

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Інформаційних управляючих систем _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

_____ другий (магістерський) _____
(рівень вищої освіти)

«Дослідження методів контролю виконаних у ІТ-проектах робіт з
використанням технології Process Mining»
(тема)

Виконав: студент _2_ курсу,
групи УПГІТМ-20-1 Чистілін А. В. _____
(прізвище, ініціали)

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

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

Освітня програма Управління проектами в
галузі інформаційних технологій _____
(повна назва освітньої програми)

Керівник д.т.н., проф. Чалий С. Ф. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ІУС

(підпис)

Петров К.Е. _____
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Інформаційних управляючих систем _____

Рівень вищої освіти _____ другий (магістерський) _____

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

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

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Чистіліну Арсенію Володимировичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів контролю виконаних у ІТ-проектах робіт з використанням технології Process Mining».
затверджена наказом по університету від 05.11.2021 р. № 1646 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2021 р.
3. Вихідні дані до роботи науково-технічні публікації, джерела інтернету, що стосуються теми кваліфікаційної роботи.
4. Перелік питань, що потрібно опрацювати в роботі: Вступ; Дослідження гнучких технологій розробки ІТ проекту; Дослідження методів контролю виконання робіт ІТ проекту; Аналіз методів та технології Process Mining; Розробка удосконаленого методу контролю виконання робіт ІТ проекту на основі структуризації логу, що відображає послідовність виконання робіт із часовими мітками; Експериментальна перевірка удосконаленого методу контролю виконання робіт ІТ проекту; Висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз матеріалів з теми роботи	08.11.2021	
2	Постановка задачі кваліфікаційної роботи	09.11.2021-11.11.2021	
3	Обробка матеріалу	12.11.2021-13.11.2021	
4	Попереднє дослідження галузі завдання	14.11.2021-15.11.2021	
5	Аналіз існуючих методів планування та обліку робіт	15.11.2021-16.11.2021	
6	Дослідження розробленого методу обліку виконання робіт у ІТ-проектах	17.11.2021-18.11.2021	
7	Апробація результатів дослідження на прикладі	19.11.2021-20.11.2021	
8	Написання пояснювальної записки	21.11.2021-28.11.2021	
9	Підготовка презентації	29.11.2021-02.12.2021	
10	Перевірка на плагіат	03.12.2021-06.12.2021	
11	Нормоконтроль	07.12.2021-10.12.2021	
12	Захист	14.12.2021	

Дата видачі завдання 08.11.2021 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф Чалий С. Ф.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 72 с., 23 рисунків, 16 джерел.

БАЗА ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА, ОБЛІК ВИКОНАННЯ РОБІТ, ПЛАНУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІТ КОМПАНІЯ

Метою кваліфікаційної роботи є дослідження методів контролю виконаних у ІТ-проектах робіт з використанням технології інтелектуального аналізу процесів.

Об'єктом дослідження є процеси обліку виконаних робіт у ІТ проектах.

Предметом дослідження, є моделі та методи обліку виконаних робіт у ІТ проектах.

Теоретичними результатами дослідження є описи удосконаленого методу метод контролю виконання робіт у ІТ проектах.

Практичними результатами, є створення модулю, який інтегрується у додаток Fluxicon Disco інтелектуального аналізу процесів та дає можливість будувати граф виконання робіт на основі логів з GitHub.

Новизна наукової роботи полягає удосконалення методу обліку виконаних робіт у ІТ-проектах за допомогою технології Process Mining, який дає можливість на основі аналізу логу процесу розробки побудувати граф послідовності робіт з урахуванням міток часу виконання цих робіт

ABSTRACT

The explanatory note to the qualification work contains: 72 pages, 23 figures, 16 sources.

DATABASE, INFORMATION SYSTEM, ACCOUNTING OF WORK PERFORMANCE, PLANNING, SOFTWARE, IT COMPANY

The purpose of the certification work is to study the methods of control of work performed in IT projects using the technology of intellectual analysis of processes.

The object of research is the processes of accounting for work performed in IT projects.

The subject of research are models and methods of accounting for work performed in IT projects.

The theoretical results of the study are descriptions of the improved method of monitoring the performance of work in IT projects.

The practical results are the creation of a module that integrates with the Fluxicon Disco application of intellectual analysis of processes and allows you to build a work schedule based on logs from GitHub.

The novelty of scientific work is to improve the method of accounting for work performed in IT-projects using Process Mining technology, which allows based on the analysis of the log of the development process to build a graph of the sequence of work taking into account timestamps

ЗМІСТ

Зміст	6
Скорочення, та умовні позначки	7
Вступ.....	8
1 Дослідження підходів до моделювання процесів розробки	9
1.1 Структуризація процесів розробки.....	9
1.2 Дослідження моделей процесу розробки	12
1.3 Аналіз методів та технології Process Mining.....	16
1.4 Постановка задачі дослідження.....	25
2 Дослідження моделей процесів розробки.....	27
2.1 Дослідження методів контролю виконання робіт ІТ проекту.....	27
2.2 Методологія Scrum.....	32
2.2 Методологія Agile	34
2.3 Удосконалений метод контролю виконання робіт	37
3 дослідження отриманих наукових результатів	40
3.1 Розробка удосконаленого методу контролю виконання робіт.....	40
4 Використання результатів дослідження на практиці	47
4.1 Структура журналів реєстрації подій.....	47
4.2 Опис програмних засобів для побудови моделі на основі аналізу логів	52
4.3 Експериментальна перевірка пропонованих підходів	54
Висновки	58
Перелік джерел посилання	59
Додаток А. Графічний Матеріал.....	61

СКОРОЧЕННЯ, ТА УМОВНІ ПОЗНАКИ

AIS – автоматизована інформаційна система;

APM – автоматизоване робоче місце;

БД – база даних;

ІС – інформаційна система;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

ПР – процес розробки;

RPA – Robotic Process Automation;

RPM – Robotic Process Management;

BPI – Business (Process) Intelligence.

ВСТУП

У сучасному світі розвиток технологій протікає вкрай швидко та стрімко, інформація є основним ресурсом планети.

Сучасний рівень інтернет технологій дозволяють здійснювати автоматичні збирання, зберігання та обробку інформації навіть малим компаніям. При цьому, незважаючи на те, що компанії мають велику кількість інформації, що збирається, як про свою організацію, так і про своїх клієнтів, однією з найбільших проблем для нині існуючих організацій, став вкрай низький рівень інформативності про реальні процеси, що протікають на підприємстві. Неправильне розуміння власних процесів розробки та часті помилки в принципах роботи тих чи інших функцій компанії призводять до значних витрат та проблем у виробництві.

У сучасних реаліях, коли моделювання процесів розробки стало однією з ключових менеджерських компетенцій, уміння менеджменту компанії грамотно створювати та налагоджувати існуючі процеси розробки є основою будь-якої успішної компанії. Моделювання процесів розробки у великих компаніях часто є таким самим складним завданням, як і підтримка їх у актуальному та оптимальному стані.

Виходячи з цього, можна зробити висновок про необхідність створення методу, що має комплексний підхід до обліку виконання робіт в ІТ проектах.

Для вирішення задачі дослідження методів контролю виконаних у ІТ-проектах робіт з використанням технології Process Mining потрібно:

- дослідити гнучкі технології розробки ІТ проекту;
- дослідити методи контролю виконання робіт ІТ проекту;
- проаналізувати методи та технології Process Mining;
- розробити удосконалений метод контролю виконання робіт ІТ проекту на основі досліджених даних;
- зробити експериментальну перевірку удосконаленого методу контролю виконання робіт ІТ проекту.

1 ДОСЛІДЖЕННЯ ПІДХОДІВ ДО МОДЕЛЮВАННЯ ПРОЦЕСІВ

РОЗРОБКИ

1.1 Структуризація процесів розробки

Процесний-підхід - це метод управління, який остаточно сформувалася в минулому столітті. Відповідно до цього методу діяльність організації сприймається як сукупність процесів.

Основним поняттям, що використовує процесний підхід є процес. Процес — це є набір взаємозалежних дій, які перетворюють вхідні дані у результати. Важливою частиною процесу, яка відображена в цьому визначенні, є систематичність дії. Дії процесу повинні повторюватися, а не випадково.

Процесний підхід був розроблений та застосований для створення горизонтальних зв'язків в організаціях. Підрозділи та співробітники, залучені в один процес, можуть самостійно координувати роботу всередині процесу та вирішувати проблеми без залучення вищого керівництва.

Процесний підхід базується на кількох принципах. Реалізація цих принципів може значно підвищити ефективність, але також вимагає високої корпоративної культури. Перехід від функціонального до управління процесами вимагає від співробітників постійної спільної роботи, навіть якщо вони можуть належати до різних відділів. Ефективність принципів, закладених у процесному підході, буде залежати від того, наскільки ця спільна робота може бути забезпечена.

Впроваджуючи управління процесами, важливо дотримуватися наступних принципів:

Принцип взаємозв'язку процесів. Організація є мережу процесів. Процесом є будь-яка діяльність, де має місце виконання робіт. Усі процеси організації взаємопов'язані між собою;

Принцип затребуваності процесу. Кожен процес повинен мати на меті, а його результати повинні бути затребувані. У результатів процесу має бути свій споживач внутрішній чи зовнішній.

Принцип документування процесів. Діяльність у процесі необхідно документувати. Це дозволяє стандартизувати процес та отримати базу для зміни та подальшого вдосконалення процесу;

Принцип контролю процесу. Кожен процес має початок та кінець, які визначають межі процесу. Для кожного процесу в рамках заданих меж мають бути визначені показники, що характеризують процес та його результати;

Принцип відповідальності за процес. У виконанні процесу можуть бути задіяні різні фахівці та співробітники, але відповідати за процес та його результати має одна людина.

Процес розробки – це систематизована послідовність потоку взаємопов'язаних операцій, що має безперервний зв'язок з іншими процесами підприємства і використовує ресурси підприємства для отримання на виході кінцевого результату у вигляді продукту чи послуги. Споживачами процесу розробки вважаються учасники, які користуються основними результатами (продуктами, роботами, послугами) окремих бізнес-процесів розробки.

Приклад структури процесу розробки представлено на рисунку 1.1



Рисунок 1.1 - Приклад схеми процесу розробки

Управління процесами організації є складною процедурою, яка має враховувати багато факторів, серед яких різновиди процесів, які

відбуваються у компанії. Важливість класифікації процесів підкреслюється необхідністю ідентифікації всіх процесів компанії з метою

більш ефективного управління ними. Всі процеси можна поділити на дві великі групи: процеси та другорядні процеси. Перша група процесів безпосередньо створює цінність для кінцевого споживача, в той час як другорядні процеси не створюють додану вартість, а необхідні для підтримки основних процесів та їх вдосконалення з метою досягнення нового якісного рівня функціонування організації.

Допоміжні процеси спрямовані на підтримку основних, додають вартість продукції. Згідно цій теорії, на підприємстві існує дев'ять основних видів діяльності, з яких п'ять з основної діяльності зображені на рисунку 1.2:



Рисунок 1.2 – Основні види діяльності

та чотири з допоміжної діяльності:

- виробнича інфраструктура;
- управління;
- розвиток технологій;

– закупки.

Таким чином, потреба у регулюванні великої кількості різнорідних дій та процесів, що відбуваються на підприємстві, призводить до підвищення ролі процесного підходу, в рамках якого процеси господарської діяльності розглядаються як процеси підприємства, а підприємство в цілому як система розробки, що зорієнтована на певні результати діяльності

1.2 Дослідження моделей процесу розробки

Зазвичай модель розробки формується з метою впровадження поліпшень процесу управління, настає час коли керівництво приходить до думки, що підприємство потрібно переводити на нову ступінь розвитку, наприклад вийти на зовнішній ринок, підвищити якість послуг або продукту, що виробляються. Системи ПР зазвичай включає у собі вбудовані засоби моделювання діяльності підприємства, що дозволяють:

– візуалізовувати діяльність підприємства, що забезпечує для керівництва можливість оцінювати явні недоліки та знаходити джерела напрямів удосконалення;

– значно зменшити час налаштування ІС під особливості підприємства; Існує кілька типів процесів розробки (рисунок 1.3).

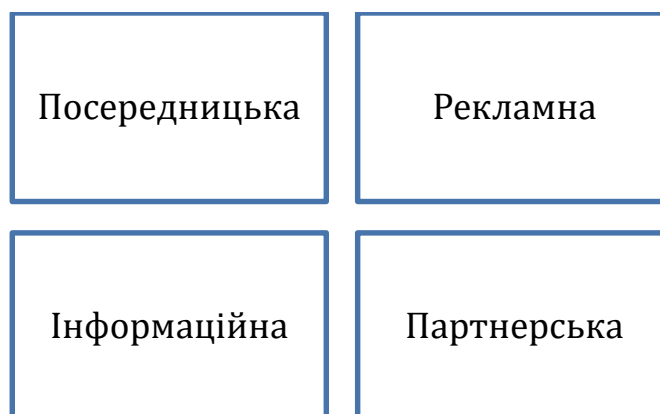


Рисунок 1.3 – Типи процесів

Посередницька – учасники ринку, котрі виступають третьою зацікавленою стороною між продавцем та покупцем, та допомагають з укладанням угод. Зазвичай, вони базуються на ринках які орієнтовані на обслуговування споживача.

Як правило, брокерські витрати є або комісіями, або процентом за кожну ідеальну транзакцію. Політика платежів може бути будь якою.

До посередницької моделі відносяться:

- біржа – пропонує розширений асортимент послуг для клієнта, що охоплює весь процес угоди, від проведення переговорів та контролю за виконанням та до ринкової оцінки товару. Біржі існують як самостійно, так і за підтримки промислових підприємств;

- система обліку та виконання замовлень – приймають заявки на продаж або купівлю товарів чи послуг;

- система зібрання запитів. Покупець встановлює кінцеву ціну певної послуги, а посередник контролює виконання заявки;

- аукціонний посередник – підтримує торговельні аукціони продавців. Посередницькі витрати продавця це плата за розміщення своєї пропозиції, а також комісійні, що безпосередньо залежать від величини транзакції. Аукціони дуже різняться за умовами проведення торгів пропонованих послуг або товарів та правил пропозиції ціни;

- платіжний посередник виступає в ролі третьої сторони для проведення платежів між продавцями та покупцями, контролюючи виконання обговорюваної угоди;

- дистриб'ютор це великай каталог, що має зв'язки з величезною кількістю оптовими та роздрібними покупцями та виробників продукції;

- пошукова система ПЗ або для контролю наявності та ціни товару або послуги, необхідним покупцю або початковим місцезнаходженням цієї інформації;

- віртуальний ринок – або віртуальна площадка інтернет-магазинів, являє собою деякий хостинг сервіс для онлайн-торгівлі. Доходи посередника

складаються з декількох факторів, з плати за початкове розміщення, щомісячні платежі за підпискою або оплати за різні транзакції. Також може надавати сервіси грошових транзакцій.

Рекламна модель в інтернет просторі виникла із моделі медіа-мовлення. Виконавцями, в цьому випадку, є веб-додатки, що надають контент, сервіси, суміщені з рекламними матеріалами банери. Реклама у банері може бути єдиним джерелом фінансового доходу для посередника. Посередник зазвичай це творець контенту або розповсюджувач контенту. Рекламна модель відпрацює краще, якщо трафік її відвідувачів великий чи вкрай спеціалізований.

До рекламної моделі відносяться:

- портал, як правило, є пошуковою системою, яка містить в собі контент і послуги різного плану. Велика кількість відвідувачів потрапляє на різні рекламні банери що створює прибуток для сервісу та дозволяє розвивати його;
- контекстна та поведінкова реклама – вбудовуваний у власні програми рекламний софт. Наприклад, плагін для браузера, котрий допомагає користувачу автоматичною ідентифікацією та заповненням форм, а також блокатори спливаючих вікон або рекламних посилань під час серфінгу сайтів. Контекстна реклама може бути таргетованою, що підстроюється під індивідуальну активність користувача;
- дошки оголошень це перелік різноманітних позицій продажу чи бажаних до придбання. Плата зазвичай поширена за розположення в лістингу, але, так само можливо і плата за доступ до даних;
- платне розміщення - продаж плейсментів для реклами, або реклама, що відображається згідно з окремими пошуковим запитами користувача;
- реєстрація користувачів - сайти з контентом який доступний безкоштовно, але потрібно пройти невелику реєстрацію з наданням демографічних даних. Реєстрація дає можливість стежити за особливостями серфінгу користувача, внаслідок чого є накопичення даних для оцінки та планування таргетованих рекламних кампаній;

– таргетована реклама – засновником напряду є Google, який розширив прицільність пошукової реклами на весь інший Інтернет. Google визначає зміст веб-сторінки, а потім автоматично надає відповідну рекламу, коли відвідувач відвідує цю сторінку.

Інформаційна - інформація про споживачів та їх купівельні звички є дуже цінною, особливо якщо зібрана інформація може бути проаналізованою та її можна використати у таргетованому маркетингу. А незалежно зібрані дані про товари можуть стати у нагоді споживачам. Інфопосередники надають інформацію як покупцям, так і продавцям.

Метапосередник – допомагають укладати угоди між покупцями та продавцями, надаючи різноманітні додаткові послуги та інформацію, крім натурального бартеру товарів та послуг між сторонами.

Рекламна мережа – надає місця на сайті з банерною рекламою. Рекламні мережі зазвичай збирають різні дані про веб-відвідувачів, які можна використовувати для аналізу маркетингової ефективності.

Стимулюючий маркетинг це програма яка сприяє підвищенню зацікавленості споживачів, наприклад пропонують купони або талони для вигідної закупівлі у своїх партнерів які займаються роздрібною торгівлею.

Партнерська - ця модель пропонує можливість здійснити покупку одразу з кількох різних сайтів. За посиланням сайту-партнера можна перейти до виробника.

Плата за переходи – сайт, який дає змогу заробляти, платять своїм партнерам за переходи відвідувачів за різними посиланням;

Банерний обмін – торгівля банерів які вбудовані в різні сайти; продаж всередині мережеских сайтів партнерів.

Поділ доходу (Revenue Sharing) – пропозиції комісійних від продажу залежно від кількості переходів відвідувачів з сайтів партнерів.

1.3 Аналіз методів та технології Process Mining

Process Mining це загальна назва підходів та методів, які призначені для вдосконалення та аналізу інформаційних систем та їх процесів завдяки вивченню системних даних про операції які було виконано. Основа ідеї полягає у тому що потрібно отримати дані поведінки процесу, його структури, які можна отримати з журналів подій, що створює інформаційна система під час її безпосереднього функціонування.

Як правило, методи процесної аналітики використовуються в таких ситуаціях, коли відсутні модель системи та формальний опис або вони мають недостатньо реальну поведінку відповідно до системи.

Для побудови моделі можна використовувати інструменти які допомагають функціонувати інформаційним системам (наприклад: системи управління потоками робіт, ERP, CRM, та інші).

Методи процесної аналітики поділяють на три блоки:

- дослідження: модель процесу будується виходячи з журналу подій, що містить запис функціонування інформаційної системи за певний проміжок часу;
- перевірка відповідності: існуюча модель порівнюється з журналом подій, аналізуються виявлені невідповідності в поведінці реальної системи та поведінці, що моделюється;
- удосконалення: існуюча модель удосконалюється шляхом розширення поведінки, що моделюється, або підвищення ефективності моделювання з використанням інформації, отриманої в ході перевірки відповідності моделі та журналу подій.

Основні методи процесної аналітики які використовуються в інформаційних системах наведені на рисунку 1.3

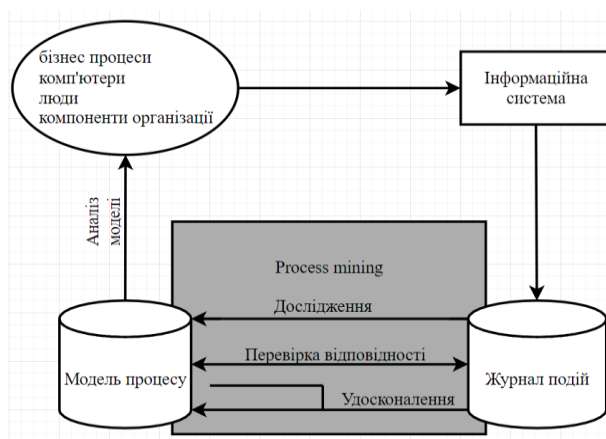


Рисунок 1.3 – Основні методи process-mining в IC [12]

Process Mining широко застосовують у багатьох галузях бізнесу та управління, тому на відміну від звичайного моделювання - моделі, які можна одержати в результаті роботи алгоритмів Process Mining, будуються на основі записів реальних подій, завдяки цьому більш реально відображають інформацію про те, що відбувалося насправді в системі, а не про те, як було заплановано. Окрім цього, моделювання, в деяких областях, може бути трудомістке або дуже затруднене, тоді як Process Mining дає змогу автоматизувати процес.

Джерелом інформації для алгоритмів процесної аналітики виступають логи роботи інформаційних систем. Однак для аналізу може бути використаний не будь який протокол. Для застосування методів РМ потрібно щоб логи містили необхідну інформацію про події а саме:

- впорядкування подій за часом їх виконання;
- всі події, мають бути ідентифіковані з екземплярами процесів;
- різнотипні події повинні відрізнятися.

Сфера розробки процесів розширилася від інструменту для спеціаліста з даних, який використовується в проекті вдосконалення, до безперервного застосування процесу інтелекту в масштабах підприємства.

Інструменти технологічного майнінгу, які підтримують лише конструювання DFG з частотами та часом, як правило, використовуються

лише в невеликих проектах. Ці проекти часто мають обмежений обсяг і тривалість. В результаті мало хто використовує результати, і немає підтримки для постійного вдосконалення.

З огляду на інвестиції, необхідні для підготовки даних, часто краще застосовувати процес видобутку в масштабах всього підприємства, коли багато людей використовують результати щодня.

Немає сенсу розглядати процес видобутку як одноразову діяльність. Проте в масштабах підприємства, безперервне застосування процесу інтелекту вимагає значних ресурсів з точки зору обчислювальної потужності та керування даними (наприклад, для вилучення даних і перетворення їх у правильний формат).

Щоб знизити поріг для більшої спільноти майнінгу процесів в організації, потрібно створити налаштовані інформаційні панелі. Тому технологічний видобуток має підтримуватися керівництвом вищого рівня, щоб зрозуміти, в якому масштабі він є найбільш ефективним.

Спочатку зусилля з розробки процесів були зосереджені на відкритті процесу. Однак з часом стало зрозуміло, що відкриття процесу є лише відправною точкою для вдосконалення процесу. Можна спостерігати поширення методів перевірки відповідності та аналізу продуктивності. Крім того, процес інтелекту часто поєднується з методами інтелектуального аналізу даних і машинного навчання, щоб знайти основні причини неефективності та відхилень. Хоча виявлення процесу залишатиметься важливим, увага переходить до кроків після відкриття за допомогою оптимізації, машинного навчання та моделювання.

Третя тенденція - це зміщення фокусу з огляду на назад до перспективного. Традиційні технологічні методи видобутку починають з історичних даних. Це можна використовувати для діагностики проблем відповідності та відповідності.

Однак організації часто більше цікавляться тим, що відбувається зараз або що буде далі. Видобуток процесів, що розглядається в минулому, можна

використовувати для фундаментального покращення процесів, але надає незначну підтримку для повсякденного управління процесами. Таким чином, дані про події необхідно постійно оновлювати, а методи видобутку процесів повинні мати можливість аналізувати випадки, які все ще виконуються. Це необхідно для контролю та впливу на запуснені екземпляри процесу. Деякі з комерційних інструментів видобутку процесів надають чудові можливості для відображення поточного стану процесу.

Наступним кроком є застосування більш перспективних методів, здатних передбачити, що станеться з окремими випадками та де, ймовірно, виникнуть вузькі місця.

Методи оперативної підтримки (тобто виявлення проблем з відповідністю та продуктивністю під час виконання, передбачення таких проблем та рекомендація дій) були навколо більше десятиліття. Однак їх якість залишає бажати кращого.

Проблема полягає в тому, що випадки (тобто екземпляри процесу) сильно впливають один на одного, коли конкурують за ресурси. Крім того, може існувати ряд контекстуальних факторів, що впливають на процеси. Просто застосовуючи наявне машинне навчання і методів аналізу даних, не можна отримати жодних надійних прогнозів. Тому потрібна додаткова робота.

Четверта тенденція — посилення уваги на фактичному вдосконаленні процесу. Майнінг процесів має тенденцію зосередитися на діагностиці, а не на втручаннях, які мають послідувати.

Статистика, отримана в результаті видобутку процесів має бути дієвим. Таким чином, процес видобутку все частіше поєднується з роботизованим процесом

Розробку процесів можна використовувати для визначення ручних завдань, які можна автоматизувати та моніторинг програмних робіт. Це дозволяє автоматизувати процеси, для яких традиційний робочий процес автоматизація була б занадто дорогою.

Наведено спектр варіантів процесу. Високочастотні варіанти є кандидатами на автоматизацію, але варіанти з меншою частотою не можуть бути автоматизовані економічно ефективно. RPA допомагає зрушити межі, коли (часткова) автоматизація все ще є економічно ефективною.

Цікаво, що процес видобутку можна використовувати до і після автоматизації для будь-якої суміші варіантів процесу. Однак RPA — це лише один із багатьох способів перетворити діагностику технологічного видобутку дії. Тут я також хотів би ввести термін роботизоване керування процесами (RPM) для позначення автоматичного процесні втручання. На відміну від RPA, RPM не автоматизує етапи операційного процесу. RPM перекладає діагностика видобутку процесу в управлінські дії. Наприклад, коли виникає вузьке місце, RPM може виконувати такі дії, як сповіщення менеджера, інформування постраждалих клієнтів, призначення додаткових працівників тощо.

Іншим прикладом може бути визначення пріоритетності цільової аудиторської діяльності при значному збільшенні процесу виникають відхилення. Ці приклади показують, що діагностика, надана технологічним видобутком, є просто відправна точка.

Описують три методи для виявлення процесу: один з використанням нейронних мереж, одна з яких використовує суто алгоритмічний підхід, а друга — марківський підхід. Останні два підходи автори вважають найбільш перспективними.

Суто алгоритмічний підхід створює кінцевий автомат, де є стани злиті, якщо їхнє майбутнє (з точки зору можливої поведінки на наступних k кроках) є ідентичні. Марковський підхід використовує суміш алгоритмічних і статистичних методів і здатний боротися з шумом.

Зверніть увагу, що результати, представлені в обмежуються послідовною поведінкою. Споріднений, але в іншому домені, є робота яка також використовує марковівський підхід, обмежений до послідовні процеси. Кук і Вольф поширюють свою роботу на паралельні процеси. Вони

пропонують конкретні показники (ентропія, кількість типів подій, періодичність, і причинно-наслідковий зв'язок) і використовуйте ці показники, щоб виявляти моделі з потоків подій. Однак вони не забезпечують підходу до створення явних моделей процесів.

Надають міру кількісної оцінки розбіжностей між а модель процесу та фактичну поведінку, як зареєстровані з використанням даних на основі подій. Ідея застосування аналізу процесів у контексті управління робочими процесами була першою запроваджено в. Ця робота заснована на графіках робочого процесу, які натхненні продукти робочого процесу, такі як робочий процес IBM MQSeries (раніше відомий як Flowmark) і InConcert. У цій роботі визначено дві проблеми.

Перша проблема полягає в тому, щоб знайти графік робочого процесу, який генерує події, що з'являються в заданому журналі робочого процесу.

Друга проблема — знайти визначення реберних умов. Наведено алгоритм вирішення першої задачі.

Підхід значно відрізняється від інших підходів: оскільки характер робочих графіків не існує необхідності визначити природу з'єднань і розщеплень. Графіки робочого процесу використовують маркери true та false, які не допускають циклічних графіків.

Тим не менш, проводиться ітерація, перераховуючи всі випадки даного завдання, а потім згортання графіка. Однак отриманий конформ графік не є повною моделлю. Преставлено інструмент, заснований на цих алгоритмах. Розроблено гірничий інструмент, придатний для відкриття ієрархічно структуровані процеси робочого процесу. Для цього потрібні всі розбиття та з'єднання повинні бути збалансованим.

Також розглядають питання технологічного видобутку корисних копалин в контексті управління робочим процесом з використанням індуктивного підходу.

Підхід також допускає паралельність. Він використовує стохастичні графіки завдань як проміжне представлення і створює робочий процес модель,

описана мовою моделювання ADONIS. На вступному етапі завдання вузли об'єднуються та розбиваються, щоб виявити основний процес. Помітна відмінність від інших підходів полягає в тому, що одне і те ж завдання може з'являтися кількома разів у моделі робочого процесу, тобто підхід дозволяє повторювати завдання.

Техніка генерації графів подібна до підходу. Природа розщеплення та з'єднання (тобто І або АБО) виявляється на етапі перетворення, де граф стохастичного завдання перетворюється на модель робочого процесу ADONIS з блоково-структуровані розбиття та з'єднання.

На відміну від попередніх статей, наступні статті характеризуються акцентом на процесах робочого процесу з одночасністю поведінки (замість додавання спеціальних механізмів для виявлення паралелізму). Для побудови використовується евристичний підхід із використанням досить простих метрик так звані «таблиці залежностей/частот» і «графіки залежностей/частот».

Представлений інший варіант цієї методики на прикладах з сфери охорони здоров'я. Попередні результати, які надають лише евристичні дані та фокусуються на таких питаннях, як шум.

Глибинний аналіз процесів можна розглядати як інструмент у контексті Business Intelligence. Описується набір інструментів BPI на вершині диспетчера процесів HR. Набір інструментів BPI включає так званий «BPI Process Mining Engine». Однак цей двигун не надає жодної техніки, як обговорювалося раніше. Замість цього він використовує загальні інструменти копіювання, такі як SAS Enterprise Miner, для генерації дерев рішень, пов'язаних атрибутів випадків з інформацією про шляхи виконання (наприклад, тривалість).

Для аналізу робочих процесів зручно мати так зване «сховище даних процесу» для зберігання слідів аудиту. Таке як сховище даних спрощує та прискорює запити, необхідні для отримання причинно-наслідкових зв'язків.

При проектуванні такого складу та супутні питання обговорюються в контексті журналів робочого процесу. Крім того, описується інструмент PISA, який можна використовувати для вилучення показників продуктивності з журналів робочого процесу.

Пізніший інструмент є комерційно доступним, а налаштована версія RRM — це моніторинг процесів персоналу (SPM) [16], який призначений для видобутку журналів Staffware.

Обчислювальна задача пошуку мінімального кінцевого акцептора, сумісного з заданими даними, є NP-складовою. Декілька більш загальних концепцій, які обговорюються в цих статтях, можна перекласти на область технологічного видобутку. Проблему можна інтерпретувати як проблему індуктивного висновку, визначену в термінах правил, простору гіпотез, прикладів і критеріїв успішного висновку. Порівняння з літературою в цій області викликає цікаві питання для аналізу процесів, наприклад, як працювати з негативними прикладами (тобто, припустимо, що крім журналу W є журнал V слідів, які неможливі, наприклад, додані експертом з області). Однак, незважаючи на численні зв'язки з роботою, є також багато відмінностей, наприклад, ми добуємо на чистому рівні, а не на послідовних або нижчих рівнях (наприклад, ланцюги Маркова, кінцевий стан машини або регулярні вирази).

Сильною аналогією з проблемою видобутку процесів є: чи можемо ми знайти найпростішу модель процесу, яка може генерувати ці сліди, враховуючи ряд слідів процесу. Багато питань, важливих у сфері вивчення мови, також є актуальними для аналізу процесів (тобто навчання лише на позитивних прикладах, як боротися з шумом, вимірювання якості моделі тощо).

Важливою відмінністю між областю граматичного висновку та доменом аналізу процесів є проблема паралельності в трасуваннях: паралельність здається неактуальною в області граматичного висновку. Незважаючи на цю важливу різницю, здається корисним дослідити, які теоретичні результати,

вимірювання та методи видобутку можна використати або оновити, щоб вони стали корисними для технологічного видобутку.

Досліджується, які перехідні системи можуть бути представлені (компактними) мережами Петрі (тобто, так звана задача синтезу). Хоча налаштування відрізняються, і наше поняття повноти набагато слабкіше, ніж знання системи переходів, існують пов'язані з цим проблеми, такі як повторювані переходи тощо.

Домінуючою перспективою технологічного видобутку є так звана перспектива контрольного потоку. Суть цієї перспективи полягає в упорядкованості завдань. Як зазначено, перспектива потоку керування може бути розширена, щоб включати інформацію про час (тобто події мають часові позначки). Однак, крім точки зору потоку управління, можна також розглянути:

- перспективу організації;
- перспективу інформації;
- перспективу застосування.

З точки зору організації, уточнюються організаційна структура та чисельність населення. Організаційна структура описує відносини між ролями (класи ресурсів на основі функціональних аспектів) і групами (класи ресурсів, засновані на організаційних аспектах), а також інші артефакти, що прояснюють організаційні питання (наприклад, відповідальність, доступність). ресурси, від людей до пристроїв, утворюють організаційну сукупність і розподіляються на ролі та групи.

Інформаційна перспектива має справу з контрольними та виробничими даними. Контрольні дані – це дані, що вводяться виключно для цілей управління процесом, наприклад, змінні, введені для маршрутизації. Виробничі дані – це інформаційні об'єкти (наприклад, документи, форми та таблиці), існування яких забезпечується не залежить від управління процесами.

Перспектива програми стосується програм, які використовуються для виконання завдань (наприклад, використання текстового редактора). У журналі подій можна знайти сліди цих інших перспектив. Подія в журналі може давати інформацію про ресурс (наприклад, працівник), який виконав відповідне завдання. Цю інформацію можна використовувати для отримання знань про перспективу організації (наприклад, ролі та групи, структури співпраці, ефективність співпраці тощо). Наприклад, процес розробки може бути використаний, щоб виявити, що випадки, які потребують співпраці двох конкретних працівників, мають значно більший час обробки.

Журнал може записувати «потік даних», наприклад, які змінні процесу оновлюються під час даної події та які їх значення. Такі журнали можна використовувати для отримання знань про інформаційну перспективу. Особливо цікаво зв'язати інформаційну перспективу з перспективою потоку керування. Наприклад, чи існує кореляція між часом потоку та певними змінними процесу (наприклад, великі замовлення займають більше часу), чи існує зв'язок між маршрутизацією справи та її змінними процесу (наприклад, випадки клієнтів, розташованих у певному регіоні? зазвичай вимагає кількох додаткових перевірок)? Аналогічно, журнал можна доповнити інформацією про програми, які використовуються отримати знання про перспективу застосування. Поки що більшість досліджень зусилля були зосереджені на перспективі потоку контролю. Тому це цікаво проблема включення організаційної точки зору, інформаційної точки зору, та/або перспективи застосування.

1.4 Постановка задачі дослідження

Існуючі методи обліку виконаних робіт у ІТ проектах використовують експертні оцінки та не враховують реальний час на виконання роботи. Однак для удосконалення процесу розробки потрібно враховувати послідовність

робіт у часі з урахуванням фактичних даних щодо моментів часу початку та завершення цих робіт. Тому дослідження щодо вирішення цієї проблеми є актуальним.

Об'єктом який треба дослідити під час роботи є процеси обліку виконаних робіт у ІТ проектах.

Предметом дослідження виступають моделі та методи обліку виконаних робіт у ІТ проектах

Метою кваліфікаційної роботи є дослідження методів контролю виконаних у ІТ-проектах робіт з використанням технології інтелектуального аналізу процесів.

Для досягнення мети потрібно провести вирішення наступних завдань:

- дослідження гнучких технологій розробки ІТ проекту;
- дослідження методів контролю виконання робіт ІТ проекту;
- аналіз методів та технології Process Mining;
- розробка удосконаленого методу контролю виконання робіт ІТ проекту на основі структуризації логу, що відображає послідовність виконання робіт із часовими мітками;
- експериментальна перевірка удосконаленого методу контролю виконання робіт ІТ проекту.

2 ДОСЛІДЖЕННЯ МОДЕЛЕЙ ПРОЦЕСІВ РОЗРОБКИ

2.1 Дослідження методів контролю виконання робіт ІТ проекту.

Цикл Демінга – це ефективна модель призначена для оптимізації та управління якістю, всіх бізнес-процесів а також процесів розробки ІТ проектів. Завдяки цій методиці, можна підвищити якість результатів у будь-якій сфері діяльності. Потрібно уважно вивчити всі процеси, щоб знайти недоліки, недоліки та усунути їх [5].

Цикл PDCA — це зрозуміла та проста модель яка допомагає покращити управління якістю. Згідно методів моделі, щоб забезпечити належний рівень якості, потрібно циклічно повторювати такі чотири кроки:

- планування;
- виконання;
- перевірка;
- корегування;

Оскільки модель циклічна - вона передбачає певну послідовність кроків, вона зображена на рисунку 2.1

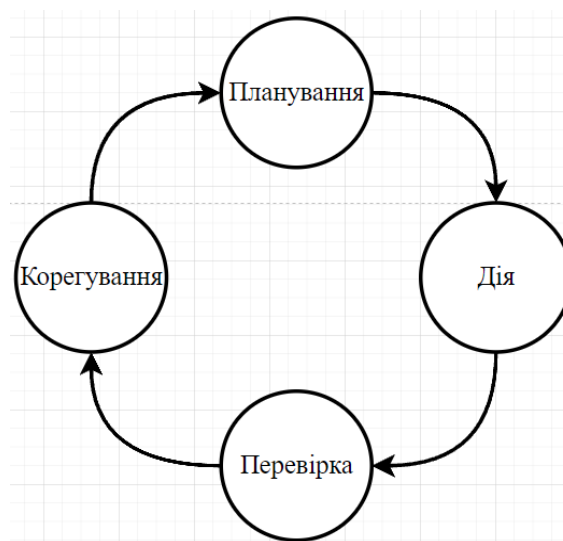


Рисунок 2.1 - Приклад циклу Демінга

Планування. Перший цикл PDCA каже про те що дії потрібно планувати до початку їх втілення. Цей крок охоплює:

- інформацію про потенціал;
- аналіз фактичного стану;
- розробку концепції планування [5].

Одним з найважливіших етапом управління проектами являється процес планування. Планування здійснюється постійно протягом усього проекту. На початку життєвого циклу проекту, зазвичай, розробляється неформальний попередній план - приблизне уявлення про те, що потрібно буде зробити в проекті. Рішення про вибір проекту багато в чому базується на оцінках попереднього плану. Офіційне та детальне планування проекту починається після прийняття рішення про його відкриття. Визначаються ключові події – формулюються віхи проекту, завдання, роботи та їх взаємозалежність.

Основні етапи процесу планування наведені в і включають дев'ять кроків. На кожному кроці керівник проекту може виявити неефективність або неможливість реалізації проекту і поставити питання про його закриття.

Робота зі складання плану включає всі етапи створення та реалізації проекту [5]. Він починається з розробки концепції проекту керівником проекту, продовжується вибором стратегічних рішень, розробкою деталей, укладенням договорів та виконанням робіт і закінчується завершенням проекту.

Під час фази проектування (також званої фазою проектування та архітектури) програмісти та системні архітектори використовують вимоги для розробки проектування системи високого рівня.

Різні технічні питання, що виникають у процесі проектування, обговорюються з усіма зацікавленими сторонами, включаючи замовника. Визначаються технології, які будуть використовуватися в проекті, навантаження на команду, обмеження, терміни та бюджет. Відповідно до заданих вимог підбираються найбільш підходящі дизайнерські рішення.

Затверджений проект системи визначає перелік розроблених програмних компонентів, взаємодію із сторонніми особами, функціональні характеристики програми, використовувані бази даних та багато іншого [5]. Проектування, як правило, закріплюється в окремому документі - документі проектної специфікації.

На цьому етапі для спрощення візуалізації процесу проектування використовуються так звані позначення — схематичне вираження характеристик системи, що розробляється. Основні позначення, які використовуються:

- макети - наприклад, прототип веб-сайту, намальований у Photoshop;
- блок-схеми;
- діаграми UML;
- діаграми ER.

Цикл PDCA дія. це фаза реалізації дій які були заплановані на попередньому етапі. Головне правило – робити все за розробленим планом, намагатися не порушувати жодного пункту.

На цьому етапі зазвичай починають розробку дизайну та безпосередньо програмного забезпечення.

Мета етапу розробки дизайну — зробити продукт зрозумілим, приємним і зручним для користувача.

На цьому етапі дизайнер активно намагається взаємодіяти з аналітиком проекту. Разом вони розробляють зрозумілий і зручний дизайн на основі підготовленого набору артефактів. Аналітик продумує логіку продукту і передає дизайнеру основу графічного інтерфейсу, зазвичай це схема приблизного розміщення елементів інтерфейсу, після чого дизайнер приступає до розробки візуальної частини. Дизайнер відповідає за зовнішній вигляд інтерфейсу, анімації, за те як різні елементи реагують на дії користувача. Для цього створюється графічні елементи, карта екранів, детальний прототип різних сценаріїв використання.

Далі з розробленим дизайном починає працювати відділ розробки,

розглянемо на прикладі веб розробки. Першим розробки займається відділ фронтенд. Завданням цього відділу є оживити екрани та форми розроблені дизайнерами, з яким користувач буде взаємодіяти під час використання кінцевого продукту. Вони створюють елементи інтерфейсу, логічно та функціонально зв'язуючи екрани один з одним за допомогою мови розмітки HTML, мови програмування Javascript, таблиць стилів CSS, різних фреймворків та бібліотек. Результатом роботи цього відділу є сторінки майбутнього сайту, розроблені точно за макетами які розробляв відділ дизайну.

Після завершення роботи фронтенд відділу до роботи приступає відділ бекенд. Їх задачу є серверна розробка. Вони мають переконатися, що відповідь від сервера надходить до клієнта, а розроблені блоки функціонують за бажанням.

Перевірка, тестування досягнутого результату це майже щоденний моніторинг виконання етапів та отриманих показників. Оцінка результатів розробки дозволяє визначити, чи все зроблено, як було заплановано на першому етапі циклу Демінга, чи необхідно виконати ще один цикл для отримання бажаного ефекту, чи відбулися покращення.

Функціональне тестування це процес оцінки поведінки додатку, що дозволяє перевірити що всі функції які були розроблені, поведуть себе так, як потрібно. Для коректної роботи продукту всі процеси повинні працювати так, як це було заплановано. Функціональне тестування може протікати використовуючи попередньо підготовлені тестові сценарії або методи дослідного тестування.

Тестування сумісності це оцінка обробки додатку різними операційними системами, браузерами і пристроями з різною роздільною здатністю екрана. Прикладом цього типу тестування є перевірка сумісності продукту з усіма останніми версіями веб браузерів Chrome, Firefox, Opera, Safari та Windows, Android, IOS, MacOS.

Тестування продуктивності або перфоманс це перевірка, спрямована на визначення крайніх меж продуктивності програми. Після тестування формується звіт, який містить таку інформацію:

- дані максимально-допустиму кількість користувачів, яких система може обробити;
- статистика виниклих помилок;
- діаграми залежності кількості користувачів до продуктивності програми;
- часу відгуку сервера для обробки найважливіших операцій;
- інформація про стабільність роботи системи та здатність витримувати постійне навантаження.

Останнім етапом циклу є коригування, рисунок 2.5. На цьому етапі проводиться фінальне вирішення проблем знайдених на попередньому кроці а також проводиться огляд коду.

Огляд коду – це процес, під час якого якому один або декілька розробників переглядають готовий код з попередніх етапів розробки, для того щоб переконатися у тому що:

- код виконує функції які повинен;
- в коді немає помилок чи проблем, які могли бути не були враховані під час розробки;
- код відповідає всім стандартам якості та вимогам [3].

Перш за все, необхідно визначити з яким середовищем для огляду коду можна буде працювати. Найпопулярнішими є GitHub, GitLab, BitBucket. Кожен із цих сервісів включає в себе набір функцій які потрібні для проведення якісного огляду коду, а також історію змін та обговорень.

Огляд коду допомагає знайти критичні та не дуже помилки на ранніх етапах розробки і допомагає позбутися від заплутаних рішень.

Розробник, який знає, що його роботу будуть перевіряти колеги, намагається робити код акуратнішим і більш організованим. На виході є код,

який з легкістю розуміють декілька розробників, а це є ознакою, що він більш якісний [3].

Як тільки команда розробників знайома з кодом на достатньому рівні, його стає легше передавати між декількома розробниками що дозволяє прискорити процес розробки. Якщо виникне потреба щоб код початий одним розробником, який з якихось причин не може його завершити - інший член команди може швидко підхопити роботу і виконати її з легкістю.

Завдяки Code Review, так званий фактор шини, або «коефіцієнт шини», зменшується. Наприклад, над проектом працюють десять розробників, якщо половина з них з якихось причин не зможуть продовжити роботу, то решта зможе завершити її. Отже огляд коду або етап коригування в будь якому випадку підвищує якість продукту [3].

2.2 Методологія Scrum

Scrum - це набір принципів, на яких будується процес розробки, що дозволяє в жорстко фіксовані і невеликі за часом ітерації, звані спринтами (sprints), надавати кінцевому користувачеві ПЗ з новими можливостями, для яких визначено найбільший пріоритет. Нові можливості, над якими належить працювати в ході спринту, визначаються на етапі планування і не можуть змінюватися на його протязі. Час спринту регламентується також етапі планування і незмінно, що робить процес створення продукту передбачуваним і гнучким [14].

Основні ролі учасників проекту, що розробляється за Scrum зображені на рисунку 2.2

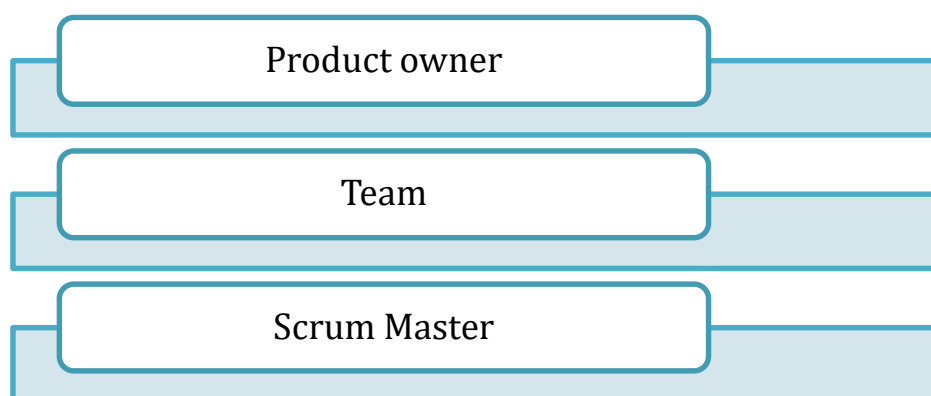


Рисунок 2.2 - Основні ролі учасників проекту

Product owner – це людина, яка знає, як хоче, щоб виглядав продукт, тобто. він формує список вимог, що заносяться в беклог продукту йдуть у розробку

Team – це команда, яка бере участь у розробці. Дуже важливо, щоб вона була самоорганізованою і самоврядною, а також могла самостійно виділити якісь із запропонованих вимог необхідно розробити в першу чергу.

Scrum Master – це людина, яка стежить за тим, щоб дотримувалися правил.

Спринт - ітерація в скрамі, під час якої створюється функціональне зростання програмного забезпечення. Жорстко фіксований за часом, його тривалість становить від 2 до 4 тижнів. Проте, різні команди підбирають довжину спринту відповідно до специфіки своєї роботи, складу команд та вимог, часто методом спроб та помилок.

Вважається, що чим коротшим є спринт, тим більш гнучким є процес розробки, релізи виходять частіше, швидше надходять відгуки від споживача, менше часу витрачається на роботу в неправильному напрямку. Завдання, які вирішуються в ході спринту, беруться з беклогу.

Беклог продукту — це перелік вимог до функціональності, впорядкований з їхньої ступеня важливості, які підлягають реалізації. Елементи цього списку називаються історіями користувача (user story) або

елементами беклога (backlog items). Беклог проекту відкритий для редагування для всіх учасників скрам процесу.

Беклог спринту - містить функціональність, вибрану власником проекту з Беклог проекту. Усі функції розбиті по завданням, кожна з яких оцінюється скрам-командою. Кожен день команда оцінює обсяг роботи, який потрібно зробити для завершення спринту.

Варто зазначити, що невід'ємною рисою скраму є щоденні мітинги, в яких пояснюється хто скільки зробив і залишилися/з'явилися у нього якісь питання.

2.2 Методологія Agile

Agile — це ітеративний підхід до управління проектами та розробки програмного забезпечення, який допомагає командам швидше і з меншою кількістю головного болю надавати цінність своїм клієнтам.

Замість того, щоб робити ставку на запуск «великого вибуху», гнучка команда виконує роботу невеликими, але витратними кроками. Вимоги, плани та результати постійно оцінюються, тому команди мають природний механізм швидкого реагування на зміни.

Більшість гнучких методологій націлені на мінімізацію ризиків шляхом зведення розробки до серії коротких циклів, які називаються ітераціями, які зазвичай тривають два-три тижні.

Кожна ітерація сама по собі виглядає як програмний проект у мініатюрі та включає всі завдання, необхідні для видачі міні-приросту за функціональністю: планування, аналіз вимог, проектування, програмування, тестування та документування.

Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі, що гнучкий програмний проект готовий до випуску

наприкінці кожної ітерації. Після закінчення кожної ітерації команда переоцінює пріоритети розробки.

У той час як традиційний підхід водоспад передбачає, що один розробник завершує свою частину проекту, а потім передає наступному учаснику, agile закликає до співпраці міжфункціональних команд. Відкрите спілкування, співпраця, адаптація та довіра між членами команди є основою agile.

Хоча керівник проекту або власник продукту зазвичай ставить пріоритет роботи, яку потрібно виконати, команда приймає рішення про те, як буде виконуватися робота, самоорганізуючись навколо детальних завдань і завдань.

Agile не визначається набором церемоній або конкретними методами розробки. Швидше, agile — це група методологій, які демонструють прихильність до жорстких циклів зворотного зв'язку та постійного вдосконалення.

Основа гнучкої методології — розбиття проектів на маленькі робочі шматочки, звані історіями користувача. Відповідно до пріоритетності завдання вирішують у рамках коротких двотижневих циклів (ітерацій)

Визначення пріоритетів дуже важливо, по суті, це і є найголовніше в гнучкій методології розробки. Вам потрібно переконатися, що команда сфокусована на одному спринті або на найважливішому результаті. Це буде гарантією того, що ви досягнете ваших бізнес-цілей. Так ваша команда не загубиться в потоці запитів і вимог і знатиме, що кожне підзавдання, над яким вона працює в даний момент, є важливим для прогресу всього проекту.

Першочерговим завданням Agile вважається важливість команди, мотивація співробітників, розширення можливостей для самореалізації, а також самостійність прийняття важливих рішень. Так, мотивований член команди інтенсивніше вкладається в роботу, надаючи якісний та швидкий результат.

Доброзичлива обстановка у колективі сприяє зацікавленості працювати на благо товару (сервісу). А розширення повноважень дозволяє оперативно

вирішити поставлене завдання, скорочуючи час, який треба було б витратити на погодження.

Крім цього, важливу роль відведено плануванню командою, де кожен є цінною частиною одного організму. Спільне обговорення завдань, внесення пропозицій та вказівку на помилки призводить до швидкої адаптації та усунення недоліків у конкретній ситуації.

Успіх справи залежить від того, наскільки це задовольняє потреби клієнта. З цією метою Agile-підхід має кілька дієвих інструментів:

- зосередження на потребах є не тільки у власника бізнесу, а й у всіх учасників проекту;
- кожен співробітник розуміє запити клієнта, а також його побажання, очікування та страхи щодо продукту, що випускається;
- працівники «відчують» замовника та сміливо пропонують свої бачення на певній ділянці роботи.

Таким чином, демократичний формат робочого процесу передбачає активну участь усіх працівників, а також їхню мотивацію.

Головними перевагами методології є:

- гнучкість та здатність підлаштовуватися під запити замовника та ринку;
- одержання на виході досконалого продукту, що перевіряється в кінці кожного спринту;
- відмінна пристосованість.

До недоліків відносять:

- відсутність часто конкретного терміну здачі всього продукту через регулярне внесення змін;
- необхідність наділяти команду повноваженнями;
- невідворотність культурних змін у всій організації;
- тривалість та вартість Agile-трансформації.

Agile – сучасний та зручний формат управління, здатний вирішити проблеми класичного проектного менеджменту. Тож високо цінується там, де

потрібно створювати інноваційні проекти. Разом про те, дана концепція ефективна у процесній діяльності (наприклад, кафе швидкого обслуговування). І також ідеально підходить для розвитку проектів у сфері ІТ-технологій, маркетингу, реклами та PR-діяльності.

2.3 Удосконалений метод контролю виконання робіт

У основі методу обліку та його удосконалення було обрано технологію Process Mining. Ця технологія не є новою, але свою популярність набирає саме зараз.

Process mining дозволяє відтворити та візуалізувати схему процесу розробки саме так, як вони протікають насправді. Бачення та оцінка ситуації від імені голови будь-якого з департаментів компанії неминуче буде більш суб'єктивною.

Обробка та аналіз логів дозволяє отримати безоцінну та комплексну модель події, виявити невідповідності нормативам, дублювання функціоналу та надлишкові витрати.

Дані про тривалість та особливості виконання процесу можна накласти на граф виконуваного процесу. Це дозволяє виявити затримки у діях окремих виконавців та цілих підрозділів, неефективні взаємозв'язки між користувачами, зацикленості та приховані недоліки у процесі.

Програмне забезпечення для інтелектуального аналізу процесів може допомогти організаціям легко збирати інформацію з корпоративних систем та надавати докладну інформацію про те, як виконуються ключові процеси.

У міру виконання роботи формуються журнали, наприклад, ось таких подій:

- створення репозиторію;
- створення нової гілки;
- комміт, пуш реквест;

- мердж;
- пул реквест;

Журнали дозволяють побачити, як насправді відбувається розробка проекту, у тому числі хто її виконав, скільки часу це займає та як відрізняється від середнього.

Аналітика процесів створює ключові показники ефективності процесу, що дозволяє компанії зосередитись на пріоритетних кроках для покращення [11].

Але на даному етапі розвитку ми стикаємося з проблемою сумісності сучасних гіт репозиторіїв та програмного забезпечення обробки журналу подій. Гіт баш не дає змогу з легкістю отримати дані в тому вигляді в якому його може прийняти програма обробки журналу подій.

Тому було прийняте рішення зробити додаток який дозволяє сформувати журнал подій вже адаптованим для роботи с такими програмами їх обробки як Fluxicon Disco.

Етапи удосконалення методу контролю виконання робіт рисунок 2.6:

- метод формує логи для побудови моделі виконання робіт ІТ проекту в розрізі виконавців та окремих робіт безпосередньо з використаних репозиторіїв;
- метод використовує дані із платформ хостінгу ІТ-проектів таких як GitHub або GitLab, які використовують розподілену системи управління версіями Git;
- в якості вхідних даних використовуються логи розробки у форматі JSON. Але цей формат не може бути використаний у програмі обробки журналу подій, тому що поді у ньому не є чітко структуровані. Тому було прийняти рішення що ісходні логи потрібно привести до чітко структурованого у вигляді таблиці формату csv;
- результатом роботи методу є граф фактичного виконання робіт, який дає можливість проаналізувати фактичну послідовність робіт, що створює умови для виявлення «вузьких місць» у процесі розробки ІТ проекту.

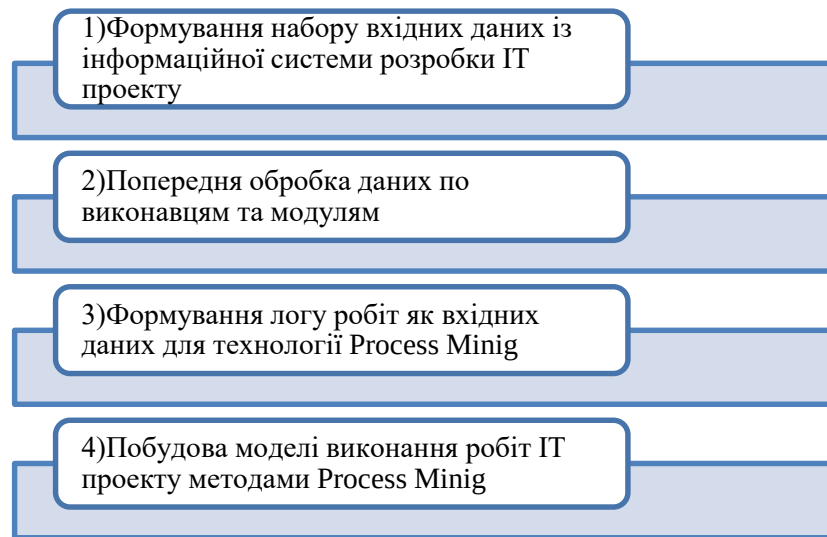


Рисунок 2.6 - Етапи удосконалення методу контролю виконання робіт.

Реалізація представлених етапів методу дає змогу відстежувати стан виконання усіх робіт проекту в реальному вигляді, на основі журналу подій, знизити можливості появлень ризиків його невиконання.

3 ДОСЛІДЖЕННЯ ОТРИМАНИХ НАУКОВИХ РЕЗУЛЬТАТІВ

3.1 Розробка удосконаленого методу контролю виконання робіт

Діаграма Ганта – це інструмент управління проектами, що ілюструє план проекту. Зазвичай вона складається із двох частин: у лівій частині наведено список завдань, а у правій — тимчасова шкала зі смугами, що зображають роботу. Діаграма Ганта також може включати дати початку та завершення завдань, контрольні точки, залежності між завданнями та виконавцями [8].

Діаграми Ганта дозволяють візуалізувати складові частини проекту та розбити його на менші завдання для зручності керування. Отримані завдання розміщуються вздовж часової шкали діаграми Ганта, після чого до неї додаються залежності між завданнями, виконавцями та контрольними точками.

Оскільки команди відстежують час виконання завдань у плані, ви можете відстежувати стан своїх проектів і за потреби вносити зміни. Діаграма Ганта може включати дати випуску, віхи чи інші важливі показники для відстеження прогресу проекту.

Є дві основні причини, чому діаграми Ганта настільки популярні в управлінні проектами. Вони дозволяють легко створювати складні плани, особливо ті, які включають кілька команд і змінюють часові рамки. Діаграми Ганта допомагають командам планувати терміни та правильно розподіляти ресурси.

Планувальники проекту також використовують діаграми Ганта, щоб отримати уявлення про проект. Вони також зображують взаємозв'язок між датами початку та закінчення завдань, контрольними точками та залежними завданнями.

Планування проекту включає багато взаємопов'язаних ітерацій, що призводять до єдиного генерального плану. Під планом проекту далі будемо розуміти систему планових заходів, задокументованих в результаті складання. Ця система складається з параметрів, пов'язаних особливим чином,

забезпечуючи вирішення окремої проблеми розробки. Ці параметри, рисунок 3.1, формуються на основі ряду функціональних напрямків проектної діяльності



Рисунок 3.1 – Функціональні напрямки проектної діяльності

Використаємо техніку PERT, за допомогою якої можна спрогнозувати оптимістичний і песимістичний час виконання завдання, розрахувати очікувану тривалість операції. Метод PERT показаний у формулі 3.1.

$$E = \frac{(O+4M+P)}{6}, \quad (3.1)$$

де E – очікувана тривалість операцій;

O – оптимістична оцінка тривалості;

M – найбільш вірогідна оцінка тривалості;

P – пісимістична оцінка тривалості.

Це рівняння є не що інше, як середньозважене значення, де найбільш ймовірна оцінка тривалості має вагу в 4 рази більшу, ніж оптимістичні та песимістичні оцінки. Такий підхід запобігає надмірному перекосу в одному напрямку.

Метод PERT - Техніка Оцінки та Аналізу Програм та проектів, часто використовується при управлінні проектами та проведенні аналізу виробничих процесів.

Сама ідея методу вкрай проста — щоб оцінити час виконання завдання чи процесу, вам необхідно знати оптимістичну, песимістичну та найімовірнішу оцінку тривалості цього завдання.

Для даного проекту спрогнозували оптимістичний і песимістичний час виконання завдань проекту, обчислення наведені в формулі 3.2.

$$E = (23 + 62*4 + 40)/6 = 52 \quad (3.2)$$

Дані отримані з обчислення приблизно сходяться з часом на виконання проекту. Якщо протягом виконання проекту будуть виникати затримки часу на виконання роботи їх потрібно буде урахувати.

Також необхідно обчислити стандартне відхилення, для цього використовується формула (3.3):

$$STD = (E - O) / 6, \quad (3.3)$$

де STD – стандартне відхилення;

E – очікувана тривалість операцій;

O – оптимістична оцінка тривалості;

Для даного проекту обчислити стандартне відхилення, обчислення наведені у формулі 3.4

$$STD = (52 - 23)/6 = 5 \quad (3.4)$$

Для реалізації проекту по розробці конвертору у якості програмного додатку була зібрана команда, що складається із наступних спеціалістів, рисунок 3.2:



Рисунок 3.2 – Команда розробки додатку

Команда повинна виконати наступні роботи:

- 1) скласти правильне технічне завдання. Початково якісне ТЗ встановлює регламент, що допомагає вести роботу у заданому напрямі без самодіяльності з обох сторін. Чітко складений план дій нівелює різночитання рішень і допомагає уникнути безглузвих помилок на півдорозі;
- 2) провести дослідження збору лог файлів з ресурсу GitHub. Потрібно зрозуміти які методи збору лог файлів надає система розподільного контролю версій проектів Git;
- 3) розробити алгоритм перетворення лог файлів на таблиці. Оскільки програми обробки журналів подій не можуть приймати лог файли в первинному вигляді потрібно їх привести до виду структурованої таблиці;
- 4) розробити програмне забезпечення. Програмне забезпечення для форматування первинних лог файлів буде створене за допомогою мови програмування Javascript;
- 5) протестувати додаток.

На рисунку 3.1 наведено робочий простір відображення Діаграма Ганта.

Задача	Початок роботи	Кінець роботи	Тривалість роботи	Затримка
Складання ТЗ	10.04.2021	12.04.2021	3	0
Дослідження збору лог файлів	12.04.2021	13.04.2021	2	0
Розробка алгоритму роботи	13.04.2021	15.04.2021	3	0
Розробка програмного забезпечення	15.04.2021	25.04.2021	11	0
Тестування	25.04.2021	27.04.2021	3	0

Рисунок 3.3 - План проекту

На рисунку 3.2 Зображена діаграма Ганта яка відповідає всім роботам які повинні бути виконані за планом проекту.

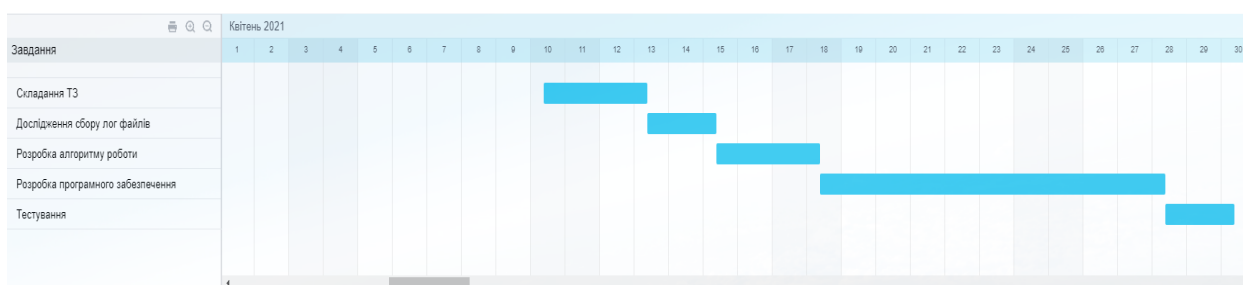


Рисунок 3.4 - Діаграма Ганта

JavaScript — це мова програмування, прототипно орієнтована. Вона відображає мову ECMAScript, прототип якого був спочатку. Перша варіація з'явилася ще в 1995 році і з тих пір постійно вдосконалювалася, поки не прийшла до нинішнього вигляду.

Частіше ця мова використовується при розробці додатків і браузерів, щоб надати їм інтерактивності.

JavaScript не призначено для створення автономних програм. Програма JavaScript вбудовується безпосередньо у вихідний текст HTML-документа і інтерпретується браузером у міру завантаження цього документа. За допомогою JavaScript можна динамічно змінювати текст завантажуваного

HTML-документа та реагувати на події, пов'язані з діями відвідувача або змінами стану документа чи вікна.

Важлива особливість JavaScript – об'єктна орієнтованість. Програмістові доступні численні об'єкти, такі як документи, гіперпосилання, форми, кадри і т.д. Об'єкти характеризуються описовою інформацією (властивостями) та можливими діями (методами).

Основною особливістю цієї мови є те, що на неї вплинули інші (Python, Java тощо) мови програмування, щоб зробити JavaScript максимально зручним і зрозумілим тим користувачам, які не мають відповідної освіти та глибокої знання - не програмісти. JavaScript є зареєстрованою торговою маркою Oracle.

Серед переваг мови програмування JavaScript виділяють:

- жоден сучасний браузер не обходиться без підтримки JavaScript;
- з використанням написаних на JavaScript плагінів та скриптів упорається навіть не фахівець;
- корисні функціональні налаштування;
- мова, що постійно вдосконалюється - зараз розробляється бета-варіація проекту, JavaScript;
- взаємодія з програмою може здійснюватися навіть через текстові редактори – Microsoft Office та Open Office;
- перспектива використання мови у процесі навчання програмування та інформатики.

Недоліками мови JavaScript вважається:

- знижений рівень безпеки через повсюдний та вільний доступ до вихідних кодів популярних скриптів;
- безліч дрібних дратівливих помилок кожному етапі роботи. Більшість їх легко виправляється, та їх наявність дозволяє вважати цю мову менш професійним, порівняно коїться з іншими;
- повсюдне поширення. Своєрідним недоліком можна вважати той факт, що частина програм, що активно використовуються (особливо додатків)

перестануть існувати за відсутності мови, оскільки цілком базуються на ньому;

На рисунку 3.5 зображено фрагмент коду який форматує отриманий файл логу подій до упорядкованої таблиці формату csv.

```
function ConvertToCSV(objArray) {
    var array = typeof objArray != 'object' ? JSON.parse(objArray) : objArray;
    console.log(array[0]);
    var str = '';

    for (var i = 0; i < array.length; i++) {
        var line = '';
        for (var index in array[i]) {
            if (line != '') line += ',';

            line += array[i][index];
        }

        str += line + '\r\n';
    }

    return str;
}

// Example
$(document).ready(function () {

    // Create Object
    var items = JSON.parse(data);

    $('#convert').on('click', function () {

        CSV = ConvertToCSV(items);
        // $('#csv').append(ConvertToCSV(items));
        var uri = 'data:text/csv;charset=utf-8,' + escape(CSV);

        var link = document.createElement("a");
        link.href = uri;
        // link.style = "visibility:hidden";
        link.download = 'Hello' + ".csv";
        // link.text = 'Download';
        console.log(link);

        $('body').append(link);
        // link.click();
        // document.body.removeChild(link);
        $('a').append('Download');
    });
});
```

Рисунок 3.5 – Приклад програмного коду

4 ВИКОРИСТАННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ НА ПРАКТИЦІ

4.1 Структура журналів реєстрації подій

Дані про події, або логи повинні бути добре описані та структуровані, але, нажаль, найчастіше ці дані не мають ніякої структури, їх використовують розробники для відстеження помилок тому коригують їх під свій лад.

Логи можна створювати з емейлів, веб-сторінок, документів. Щоб мати можливість їх обробляти, їх потрібно отримати та інтерпретувати у відповідний до журналу подій формат.

Процес розробки процесів потребує якісного журналу поді, адже якість результату технологічного процесу залежить від даних який він отримує. Існує кілька важливих критеріїв оцінювання якості даних про події. Перш за все, події мають бути достовірними, тобто записані події дійсно відбулися і що їх атрибути правильні.

По-друге, будь-які зареєстровані події мають бути чітко визначені семантично [14]. На жаль, зазвичай журнали логів є допоміжним продуктом, який використовують розробники для виправлення дефектів.

В будь яку операційну систему вбудована можливість записувати події, але ця інформація не використовується для аналізу процесу чи інших цілей користувачами, а більшість з них навіть блокує досуп розробникам до цих лог файлів [1]. Це може бути пов'язано з відсутністю бажання та знань в цій області, хоча з логів можна дізнатися всі подробиці роботи системи. Як наслідок, організаціям досить важко отримати повну інформацію про процеси, що відбуваються. Типовими розширенням файлів для зберігання журналів подій є формати XES і MXML.

До недавнього часу стандартом для зберігання та обміну даними між журналами подій був MXML, який був створений у 2003 році і був адаптований інструментом для процесного аналізу – системою Disco. Формат XES заснований на XML і є наступником формату MXML, але на відміну від формату MXML, XES є менш обмеженим і дійсно розширюваним форматом.

На даний момент формат XES є стандартним форматом для зберігання журналів подій, обраним організацією IEEE Task Force on Process Mining (ця організація була створена IEEE з метою сприяння дослідженням, розробкам, розумінням науки про процес видобутку). Для того, щоб отримати дані необхідного формату з інших джерел даних, використовуються різні інструменти. Після визначення відповідних даних ви можете почати безпосередньо витягувати дані та працювати з ними.

На малюнку 4.1 наведено приклад фрагмента типового журналу подій, де кожен рядок таблиці відповідає конкретній події. Таблиця містить події, які стосуються процесу обробки запитів на компенсаційні виплати.

Журнал подій містить дані, що стосуються конкретного окремого процесу, тобто, щоб почати роботу з даними, потрібно переконатися, що всі події в журналі стосуються одного процесу.

Кожна подія в журналі представлена як дія. Журнал подій (рисунок 4.1) містить такі події – «Зареєструвати запит», «Перевірити квиток», «Відхилити запит», «Прийняти рішення» та інші.

Події повинні бути відсортовані, щоб мати можливість досліджувати залежності в моделі процесу [5]. Таблиця також містить додаткову інформацію для кожної події, наприклад, усі події мають позначки часу, ресурси та вартість. Позначки часу містять інформацію про дату та час події.

Ця інформація може виявитися корисною при аналізі характеристик деяких властивостей (наприклад, час очікування між двома діями). Ресурси представляють собою людей, які виконували то чи інше дію. В контексті видобутку процесів ці властивості називаються атрибутами.

id	commit id	author	date	message
Case 1	395030f220a837a9ae320c33a2a0bb8e01605b84	CrexysQ	2020-02-18	minor-fix
Case 2	7bc36b7e1e0948e78ef27f389a4daa43edec65aa	CrexysQ	2019-11-25	Merge-branch-master-of-https-github.com-CrexysQ-ToDoList1
Case 3	1536561810d0cc144a36a0cce53d13e3dbbc1bc0	CrexysQ	2019-11-25	Deleted-set-theme-service
Case 4	42d60a60007a8f5af4d179001079850ea6958f6	Arsenii Chystilin	2019-11-22	Changed-theme
Case 5	907858e9d072e80d5f5b89556627c9cf69c3722	CrexysQ	2019-11-21	Added-material-chnged-signUp-and-signIn-pages
Case 6	dfc52c35f58d10ecd7bf196b28028b1975f3d1f2	CrexysQ	2019-11-21	Updating-packages
Case 7	e23e70b6a52deba6d13d7764764653d2e6b281e3	CrexysQ	2019-07-23	deleted-post-install
Case 8	12e5807204c5c619b3d50b42f71d3152835e2648	CrexysQ	2019-07-23	Merge-branch-master-of-https-github.com-CrexysQ-ToDoList1
Case 9	5202fd32c00a080814ffc365e7744fa4d298e3a5	CrexysQ	2019-07-22	fix-for-heroku
Case 10	54a2be42c4a73a3d5e111e986944e4fb7b8296a9	CrexysQ	2019-07-22	asd
Case 11	6cf1a01fedf4f4667a518c967ade7b34f43f4ab5	CrexysQ	2019-07-22	Commented-favicon
Case 12	1a3305b23f817563d56675f62b10123be6ef1642	CrexysQ	2019-07-22	Added-fontawesome-partially-added-new-design
Case 13	296bd1b669bb3ea50ffb12df969b7e26b5ff18b9	CrexysQ	2019-07-17	Added-datepicker-and-calendar-partially-redizined
Case 14	cfdf10e97ad43e893ec498305c3c06f144ea49ce	CrexysQ	2019-07-09	Partly-changed-view
Case 15	6b6be9d33cbfdb72b7bb0b00b9519a18f8cd463d	CrexysQ	2019-06-25	added-timers-for-users
Case 16	f63ac2a1024495cdc378d95b43d29bacb30c003c	CrexysQ	2019-06-25	added-theme-for-user
Case 17	a0dfe82a596cdad504f5338b6d2568514daf336d	CrexysQ	2019-06-24	Work-with-theme-service
Case 18	b4e0026c7c986d67290cc8fe4fa8558728570065	CrexysQ	2019-06-21	
Case 19	846e2162b81b5241fe45aa3d14b7dc65bb1d8de4	CrexysQ	2019-06-20	Added-users
Case 20	65f38950d90dc4cfb8934913684abcd56fbbd2f	CrexysQ	2019-06-20	Little-fix-for-guard
Case 21	2e8088c23f2a27526a415c61965d22230c33cfe8	CrexysQ	2019-06-20	a
Case 22	cec064d281341b4d44843dd7c994ea54364e7650	CrexysQ	2019-06-20	Added-project-files
Case 23	1803f86d67cede747ef92989c1632d30bbad7b77	Arsenii Chystilin	2019-05-14	initial-commit
Case 24	a0dfe82a596cdad504f5338b6d2568514daf336d	CrexysQ	2019-06-24	Work-with-theme-service
Case 24	a0dfe82a596cdad504f5338b6d2568514daf336d	CrexysQ	2019-06-24	Work-with-theme-service
Case 25	a0dfe82a596cdad504f5338b6d2568514daf336d	CrexysQ	2019-06-24	Work-with-theme-service
Case 26	a0dfe82a596cdad504f5338b6d2568514daf336d	CrexysQ	2019-06-24	Work-with-theme-service
Case 27	a0dfe82a596cdad504f5338b6d2568514daf336d	CrexysQ	2019-06-24	Work-with-theme-service
Case 28	7bc36b7e1e0948e78ef27f389a4daa43edec65aa	CrexysQ	2019-11-25	Merge-branch-master-of-https-github.com-CrexysQ-ToDoList1
Case 29	7bc36b7e1e0948e78ef27f389a4daa43edec65aa	CrexysQ	2019-11-25	Merge-branch-master-of-https-github.com-CrexysQ-ToDoList1

Рисунок 4.1 – Приклад журналу подій

JSON – це текстовий формат спілкування, заснований на JavaScript. Але формат не залежить від JS і може використовуватися будь-якою мовою програмування.

Значення які можуть бути використані у файлі JSON зображені на рисунку 4.2.

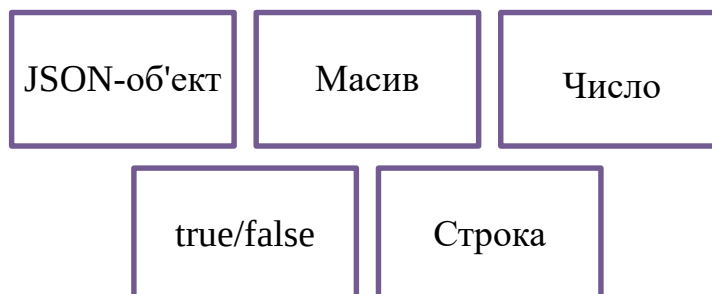


Рисунок 4.2 – Формати даних які можна використовувати у JSON

JSON об'єкт – це неупорядкований набір пар ключ/значення. Об'єкт починається відкриваючою фігурною дужкою і закінчується закриваючою фігурною дужкою. Після кожного імені ставиться двокрапка, пари ключ/значення розділяються комою.

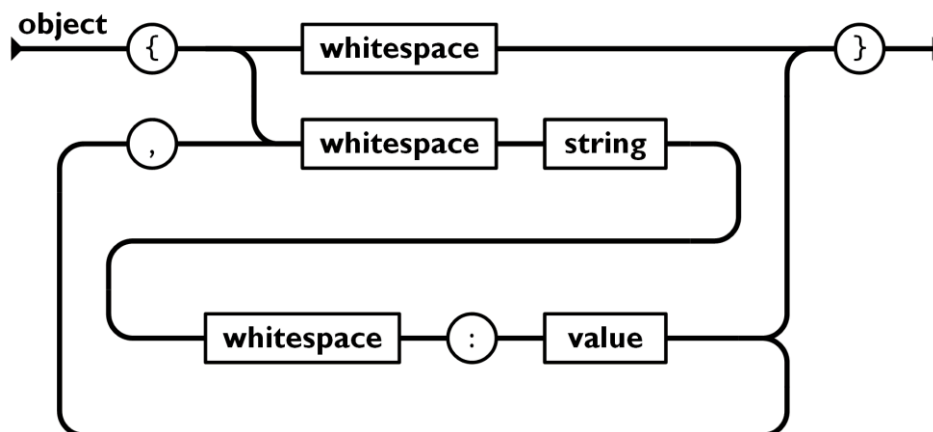


Рисунок 4.3 – Схема структури JSON об'єкту

Масив — це впорядкована колекція значень. Масив починається з відкриваючої квадратної скобки і закінчується закриваючою. Значення відокремлюються комами.

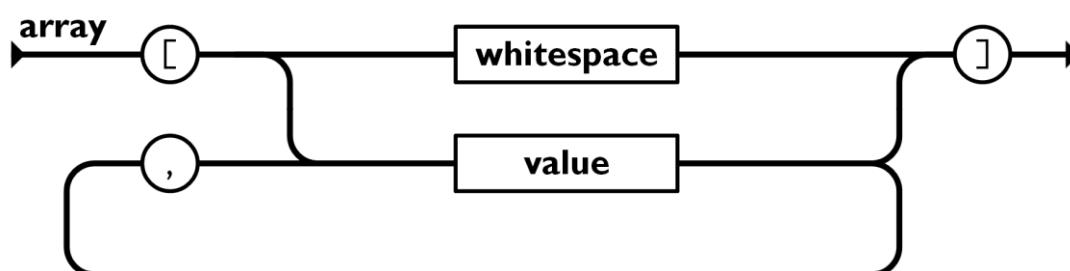


Рисунок 4.4 – Схема структури Масиву

Незважаючи на свою назву, JSON можна використовувати не тільки в JavaScript, але і в будь-якій іншій мові програмування.

JSON також має значну перевагу перед іншими форматами. На відміну від них, він більш компактний, що дуже важливо при обміні даними в Інтернеті. Крім того, JSON легший у використанні, його набагато легше читати та писати.

У веб-розробці JSON дуже часто використовується як формат для передачі інформації від веб-сервера до клієнта (веб-браузер) для запиту AJAX.

Формат представлення даних JSON має наступні переваги:

- зручні та швидкі методи, призначені для перетворення (розбору) рядка JSON в об'єкт JavaScript і назад;
- чітка і проста структура даних;
- дуже малий розмір порівняно з іншими форматами. Це пов'язано з тим, що формат JSON містить мінімально можливе форматування, тобто. при його написанні використовується лише кілька спеціальних символів. Це дуже важлива перевага, оскільки дані, представлені у форматі JSON, будуть завантажуватися швидше, ніж якби вони були представлені в інших форматах.

Завдяки тому, що цей формат має багато переваг, він використовується не тільки в JavaScript, а й у багатьох інших мовах, таких як C, Ruby, Perl, Python, PHP тощо.

На малюнку 4.5 показано приклад документа JSON, який зберігає інформацію з логів проекту на GitHub . Процес складається з однієї послідовності подій, яка в свою чергу складається з чотирьох подій.

```
[
  {
    "commit": "395030f220a837a9ae320c33a2a0bb8e01605b84",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2020-02-18",
    "message": "minor-fix"
  },
  {
    "commit": "7bc36b7e1e0948e78ef27f389a4daa43edec65aa",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2019-11-25",
    "message": "Merge-branch-master-of-https-github.com-CrexysQ-ToDoList1"
  },
  {
    "commit": "1536561810d0cc144a36a0cce53d13e3dbbc1bc0",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2019-11-25",
    "message": "Deleted-set-theme-service"
  },
  {
    "commit": "42d60a6000f7a8f5af4d179001079850ea6958f6",
    "author": "anuitexdev <anuitexdev@anuitex.com>",
    "date": "2019-11-22",
    "message": "Changed-theme"
  },
  {
    "commit": "907858e9d072e80d5f55b89556627c9cf69c3722",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2019-11-21",
    "message": "Added-material-chnged-signUp-and-signIn-pages"
  },
  {
    "commit": "dfc52c35f58d10ecd7bf196b28028b1975f3d1f2",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2019-11-21",
    "message": "Updating-packages"
  },
  {
    "commit": "e23e70b6a52deba6d13d7764764653d2e6b281e3",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2019-07-23",
    "message": "deleted-post-install"
  },
  {
    "commit": "12e5807204c5c619b3d50b42f71d3152835e2648",
    "author": "CrexysQ <themithrossfilms@gmail.com>",
    "date": "2019-07-23",
    "message": "Merge-branch-master-of-https-github.com-CrexysQ-ToDoList1"
  }
]
```

Рисунок 4.5 – Журнал событий в формате JSON

4.2 Опис програмних засобів для побудови моделі на основі аналізу логів

На рисунку 4.6 наведено детальний огляд архітектури програмного забезпечення Fluxicon Disco, де показано взаємозв'язок між журналами подій, фреймворком та плагінами.

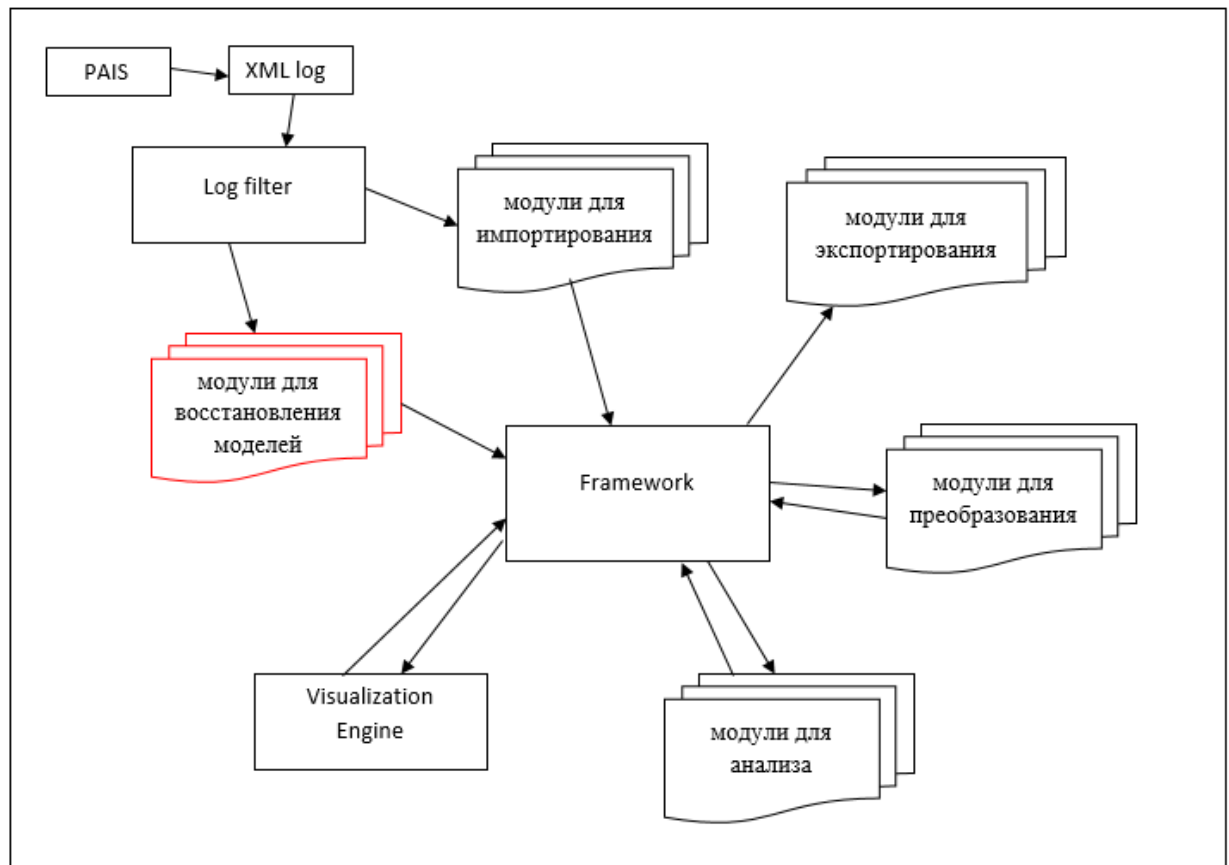


Рисунок 4.6 – Удосконалена структура платформи Disco

Як наведено на вище, журнали подій залежні від процесу інформаційної системи. Для зчитування логів журналу подій Fluxicon Disco фільтрує журнал. Також платформа Disco дозволяє включати в себе декілька різних типів функцій:

Імпорт - дозволяє під час виконання використовувати раніше створені моделі;

Експорт - дозволяє зберігати створені моделі, а також виконувати операції над журналами подій;

Відновлення моделі - реалізація алгоритму зчитування інформації логів журналу подій і представлення її у вигляді різноманітних графічних моделей;

Fluxicon Disco це набір програмних засобів, які потрібні для роботи основних функцій Fluxicon Disco, а також для отримання і аналізу інформації про процеси розробки від журналу подій.

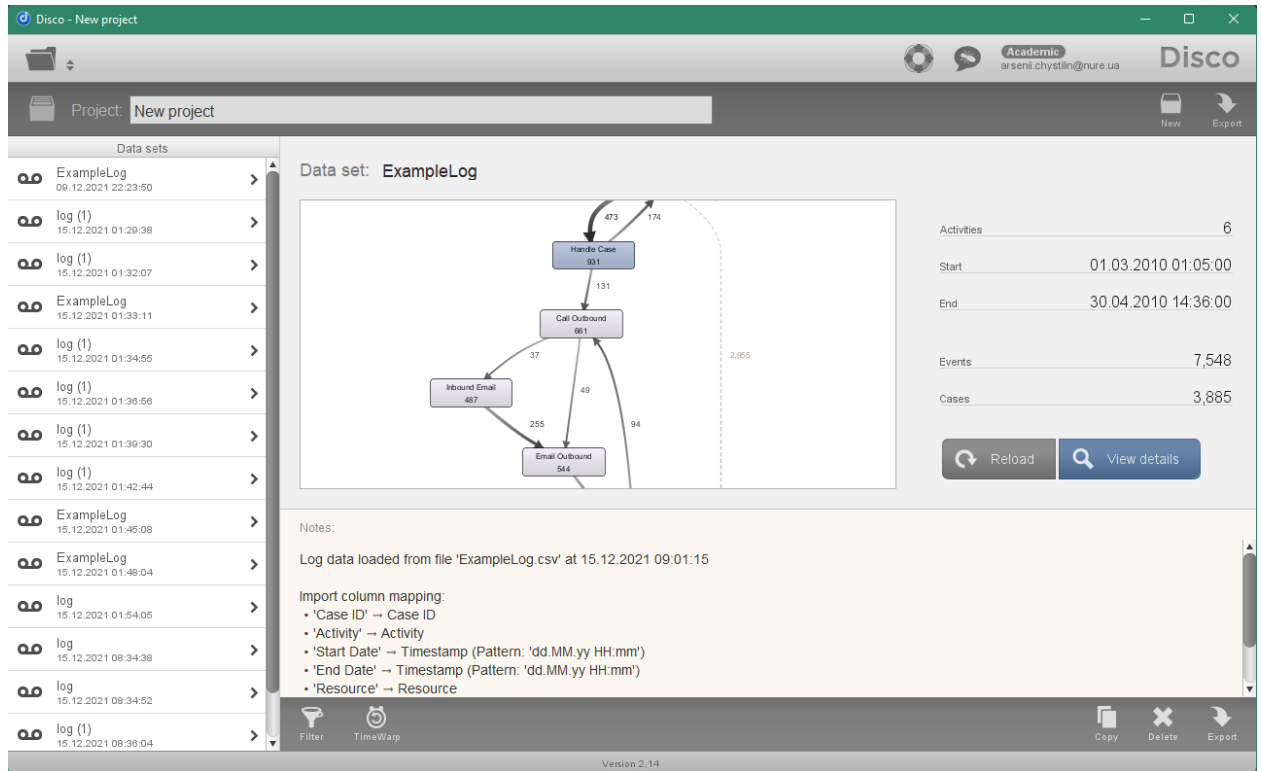


Рисунок 4.7 – інтерфейс програмної платформи Disco

В результаті завантаження журналу подій у Disco та подальшої його обробки обраним методом можна отримати графічну модель бізнес-процесів.

4.3 Експериментальна перевірка запропонованих підходів

Після того як запустилась програмна платформа Disco в меню початкового вибору за допомогою клавіші імпорту необхідно загрузити журнал подій (рисунок 4.8).

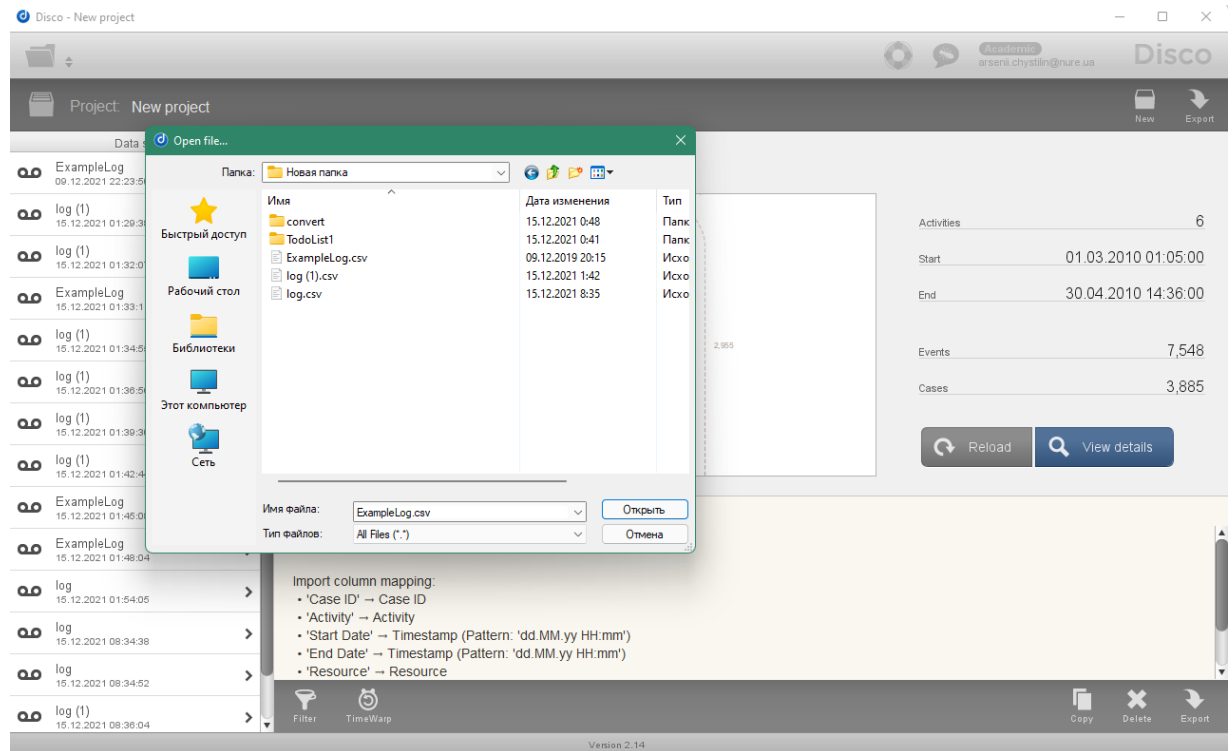


Рисунок 4.8 – Вибір файлу журналу подій

Після того як журнал подій було імпортовано, для створення графічної моделі необхідно обрати метод обробки журналу подій, як показано на малюнку 4.9.

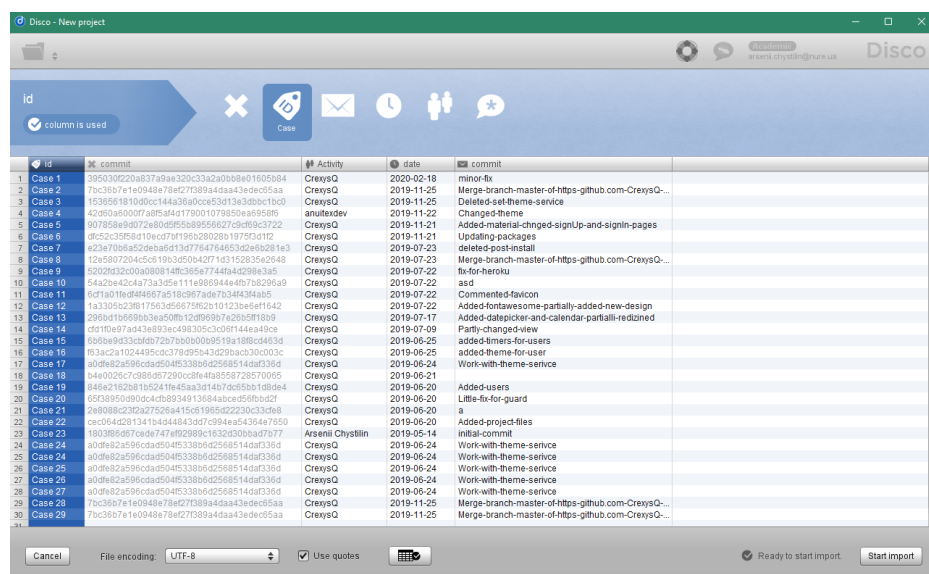


Рисунок 4.9 – Інтерфейс способу обробки журналу подій

На екрані підготовки журналу подій (рисунок 4.9) до імпорту колонкам журналу подій можна присвоїти такі лейбли як Case, Activity, Timestamp, Resource, Remove, та також є можливість задати власний лейбл. Також можна вибрати тип кодування файлу.

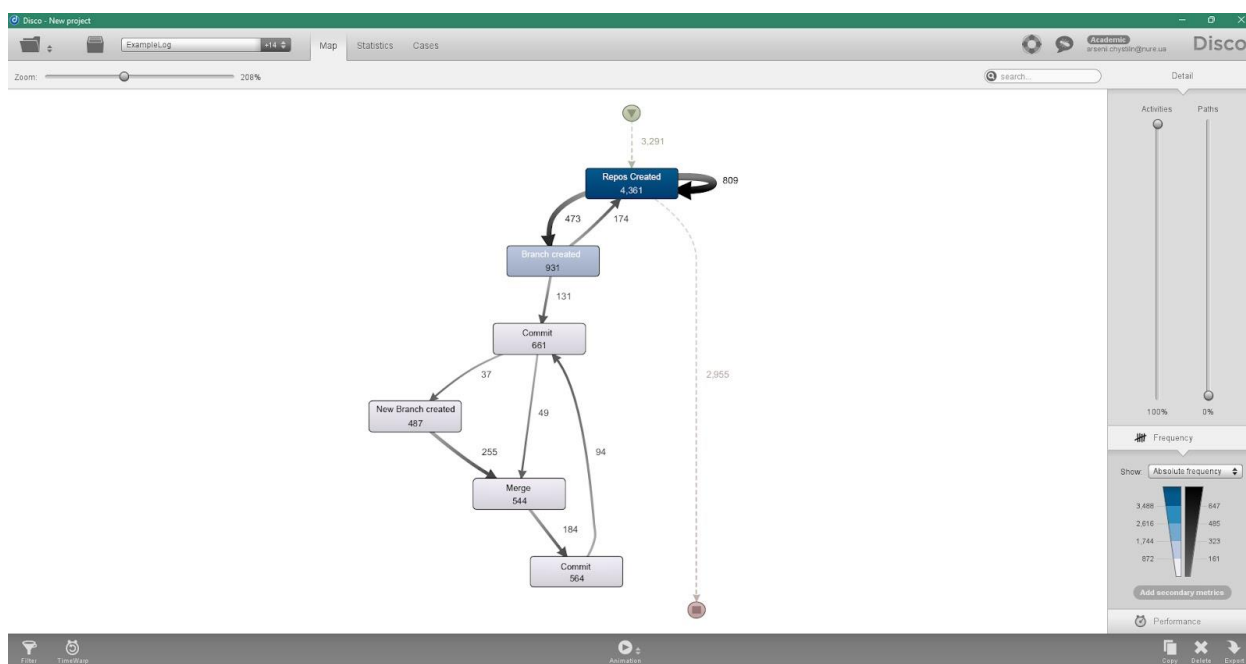


Рисунок 4.10 – Створений граф

На екрані обзору створеного графу (рисунок 4.10) можна побачити сам граф який програма Fluxicon Disco створила на основі завантаженого журналу подій. Окрім самого графу на цьому екрані відфільтрувати події, вибрати кількість відображаних подій у процентному відношенні. Також є можливість запустити анімацію процесу.

Екран обзору статистики журналу подій можна побачити на рисунку 4.11

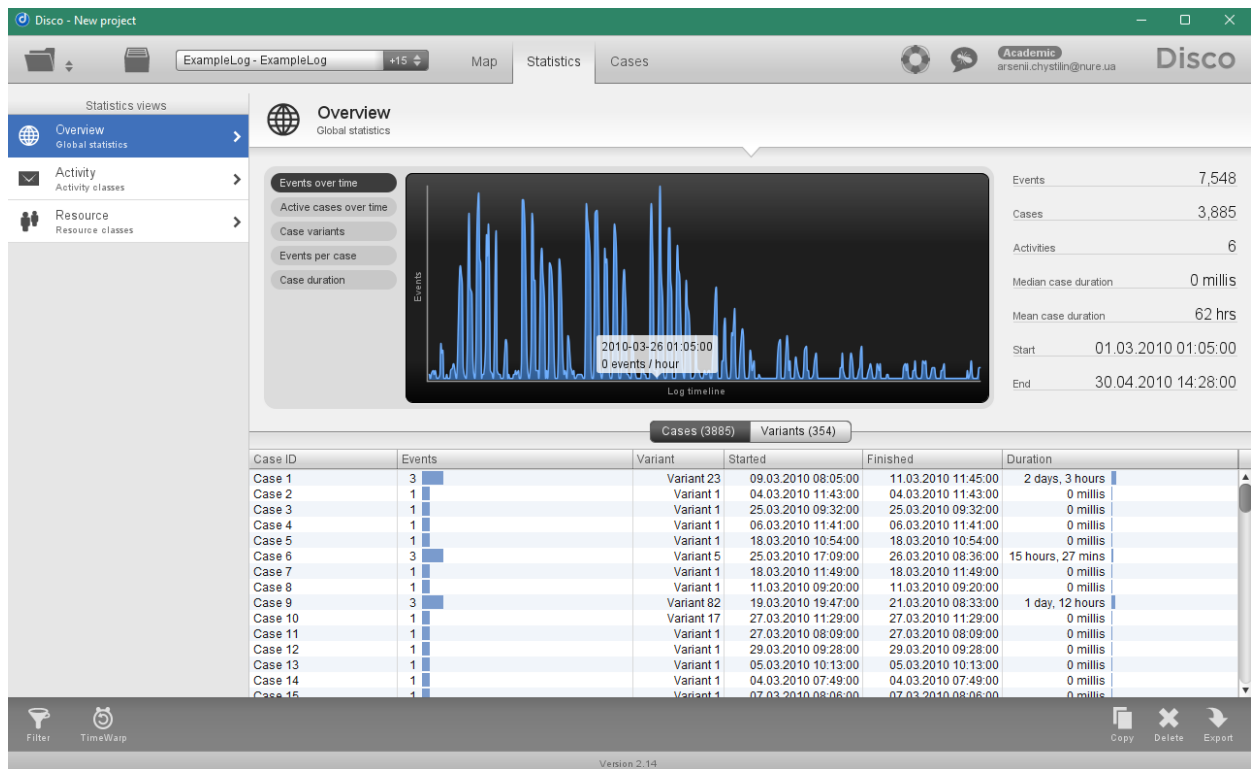


Рисунок 4.11 – Экран статистики журналу подій.

На рисунку 4.11 можна подивитися графік на якому зображено кількість подій за плином часу, а також продивитися кількість подій у кейсах. Це може допомогти проаналізувати в котрій годині дня над розробники активніше працюють над проектом

ВИСНОВКИ

За результатами магістерської кваліфікаційної роботи було проведено дослідження гнучких технологій розробки ІТ проекту та методів контролю виконання робіт ІТ проекту. Отримані результати дозволяють свідчити про актуальність використання досліджуваних технологій розробки ІТ проектів та методів їх контролю.

Проведено Аналіз методів та технології Process Mining. За результатами цього дослідження встановлено основні недоліки існуючих методів технології Process Mining при її використанні під час розробки ІТ-проектів. За результатами досліджень визначено мету та основні задачі кваліфікаційної роботи.

Розроблено удосконалений метод контролю виконання робіт ІТ проекту на основі структуризації логу, що відображає послідовність виконання робіт із часовими мітками.

Зроблено експериментальну перевірку удосконаленого методу контролю виконання робіт ІТ проекту. Наведено приклад роботи вдосконаленого методу контролю виконання робіт у ІТ-проектах.

Метод процесної аналітики дає змогу розвивати побудову дискретної моделі процесу, яка була представленою альфа-алгоритмом інтелектуального аналізу процесів. На відміну від інших існуючих підходів, ознаки послідовного і паралельного виконання визначаються не для окремих подій, а для інтервалів подій, що дає можливість відображати в моделі не стани, а дії процесу з урахуванням тривалості процесу.

За результатами можна зробити висновок що, поставлена мета дослідження методів планування та обліку була успішно виконана, і доведена наукова новизна.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської кваліфікаційної роботи за спеціальністю 122 – „Комп'ютерні науки” програма «Управління проектами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с.
2. Варфоломеева, А.О. Информационные системы предприятия: Учебное пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. - М.: НИЦ ИНФРА-М, 2013. - 283 с.
3. Мартин Р. Чистый код: создание, анализ и рефакторинг. СПб: Питер, 2010. – 464 с.
4. White A. Major JavaScript Engines. JavaScript Programmer's Reference. 8-е 2nd Edition. Packt Publishing, 2017. – 246 с.
5. Демінг, WE, 1981. Управління статистичними методами для забезпечення якості та продуктивності. Нью-Йоркський університет, «Вища школа бізнесу», Нью-Йорк.
6. Van der Aalst W.M.P., Adriansyah A., Van Dongen B.F. Replaying history on process models for conformance checking and performance analysis // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Vol. 2(2). Wiley Online Library. 2012. P. 182–192.
7. Van der Aalst W.M.P., Weijters A.J.M.M., Maruster L. Workflow Mining: Discovering Process Models from Event Logs // IEEE Transactions on Knowledge and Data Engineering, 2004. Vol. 16(9). P. 1128–1142.
8. A.J.M.M. Weijters and J.T.S. Ribeiro. Flexible Heuristics Miner (FHM). BETA Working Paper Series, WP 334, Eindhoven University of Technology, Eindhoven, 2010.

9. Маклаков С.В. Моделирование бизнес-процессов с BPwin. - М.: ДИАЛОГ-МИФИ, 2002. - 224 с.
10. W.M.P. van der Aalst, A. Adriansyah, A.K.A. de Medeiros. The Process Mining Manifesto. - F.Daniel, K.Barkaoui, S.Dustdar (Eds.): BPM 2011 Workshops, Part I, Springer-Verlaq. - 2012. - pp.169-194.
11. Барсегян А.А., Куприянов М.С., Холод И.И., Тесс М.Д., Елизаров С.Иб. Анализ данных и процессов 3-е изд. Санкт-Петербург: БХВ-Петербург, 2009.
12. A. K. Alves de Medeiros, B. F. van Dongen, W.M.P. van der Aalst, and A.J.M.M. Weijters. Process Mining for Ubiquitous Mobile Systems: An Overview and a Concrete Algorithm. In L. Baresi, S. Dustdar, H. Gall, and M. Matera, editors, Ubiquitous Mobile Information and Collaboration Systems (UMICS 2004), volume 3272 of Lecture Notes in Computer Science, pages 154—168. Springer-Verlag, Berlin, 2004.
13. A. K. Alves de Medeiros, B. F. van Dongen, W.M.P. van der Aalst, and A.J.M.M. Weijters. Process Mining: Extending the α -algorithm to Mine Short Loops. BETA Working Paper Series, WP 113, Eindhoven University of Technology, Eindhoven, 2004.
14. B. F. van Dongen and W.M.P. van der Aalst. EMiT: A Process Mining Tool. In Jordi Cortadella and Wolfgang Reisig, editors, ICATPN, volume 3099 of Lecture Notes in Computer Science, pages 454—463. Springer, 2004.A
15. Guzzo A. Pontieri L. Sacca D. Greco, G. Mining expressive process models by clustering workflow traces. In Dai, H., Srikant, R., Zhang, C., eds.: PAKDD., Springer, 3056:52-62, 2004.
16. B.F. van Dongen, A.K.A. de Medeiros, and L.Wenn. ProcessMining: Overview and Outlook of Petri Net Discovery Algorithms. In K. Jensen and W.M.P. van der Aalst, editors, Transactions on Petri Nets and Other Models of Concurrency II, volume 5460 of Lecture Notes in Computer Science, pages 225—242. Springer, Berlin, 2009.