks	[number]
2	General	Number of completed (closed) development tasks	[number]
3	General	Number of incomplete (open) development tasks	[%] [number]
4	General	Average number of development tasks	[number]
5	General	Average time of development task in the project board – from the moment it was added, to the moment it was closed	[number/time]
6	General	Average time needed to resolve an issue	[number/time]
7	Task estimation	Effort estimation accuracy for development tasks	[%]
8	Task estimation	Average difference between estimated effort (“estimated” attribute) and real effort (“spend” attribute)	[number/time]
9	Task estimation	Number of development tasks without estimation of effort (“estimated”)	[%] [number]
10	Task estimation	Number of development tasks without real effort (“spend”)	[%] [number]
11	Task estimation	Total sum of estimated effort values (“estimated”)	[number]
12	Task estimation	Sum of effort actually spent (“spend”)	[number]
13	Task implementation	Average task implementation time based on project board	[number/time]
14	Task implementation	Average time-to-implementation of task based on project board	[number/time]
15	Task implementation	Number of tasks with unassigned “Milestone” (sprint)	[%] [number]
16	Task implementation	Number of tasks not yet assigned to any developer	[%] [number]
17	Task implementation	Number of ongoing development tasks not belonging to the current sprint	[%] [number]
18	Bug fixing	Number of development tasks with reported bugs	[%] [number]
19	Bug fixing	Average time of task correction based on project board	[number/time]
20	Bug fixing	Average time-to-correct of task based on project board	[number/time]
21	Bug fixing	Percentage of ‘non-bug’ type tasks with respect to total tasks on the board	[%] [number]
22	Testing	Average time-to-test of development tasks	[number/time]
23	Testing	Average testing time based on project board	[number/time]
24	Testing	Percentage of tasks waiting for testing	[%] [number]
25	Others	Number of merge requests without discussion / comments during the code review	[%] [number]

Було виділено наступні групи показників.

Загальні показники. Відповідно до спостережень команди Scrum, ми пропонуємо вказувати загальну кількість і середню кількість завдань розробки (метрики №1 і №4), кількість завдань на основі їх статусу, наприклад, виконано,

закрито тощо (№2 і № 3) і середній час життя завдань (№5 і №6). Показники №2 і №3 можна розширити, а також проаналізувати кількість завдань, позначених як «виконується», «тестування», «готово» тощо, але ми відкинули це, щоб зберегти простий підхід у цій першій ітерації. Кожну загальну метрику можна розрахувати в різних вимірах, тобто для кожного розробника, для конкретного проекту, для області (інтерфейс/бекенд), для спринту чи випуску, звужуючи часовий проміжок до конкретного діапазону.

Метрики оцінки завдання. Ця категорія включає показники, пов'язані з плануванням розподілу зусиль та аналізом споживання зусиль/ресурсів. Вони можуть вказувати на точність такої оцінки, включаючи середнє відхилення оцінки по відношенню до реальних витрат зусиль (метрики №7 і №8), загальну суму оцінених або використаних ресурсів (№11 і №12), а також повноту завдання оцінка (№9 і №10). Подібно до загальних показників, показники, пов'язані з оцінкою завдання, також можна розрахувати в різних вимірах, наприклад з урахуванням виконавців, проектів, спринтів або діапазону часу.

Для окремих етапів процесу розробки виділено метрики, пов'язані з виконанням завдань, виправленням помилок і тестуванням.

Показники виконання завдання. Ця категорія включає показники на основі часу, що вказують середній час впровадження та середній час очікування впровадження (метрики №13 і №14). Інші три метрики в цій категорії пов'язані зі статусом виконання завдання, а саме призначенням даному спринту (№15 і №17) і певному розробнику (№14).

Показники виправлення дефектів. Вони включають кількість завдань, в яких повідомлено про помилку (показники №18 і №21) та ознаки середнього часу, необхідного/часу очікування для виправлення помилки (19 і 20).

Метрики тестування. Подібним чином показники №22 і №23 вказують середній час тестування та середній час очікування на тестування, тоді як показник №24 показує поточний відсоток завдань, що очікують на перевірку.

Інші показники. Крім того, ми пропонуємо метрику, що показує кількість незакоментованих запитів на злиття (метрика №25), визначених розробниками проекту як релевантні.

2.2 Оцінка метрик процесу

Далі обговоримо вплив використання метрик процесу в пілотному проекті (що відповідає етапу оцінки циклу «Дія-дослідження») на три дослідницькі питання:

- a) моніторинг ефективності процесу;
- b) підтримання стабільної якості продукції рівень;
- c) покращення оцінки завдань під час спринтів.

Процес використання метрик керував Власник проекту CONTRA. Що стосується продуктивності процесів і якості продукції, власник проекту щомісяця аналізував стратегічні показники, відображені інформаційною панеллю Q-Rapids для цих двох концепцій.

2.2.1 Продуктивність процесу розробки

На рисунку 2.1 показано приклад, у якому ми бачимо, що якість продукту залишається на стабільному рівні, тоді як у випадку продуктивності процесу є значні зміни.

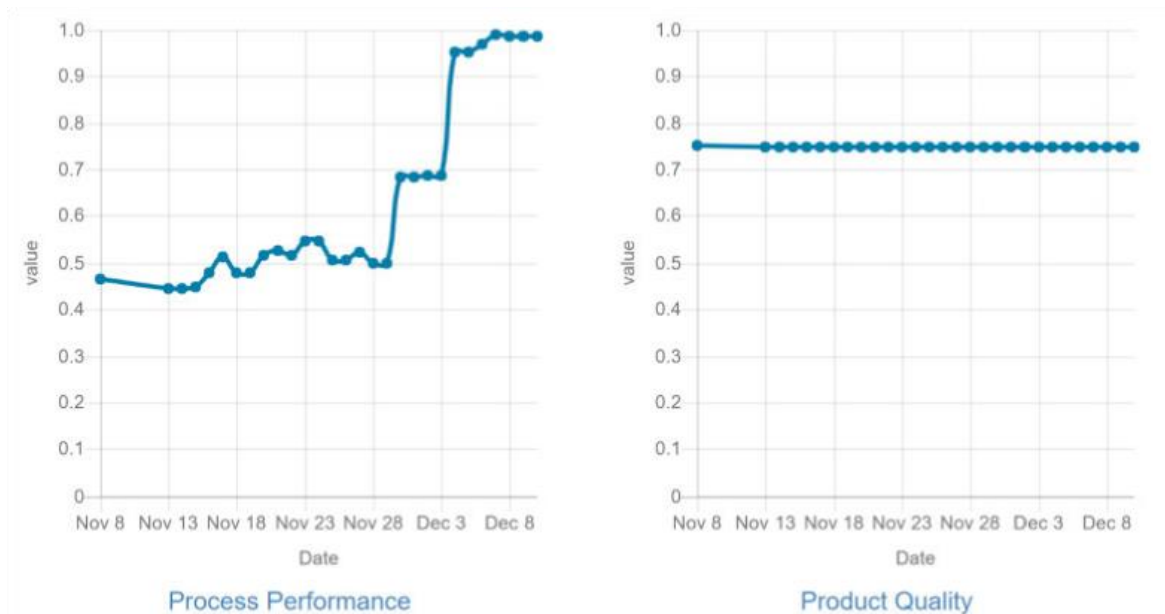


Рисунок 2.1 – Зміна продуктивності процесу та якості продукту

Власник продукту та Scrum Master хотіли знати причину зміни поведінки продуктивності процесу, тому вони використали можливість детального перегляду Q-Rapids, застосовану до стратегічного індикатора ефективності процесу (див. рисунок 2.2).

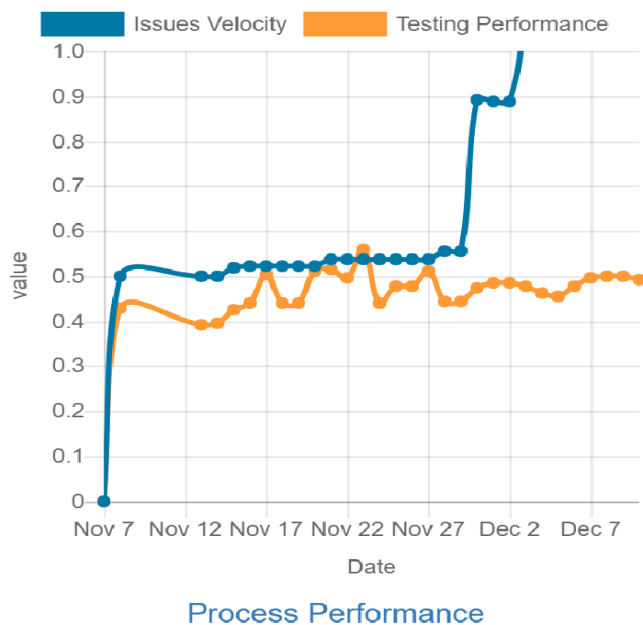


Рисунок 2.2 – Індикатор продуктивності процесу

У результаті вони змогли визначити, що фактор швидкості вирішення дефекту значно покращився протягом місяця, тобто команда розробників збільшує швидкість розробки. Натомість продуктивність процесу лише зазнала незначних коливань.

Далі, щоб більш детально зрозуміти причину підвищення швидкості проблем, вони використали можливість детального перегляду, щоб візуалізувати еволюцію їхніх показників впливу. На рисунку 2.3 показано, що показник середньої пропускної здатності вирішених проблем (за останні 7 днів) покращився протягом місяця.



Рисунок 2.3 – Індикатори ефективності процесів закриття артефактів та тестування

Під час обговорення в команді причини цього вдосконалення стало очевидним, що візуалізація показників через інформаційну панель дозволила власнику продукту покращити своє розуміння кількох аспектів процесу Scrum, які залишалися невідомими до використання Q-Rapids. Іншими словами, власники продуктів поклалися на неофіційні свідчення, а не на дані, зібрані в реальному часі щодо процесу їх розробки.

2.2.2 Якість продукту

Як показано на рисунку 2.1, метрика, показана на інформаційній панелі Q-Rapids, не принесла додаткової цінності процесу розробки в цьому пілотному проєкті. Причина може полягати в тому, що інші інструменти, такі як SonarQube, уже були в компанії, і тому фактори якості коду вже були розглянуті. У будь-якому випадку команда Scrum вважала позитивним те, що незважаючи на додавання нових функцій до продуктів (тобто багато завдань було закрито під час спринту), якість продукту залишалася на стабільному рівні без технічного боргу.

2.3 Оцінка завдань (тасків)

Про відсутність механізмів візуалізації оцінки завдання повідомлялося ще до початку дослідження. Використовуючи метрики процесу, візуалізовані за допомогою Q-Rapids, власник продукту зміг побачити, як частину фактора продуктивності тестування, низькі значення метрики щільності очікуваних заявок розробки (останні 7 днів) (рисунок 2.3(b), зелений лінія). Щоб дізнатися більше, власник продукту перейшов до перегляду показників Kibana та помітив, що за проаналізований проміжок часу було 37 проблем без оцінки (див. рисунок 2.4).

Ця можливість плавного переходу на Kibana з QRapids була високо оцінена командою CONTRA. Наприклад, використовуючи інформаційну панель Kibana, дивлячись на кругову діаграму, власник продукту може визначити ключових осіб, наприклад, розробників із найбільшою кількістю призначених проблем (див. рисунок 2.5). Інформаційна панель Kibana також містить інформацію для аналізу іншої статистики, пов'язаної з даним розробником, наприклад, середній час виправлення помилки або виправлення (див. рисунок 2.6).

filters	Unique count of iid	Last 50 iid
READY	65	659, 677, 717, 712, 659, 714, 724, 720, 699, 706, 630, 712, 659,
estim=0	37	659, 717, 712, 659, 720, 712, 659, 712, 659, 645, 659, 710, 709,
spent=0	10	717, 714, 704, 707, 704, 704, 696, 695, 696, 660, 686, 667, 650

filters	Unique count of iid	Last 50 iid
READY	65	478, 718, 721, 721, 478, 693, 715, 712, 659, 706, 630, 698, 681, 630, 706, 683, 711, 645, 659,
estim=0	37	704, 623, 707, 645, 697, 704, 694, 697, 704, 645, 694, 658, 697, 696, 695, 687, 696, 660, 645,
spent=0	10	

filters	Unique count of iid	Last 50 iid
READY	65	710, 709, 704, 623, 707, 645, 697, 704, 694, 697, 704, 645, 694, 658, 697, 683, 696, 695
estim=0	37	645, 686, 479, 684, 657, 682, 682, 680, 645, 657, 481, 672, 646, 663, 631, 666, 653, 625
spent=0	10	

Рисунок 2.4 – Інформаційна панель Kibana, включно з непрогнозованими проблемами

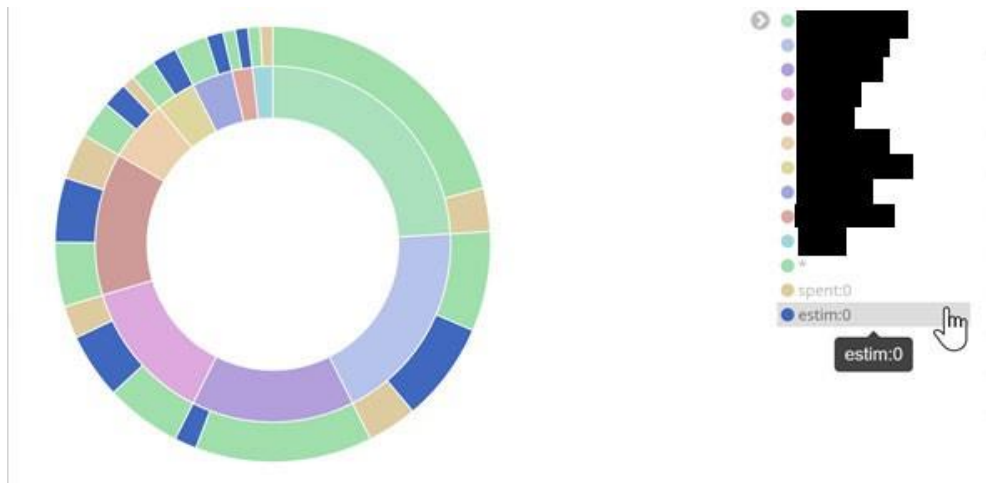


Рисунок 2.5 – Інформаційна панель Kibana, включаючи діаграму проблем, призначених розробникам. Чорні прямокутники використовуються, щоб приховати імена

filters	Unique count of iid	Last 50 iid
READY	9	809, 713, 804, 628, 713, 775, 796, 700, 747, 688, 713
estim=0	7	809, 804, 628, 775, 796, 747, 688
spent=0	3	809, 628, 796

Рисунок 2.6 – Інформаційна панель Kibana зі статистикою розробників

Після вибору прізвища конкретного розробника дані, пов'язані з цим розробником, відфільтровуються. Тепер власник продукту може бачити, що 7 із

9 завдань, виконаних цим розробником, не були оцінені з точки зору часу виконання. Це допомагає виявити ті завдання, які не були оцінені, і вжити заходів, що призводить до покращення оцінки завдань від спринту до спринту.

2.4 Вибір показників

Щоб оцінити корисність метрик для команди Scrum було проведено ретроспективну сесію за участю Scrum Master і одного розробника з проекту, а також трьох дослідників з проекту Q-Rapids.. Ретроспективна сесія була поділена на три основні заходи:

- a) дослідження відповідних метрик процесу, які використовувалися протягом шести місяців у пілотному проекті;
- b) відкрита сесія зворотного зв'язку з обговоренням причин впливу використання метрик процесу;
- c) документування результатів сеансу в шаблоні впливу метрик процесу.

У результаті цього сеансу показники, які вважалися більш цінними (див. таблицю 2.1).

Показник №7: Точність оцінки для завдань розробки (для кожного розробника в проекті за певний проміжок часу).

Показник №9: кількість завдань розробки з відсутністю оцінки зусиль, які потрібно витратити («оцінка») на проект на розробника.

Показник №10: кількість завдань розробки з недостатньою цінністю використаних зусиль («витрати») на проект на розробника.

Показник №11: Загальна сума оцінених значень зусиль ("оцінка") на проект на розробника.

Показник №12: сума використаних зусиль («витрати») на проект на розробника.

Показник №18: кількість завдань розробки, про які повідомлено про помилку.

Показник №19: Середній час виправлення завдання на основі дошки проекту.

Показник №20: середній час виправлення завдання на основі дошки проекту.

Показник №21: Відсоток завдань типу «без помилок» до загальної кількості завдань на дошці.

Ці показники значно покращують керування такими процесами, як оцінка завдань і виправлення помилок, які є ключовими для швидкої розробки якісного та стабільного програмного забезпечення. Крім того, після застосування цих показників управління командою стало більш ефективним і прозорим.

2.5 Загальна оцінка метрик процесу

Загальна оцінка метрик процесу у випадку пілотного проекту довела їх цінність для компанії. Запропоновані, розраховані та візуалізовані метрики процесу (з використанням інформаційної панелі Q-Rapids, переглядів Kibana або навіть спеціальних візуалізацій, розроблених в компанії) були оцінені як корисні, і деякі з метрик тепер використовуються на практиці не тільки в пілотному проекті, але для всієї компанії.

Після того, як вибір зроблено, на кожній ретроспективній зустрічі команди Scrum команда зазвичай витрачає від 15 до 20 хвилин на візуалізацію та аналіз вибраних метрик процесу. Показники процесів чудово підходять, оскільки ця частина зустрічі присвячена людям, процесам, інструментам, отриманим урокам і тому, як покращити спосіб роботи. Звичайно, роль власника продукту та Scrum-майстра полягає в тому, щоб зробити ці обговорення та демонстрації цікавими, але це виявилось легкою роботою для них, оскільки розробники зазвичай люблять статистику та деякі тенденції/графіки, такі як ті, що показані в попередньому розділ. Ці результати, тенденції та значення показників

використовуються для мотивації команди та покращення процесу, а також для пошуку проблем для їх вирішення.

Найважливішою перевагою метрик процесу, яку сприймає компанія, є зосередженість на ефективності процесу та команди. Запропоноване рішення покращило спосіб, у який розробники тепер звітують про час, витрачений на проблеми/квитки, і дозволяє порівнювати із запланованими зусиллями.

Включення інформаційних панелей і метрик процесу в процес розробки програмного забезпечення ІТТІ покращило готовність і ефективність звітування про витрачений час і планування зусиль. Крім того, розрив між запланованими та витраченими зусиллями постійно зменшується, що означає, що власники продуктів, Scrum-майстри, а також розробники проводять оцінку набагато краще.

Із більш практичних переваг і можливих рішень запропоновані показники дозволяють ефективно відстежувати завдання/проблеми в проекті для кожного розробника та спринту (або вибраного періоду часу).

Як було показано в попередньому розділі, тепер можна перевірити ефективність кожного розробника. Ми з'ясували, що оптимальний зареєстрований час/зусилля має бути близьким до 4 днів (це означає, що 1 день витрачається на аспекти, про які не повідомляється (наприклад, коли досвідчені розробники допомагають менш досвідченим), і це добре зрозуміло та виправдано).

Власники продукту та Scrum-майстри здебільшого шукають інформацію про вузькі місця процесу та, загалом, скільки часу «лежить» заявка на кожній фазі процесу. Схема процесу в компанії представлена на рисунку 2.7.

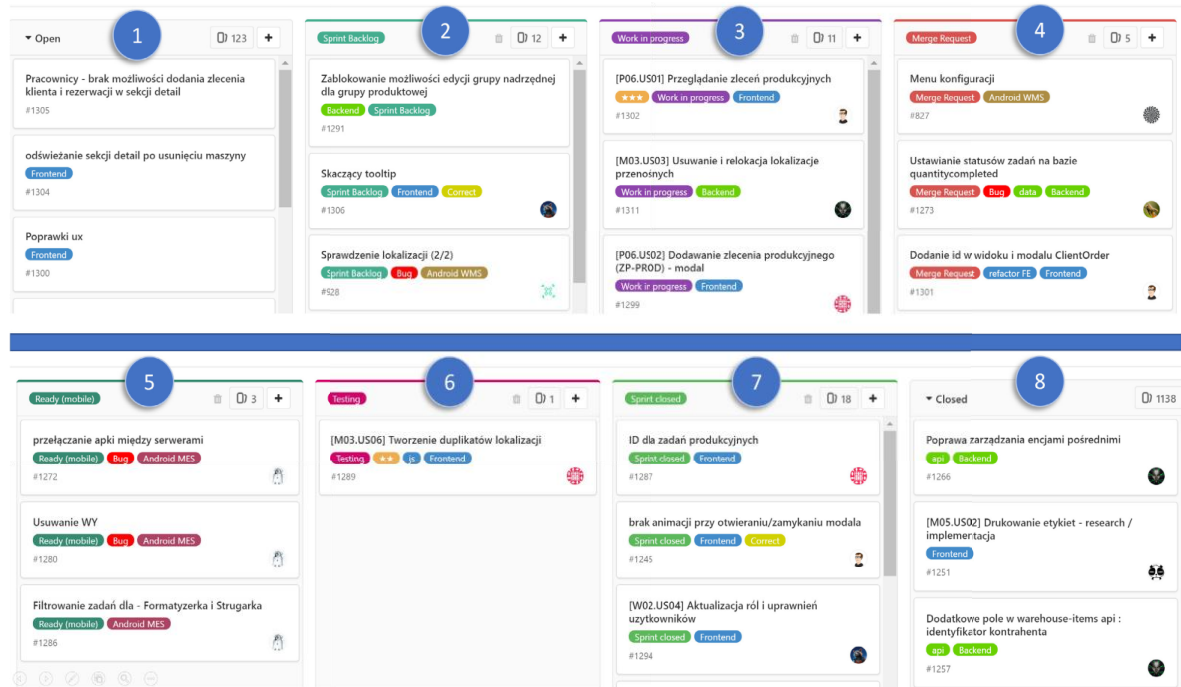


Рисунок 2.7 – Приклад Scrum-панелі моніторингу процесу розробки

Процес складається з 8 кроків, а запропоновані метрики процесу чудово показують статус заявок у проекті, на кожній фазі процесу, на кожного розробника та в заданий проміжок часу.

На пілотному проекті рішення спочатку показало, що вузьке місце знаходиться на етапі тестування. Така ситуація сприяла швидкому вирішенню залучити більше тестувальників, але не вирішила всіх проблем. Переглядаючи метрики процесу, компанія виявила, що вузьке місце змістилося до фази «запиту на злиття». Така ситуація означала, що досвідчені розробники (ті, хто може виконувати перевірку коду та злиття) не мають достатньо часу та ресурсів для виконання завдань на цьому етапі процесу розробки програмного забезпечення. Тепер цю ситуацію вирішено шляхом надання прав на виконання «злиття» розробникам із середнім досвідом, щоб покращити загальний процес.

Важливим аспектом, на який слід звернути увагу, є те, що деякі із запропонованих метрик процесу, хоча й розраховуються для кожного розробника, мають розумно використовуватися власниками продукту та майстрами Scrum, беручи до уваги безліч людських факторів і аспектів. Це

надзвичайно важливо, особливо зараз, коли у світі IT є ринок працівників, а розробників бракує. Використання метрик процесу також відповідає

Як і в будь-якому емпіричному дослідженні, описані методи дослідження та результати можуть мати обмеження. У цьому розділі обговорюються можливі загрози валідності з точки зору побудови, висновку, внутрішньої та зовнішньої валідності та наголошується на застосованих заходах пом'якшення.

Ретроспективна сесія дозволила продовжити розробку практичної значущості метрик процесу з двома членами команди Scrum. Була поставлена мета створити безпечне середовище, заохочуючи учасників висвітлювати будь-які негативні аспекти та вносити пропозиції щодо покращення показників процесу. Нарешті, деякі з результатів можуть бути спричинені не оптимальною реалізацією метрик процесу. Тим не менш, ці результати корисні для інших, щоб дізнатися, як побудувати таку інфраструктуру в реалістичних умовах.

Щоб забезпечити надійність цієї оцінки, план вимірювання та процедура були детально задокументовані. Крім того, результати були перевірені командою Scrum. Таким чином зменшено такі ризики, як пошук результатів під час аналізу, що призвело б до суб'єктивних висновків.

Було оцінено інтегроване рішення Q-Rapids, намалювавши зручний приклад для Scrum-майстра та розробника. Одним з обмежень роботи є те, що ми не змогли отримати випадкову вибірку учасників пілотного проекту. Крім того, ми заздалегідь визначили протокол оцінки, який включав конкретний опис запланованої нами процедури та порядок використання матеріалів, тобто пояснення з усіма кроками, які потрібно було виконати. Після того, як усі партнери погодили остаточний варіант інструкцій з оцінки, ми провели відповідну оцінку. Це має пом'якшити той факт, що нам потрібно було розділити роботу з проведення оцінювання між різними дослідниками та партнерами. Деякі з п'яти дослідників, які проводили оцінку, брали участь у розробці компонентів інструменту Q-Rapids. Щоб мінімізувати цю упередженість, ми переконалися, що в кожному випадку були присутні принаймні два дослідники; один у ролі

модератора/експериментатора, а інший у ролі спостерігача, щоб підкреслити, що учасники можуть говорити вільно.

Метою було краще зрозуміти сприйняття практиків. Ми охарактеризували навколишнє середовище якомога реалістичніше та вивчили придатність нашого відбору параметрів (метрик). GitLab є одним із найбільш широко використовуваних інструментів керування програмним забезпеченням у компаніях, що займаються розробкою програмного забезпечення малого та середнього бізнесу. Таким чином, ми можемо очікувати, що цей аналіз показників може надати компанії-розробнику програмного забезпечення корисну інформацію для покращення якості своїх процесів.

РОЗДІЛ 3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

3.1 Аналіз небезпеки і шкідливості при розробці програмного забезпечення

Організація робочого місця розробника ПЗ впливає на його працездатність.

У своїй діяльності розробник використовує комп'ютер, пристрої збереження інформації, а тому є необхідність забезпечення зручного доступу до всіх технічних засобів. Тому в даному розділі докладніше розглянемо відомості про систему ергономічних норм і принципів організації робочого місця, на котрому проводяться роботи зі створення модуля збору статистики.

Під робочим місцем розуміється зона, оснащена необхідними технічними засобами, у якій відбувається трудова діяльність виконавця або групи виконавців, які спільно виконують одну роботу або операцію.

Організація робочого місця полягає у виконанні заходів, які забезпечують безпечний і раціональний трудовий процес і ефективне використання знарядь та предметів праці, що підвищує продуктивність праці і знижує стомлюваність працівника.

Організація робочого місця залежить від характеру розв'язуваних задач і особливостей предметно-просторового оточення, що визначають робоче положення тіла і можливість пауз для відпочинку, типи і способи засобів відображення і керування, необхідність у засобах захисту, спецодягу, простору для налагодження і ремонту устаткування.

Одним з компонентів діяльності на робочому місці є робочі рухи. Їхня раціональна організація створює умови для зниження стомлення, резерви для підвищеної працездатності. Просторові характеристики руху оператора визначаються траєкторіями руху і розмірами моторного поля (зони досяжності).

При організації робочого місця необхідно забезпечити нормальні умови огляду. Зону огляду описує кут, вершина якого знаходиться в центрі ока, а

сторони складають границі, в яких людина при фіксованому положенні голови й ока добре розрізняє їхнє місцезнаходження.

У горизонтальній площині цей кут складає 300 – 400. При організації робочого місця кут огляду можна взяти 500 – 600, включаючи зону менш ясного огляду. Допустимий кут огляду по горизонталі 900. У вертикальній площині оптимальний кут огляду 100 вверх і 300 вниз від лінії погляду, а допустимий 300 вгору і 400 вниз від лінії погляду.

Щоб зберегти нормальну гостроту зору, робочу поверхню розташовують від очей на відстані від 0,3 м до 0,75 м. Робочі меблі повинні бути зручними для виконання робочих операцій. В даному випадку робочий стіл є основним устаткуванням. Особливо важливе значення має висота столу, його конструкція, яка повинна передбачати шухляди для розміщення інструментів, документації.

Важливе значення має конструкція робочих крісел. Погано підібрані крісла можуть бути причиною надмірної стомлюваності.

Нахил і висота крісла повинні регулюватися відповідно до висоти робочої поверхні і росту працюючого. Рекомендована ширина крісла 370 – 400 мм, глибина 370 – 420 мм, висота спинки 370 – 1000 мм від рівня крісла. Для розміщення ніг необхідно передбачити вільний простір під робочою площиною [35].

Праця людини, що протікає в умовах надмірного нервово-емоційного напруження, довготривалих статичних навантажень, обмеженої рухової активності призводить до неврозів, відхилень у психіці, захворювань опорно-рухового апарату, серцево-судинної системи тощо. Комп'ютери, телебачення, системи зв'язку та інші засоби, що використовують досягнення радіоелектроніки, є генераторами цілої низки електромагнітних випромінювань, вплив яких на організм людини ще не зовсім вивчений.

З широким впровадженням автоматизації та комп'ютеризації виникла потреба врахування психологічних можливостей людини, таких як швидкість реакції, особливості пам'яті та уваги, емоційний стан та ін. Поява операторської

діяльності призвела до суттєвих змін у фаховій структурі праці. Зменшилися фізична важкість праці, ризик виробничого травматизму, однак разом з тим, на працюючу людину посилюється вплив нових, раніше не відомих чи мало вивчених несприятливих виробничих факторів фізичного, хімічного і особливо психофізіологічного характеру.

Проте, розвиток сучасної обчислювальної техніки відбувається не лише у бік покращення її технічних параметрів, але також звертається увага безпеку використання цієї техніки людиною шляхом зменшення потужності випромінювачів, зменшенням рівня випромінювання з моніторів, зменшення напруг живлення, покращення ергономічних характеристик.

Таким чином, в розділі з охорони праці виконано огляд питань безпечної роботи при створенні сайту та встановлено, що умови такої роботи відповідають вимогам з охорони праці, які застосовуються в галузі інформаційних технологій.

3.2 Інформаційно-психологічні небезпеки

Сучасні реалії постіндустріального суспільства, зумовлені значним ростом інформації, відкривають ще одну сферу життєдіяльності людини – інформаційну. Сучасні засоби комунікації і обробки інформації створили принципово нові умови існування людини, що зумовило появу грандіозного проекту об'єднання національних інформаційних і телекомунікаційних структур в глобальну інформаційну інфраструктуру.

Життєдіяльність людини реалізується одночасно зі світом природи і у специфічному для людського суспільства інформаційному середовищі, що має свої закономірності розвитку і функціонування. Інформаційна сфера стає такою ж важливою складовою суспільного життя, як економічна, виробнича, побутова, політична, військова та ін. Нові інформаційні технології, засоби масової комунікації багатократно підсилили можливості впливу на свідомість і підсвідомість як окремої людини, так і на великі групи людей та населення країни загалом.

Інформаційна сфера – сукупність таких елементів:

- об'єкти інформаційної взаємодії чи впливу;
- особисто інформація, призначена для використання суб'єктами інформаційної сфери;
- інформаційна інфраструктура, що забезпечує можливість здійснення обміну інформацією між суб'єктами;
- суспільні відносини, що складаються у зв'язку з формуванням, переданням, розповсюдженням і збереженням інформації.

Особистість, активний соціальний суб'єкт, його психіка піддаються безпосередньому впливу інформаційних чинників (передумов, що чинять опір чи утруднюють формування і функціонування адекватної інформаційно-орієнтуючої основи суспільної поведінки людини (життєдіяльності у суспільстві)), які трансформуються, через його поведінку, діяльність (бездіяльність), здійснюють деструктивний, дисфункційний вплив на його життєдіяльність.

До основних загроз інформаційно-психологічної безпеки відносять можливість настання негативних наслідків для суб'єктів, що піддаються інформаційно-психологічному впливу, які виражаються в таких формах:

- нанесення шкоди здоров'ю людини;
- блокування на неусвідомленому рівні волі, волевиявлення людини, штучне привиття їй синдрому залежності;
- втрата здатності до політичної, культурної, моральної самоідентифікації людини;
- маніпуляція суспільною свідомістю;
- руйнування єдиного інформаційного і духовного простору України, традиційних устроїв суспільства і суспільної моральності, а також порушення інших життєво важливих інтересів особистості, суспільства, держави.

Наприклад, культ жорстокості, насильства, порнографії, розбещеності тощо, які пропагують у засобах масової інформації, друкованих виданнях,

комп'ютерних іграх, мережі Інтернет веде до неусвідомленого бажання у підлітків і молоді, а також дорослих з нестійкою психікою, копіювати запропоновані моделі поведінки. Цей вид пропаганди знижує рівень порогових обмежень і правових заборон, що поряд з іншими умовами відкриває шлях для багатьох правопорушень. Це своєю чергою наносить непоправну шкоду не тільки окремій особистості, але й суттєві збитки національним інтересам країни.

Отже, джерелом інформаційно-психологічної небезпеки є та частина інформаційного середовища, яка через визначені причини неадекватно відображає реалії, вводить в оману людину, засліплює її ілюзією.

Інформаційно-психологічні загрози зумовлені розробкою, виготовленням, розповсюдженням та використанням суб'єктами негативних інформаційно-психологічних впливів, спеціальних засобів і методів такого впливу.

Концепція інформаційно-психологічної безпеки.

Сучасне розуміння безпеки в контексті врахування відношення інтересів особистості, суспільства і держави висуває завдання розгляду нового аспекту цієї проблеми – безпеки в інформаційній сфері життєдіяльності людини, тобто інформаційно-психологічної безпеки.

В інформаційному середовищі, що є складовим системним утворенням, виділяється процесуальна складова як найбільш динамічна і змінна її частина – інформаційно-комунікативні процеси, які активно впливають на індивідуальну, групову і суспільну психологію (індивідуальну, групову, масову свідомість). Маніпулюючи станом інформаційного середовища, змінюється стан духовної сфери суспільства, деформація і деструктивні зміни якої у формі психоемоційної і соціальної напруженості, спотворених норм і неадекватних соціальних стереотипів і установок, оманливих і неприродних орієнтацій та цінностей. Це своєю чергою впливає на стан і процеси у всіх основних сферах суспільного життя, в тому числі політичній і економічній.

Вперше у пострадянському просторі про проблему інформаційно-психологічної безпеки було зазначено в листопаді 1995 р. на науково-практичній

конференції, організованій Інститутом психології Російської академії наук. На цій та подальших конференціях було розкрито роль знання технологій інформаційно-психологічного впливу, метою якого є маніпуляція, для вироблення напрямів реформування психологічного захисту особистості і особистої інформаційно-психологічної безпеки.

Інформаційно-психологічну безпеку особистості визначають такими основними причинами.

Зростання тиску інформаційного середовища визначає необхідність формування нових механізмів та засобів виживання людини як особистості й активного соціального суб'єкта у сучасному суспільстві.

Взаємодія психіки людини з інформаційним середовищем відрізняється якісною специфікою і не має аналогів у комунікації інших біологічних, технічних, соціальних і соціотехнічних структур.

Основною і центральною "мішенню" інформаційного впливу є людина, її психіка.

Отже, інформаційно-психологічну безпеку можливо розглядати як стан захищеності особистості, різних соціальних груп і об'єднань людей від дій, впливів, які здатні проти їхньої волі і бажання змінити психічні стани та психологічні характеристики людини, модифікувати її поведінку і обмежувати свободу вибору, зумовило потребу переосмислення інформаційної взаємодії, а також деяких інших соціально-психологічних процесів і явищ у сучасному суспільстві.

Інформаційно-психологічна безпека – стан захищеності окремих осіб чи груп осіб від негативних інформаційно-психологічних впливів і пов'язаних з цим інших життєво важливих інтересів особистості, суспільства, держави в інформаційному середовищі.

Негативний інформаційно-психологічний вплив – процес зміни психічних станів і характеристик людей під впливом інформаційно-комунікативних процесів як динамічного компонента інформаційного середовища. Цей вплив

спрямований на людину чи групу осіб (у тому числі без їхньої згоди) з метою примусу до визначеної поведінки, оцінки ситуації, керування та корекції індивідуальної та колективної свідомості. Він здійснюється з використанням спеціальних засобів і методів впливу на психіку людини, унаслідок чого він приводить до негативних наслідків для особистості, суспільства і держави.

Спеціальні засоби впливу – технічні і програмні засоби, що використовують для використання з метою негативного інформаційно-психологічного впливу на людину чи групу людей.

Спеціальні методи впливу – послідовність прийомів впливу на психіку людини, використання яких приводить до негативних наслідків для особистості, суспільства та держави.

Головним об'єктом забезпечення інформаційно-психологічної безпеки в інформаційному середовищі у сфері індивідуальної безпеки є усвідомлення інформації, здатність людини адекватно сприймати навколишню дійсність, своє місце в зовнішньому світі, формувати відповідно до свого життєвого досвіду визначені переконання і приймати стосовно них рішення.

Інформаційно-психологічна безпека має спиратися на стандарти інформаційно-психологічної безпеки – затверджені у визначеному порядку інформаційно-психологічного впливу, який не викликає негативних наслідків для психіки людини.

ВИСНОВОК

У цій роботі представлено підхід до визначення та використання метрик процесу, пов'язаних із гнучкою розробкою програмного забезпечення. Це було реалізовано за допомогою: формулювання набору метрик процесу, їх оцінки в реальному проекті та опису практичного та емпіричного використання в конкретній компанії МСП.

Точніше, з точки зору питань дослідження:

- RQ1: Основним внеском роботи є дуже необхідне рішення для моніторингу ефективності кожного етапу процесу розробки програмного забезпечення. Рішення включає підмножину показників ефективного процесу.
- RQ2: Дійсно, головною перевагою для компанії є позитивний вплив на стабільність системи з додаванням нових функцій. По суті, запропонований програмний продукт (CONTRA) потребує налаштування під кожного клієнта (при тому, що сфери діяльності клієнтів значно відрізняються). Запропоновані показники процесу постійно використовуються для забезпечення якості та стабільності програмного забезпечення.
- RQ3: Цінністю запропонованого рішення є механізми для візуалізації оцінки завдання, наприклад, відстеження завдань (квитків, випусків) у реальному часі (в режимі реального часу). Використовуючи ці візуалізації метрик процесу, власник продукту в команді Scrum зміг покращити оцінки завдань.

Незважаючи на те, що наші висновки базуються на конкретній компанії та продукті малого та середнього бізнесу, ми вважаємо, що представлені висновки щодо показників процесу та використання Q-Rapids можуть бути застосовані в ширшому контексті. «Якщо сили всередині організації, які спонукали до спостережуваної поведінки, ймовірно, існують в інших організаціях, цілком ймовірно, що ці інші організації також демонструватимуть подібну поведінку» [27].

Насправді більшість компаній, що займаються розробкою програмного забезпечення малого та середнього бізнесу, використовують процеси, подібні до SCRUM, як показано на рис. 2.7, і їх цікавлять практичні показники, пов'язані з процесами та ефективністю команди. Незважаючи на те, що використовувані інструменти, мітки чи назви етапів процесу можуть відрізнятися, рішення є загальним, хоча, звичайно, потребує певного налаштування. Варто зазначити, що в цій роботі показано реальні переваги, засновані на реальній реалізації для GitLab; однак інші інструменти, такі як Jira, також можна використовувати як джерело даних, і фактично, з'єднувачі для Jira вже реалізовані та доступні як результат проекту Q-Rapids.

Дійсно, запропоновані показники та відповідні рішення задовольняють поточну потребу в інструментах, пов'язаних із процесами розробки програмного забезпечення Agile. Більшість інструментів зосереджено на якості програмного забезпечення або безперервній інтеграції без заходів для процесу. По суті, існує лише одне конкуруюче рішення, яке можна використовувати для аналізу процесу, а саме GitLab Time Tracker. Однак, як показано в документі, тут пропонується ширший набір обчислених показників процесу, кращу візуалізацію, а також значно розширені можливості аналізу.

ПЕРЕЛІК ПОСИЛАНЬ

1. P. Rodríguez, J. Markkula, M. Oivo, and K. Turula, “Survey on agile and lean usage in finnish software industry,” in Proc. ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM), Lund, Sweden, 2012, pp. 139–148.
2. K. Dikert, M. Paasivaara, and C. Lassenius, “Challenges and success factors for large-scale agile transformations: A systematic literature review,” *J. Syst. Softw.*, vol. 119, pp. 87–108, Sep. 2016.
3. Волович, В., Береженко, Б. М., & Боднарчук, І. О. (2022). Задача проєктування програмної архітектури в процесах забезпечення якості. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 104-106.
4. Гузеляк, О., Шевчук, Ю., Береженко, Б. М., & Боднарчук, І. О. (2022). Програмна архітектура в розподілених командах гнучких проєктів. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 110-112.
5. Боднарчук, І., Харченко, О., Хоміцький, Б., & Шимчук, Г. (2019). Проєктування архітектури програмних систем в проєктах з гнучкими методами управління. Матеріали XXI наукової конференції Тернопільського національного технічного університету імені Івана Пулюя, 46-48.
6. Kharchenko, A., Raichev, I., Bodnarchuk, I., & Matsiuk, O. (2021, October). The Survey of Global Software Design Processes. In 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 291-294). IEEE.
7. Strutynska, I., Kozbur, H., Dmytrotsa, L., Bodnarchuk, I., & Hlado, O. (2019, October). Small and medium business structures clustering method based on their digital maturity. In 2019 IEEE International Scientific-Practical Conference

- Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 278-282). IEEE.
8. T. Dybà, “Factors of software process improvement success in small and large organizations: An empirical study in the scandinavian context,” in Proc. 9th Eur. Softw. Eng. Conf., pp. 148–157, 2003.
 9. A. Fuggetta, “Software Process: A Roadmap,” in Proc. Conf. Future Softw. Eng., 2000, pp. 25–34.
 10. E. Kupiainen, M. V. Mäntylä, and J. Itkonen, “Using metrics in agile and lean software development—A systematic literature review of industrial studies,” *Inf. Softw. Technol.*, vol. 62, pp. 143–163, Jun. 2015.
 11. [41] J. Soini, “A survey of metrics use in finnish software companies,” in Proc. Int. Symp. Empirical Softw. Eng. Meas., Sep. 2011, pp. 49–57.
 12. [9] M. Díaz-Ley, F. García, and M. Piattini, “Implementing a software measurement program in small and medium enterprises: A suitable framework,” *IET Softw.*, vol. 2, no. 5, pp. 417–436, 2008.
 13. A. M. Bhatti, H. M. Abdullah, and C. Gencel, “A model for selecting an optimum set of measures in software organizations,” in Proc. Eur. Conf. Softw. Process Improvement., 2009, pp. 44–56.
 14. M. Choraś, R. Kozik, D. Puchalski, and R. Renk, “Increasing product owners’ cognition and decision-making capabilities by data analysis approach,” *Cognition, Technol. Work*, vol. 21, no. 2, pp. 191–200, May 2019.
 15. C. R. Prause, A. Hönle, “Emperor’s new clothes: Transparency through metrication in customer-supplier relationships,” in *Product-Focused Softw. Process Improvement (Lecture Notes in Computer Science)*, vol. 11271, M. Kuhrmann, Ed. Cham, Switzerland: Springer, 2018.
 16. M. G. Mendonca and V. R. Basili, “Validation of an approach for improving existing measurement frameworks,” *IEEE Trans. Softw. Eng.*, vol. 26, no. 6, pp. 484–499, Jun. 2000.

17. P. Ram, P. Rodriguez, M. Oivo, and S. Martinez-Fernandez, "Success factors for effective process metrics operationalization in agile software development: A multiple case study," in Proc. IEEE/ACM Int. Conf. Softw. Syst. Processes (ICSSP), May 2019, pp. 14–23.
18. M. Söylemez and A. Tarhan, "Challenges of software process and product quality improvement: Catalyzing defect root-cause investigation by process enactment data analysis," *Softw. Qual. J.*, vol. 26, no. 2, pp. 779–807, Jun. 2018.
19. E. Kupiainen, M. V. Mäntylä, and J. Itkonen, "Using metrics in agile and lean software development—A systematic literature review of industrial studies," *Inf. Softw. Technol.*, vol. 62, pp. 143–163, Jun. 2015.
20. A. Tarhan and S. G. Yilmaz, "Systematic analyses and comparison of development performance and product quality of incremental process and agile process," *Inf. Softw. Technol.*, vol. 56, no. 5, pp. 477–494, May 2014.
21. M. Unterkalmsteiner, T. Gorschek, A. K. M. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt, "Evaluation and measurement of software process Improvement—A systematic literature review," *IEEE Trans. Softw. Eng.*, vol. 38, no. 2, pp. 398–424, Apr. 2012.
22. M. P. Boerman, Z. Lubsen, D. A. Tamburri, and J. Visser, "Measuring and monitoring agile development status," in Proc. IEEE/ACM 6th Int. Workshop Emerg. Trends Softw. Metrics, May 2015, pp. 54–62.
23. T. Tahir, G. Rasool, W. Mehmood, and C. Gencel, "An evaluation of software measurement processes in pakistani software industry," *IEEE Access*, vol. 6, pp. 57868–57896, 2018.
24. A. Tosun, A. Bener, and B. Turhan, "Implementation of a software quality improvement project in an SME: A before and after comparison," in Proc. 35th Euromicro Conf. Softw. Eng. Adv. Appl., 2009, pp. 203–209.
25. E. Caballero, J. A. Calvo-Manzano, and T. S. Feliu, "Introducing Scrum in a very small enterprise: A productivity and quality analysis," *Commun. Comput. Inf. Sci.*, vol. 172, no. May 2014, pp. 215–224, 2011.

26. V. Basili, G. Caldiera, and H. Rombach, "The goal question metric approach," *Encyclopedia Softw. Eng.*, vol. 1, pp. 528–532, 1994.
27. P. B. Seddon and R. Scheepers, "Towards the improved treatment of generalization of knowledge claims in IS research: Drawing general conclusions from samples," *Eur. J. Inf. Syst.*, vol. 21, no. 1, pp. 6–21, Jan. 2012.
28. Жидецький, В. Ц., Джигирей, В. С., & Мельников, О. В. (2000). Основи охорони праці. Львів: Афіша, 350, 132-136.
29. Навакатіян О.О., Кальниш В.В., Стрюков С.М. Охорона праці користувачів комп'ютерних відеодисплейних терміналів. - К.:1997. - 400с.