НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

«На правах рукопису»
УДК 519.68; 005.334; 007.51/.52

«До захисту допущено»
Завідувач кафедри ММСА
О.Л. Тимощук
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 122 Комп'ютерні науки

на тему: «Аналіз ризиків проекту за допомогою текстового інтелектуального аналізу даних коментарів в системі управління проектами jira**»**

Виконав (-ла):
студент (-ка) ІІ курсу, групи КА-74м
Лєднікова Анна Андріївна _____

Керівник:
д.т.н., професор
Бідюк П.І _____

Рецензент:
Завідувач кафедри АУТС НТУУ "КПІ ім. І. Сікорського"
д.т.н., професор С.Ф. Теленик _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ
2019

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
«Інститут прикладного системного аналізу»
Кафедра математичних методів системного аналізу

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 122 «Комп'ютерні науки» («Інтелектуальний аналіз даних в управлінні проектами»)

ЗАТВЕРДЖУЮ
Завідувач кафедри ММСА
О.Л. Тимощук

«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Лєдніковій Анні Андріївні

1. Тема дисертації «Аналіз ризиків проекту за допомогою текстового інтелектуального аналізу даних коментарів в системі управління проектами jira», науковий керівник дисертації Бідюк Петро Іванович, д.т.н., профессор, затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом дисертації

3. Об'єкт дослідження *проектні ризики*

4. Предмет дослідження *методи аналізу проектних ризиків і коментарів*

5. Перелік завдань, які потрібно розробити:
1) дослідження питання ризиків проектів сфери IT та методів їх виявлення;

2) дослідження існуючих методів та алгоритмів для інтелектуального аналізу тексту на предмет тригерів ризиків;

3) розробка методології використання інтелектуального аналізу тексту для ідентифікації та аналізу ризиків проекту;

4) розробка ПЗ для проведення експериментів за даною методологією;

5) аналіз результатів та рекомендації щодо подальших досліджень;

6) розробити план для створення стартап-проекту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: (1) – постановка задачі дослідження; (2) – огляд використаних методів моделювання і прогнозування; (3) – критерії для аналізу адекватності моделей та їх якості ; (4) – результати моделювання.

7. Орієнтовний перелік публікацій:

8. Дата видачі завдання 1 *жовтня 2017*

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Пошук літератури та поверхневе дослідження за темою	12.03.2018 – 30.04.2018	Виконано
2	Огляд сучасних методів інтелектуального аналізу текстових даних на предмет емоцій	01.05.2018 – 30.05.2018	Виконано
3	Огляд сучасних методів ідентифікації та аналізу ризиків	01.06.2018 – 30.07.2018	Виконано
4	Розроблення та опис методології	01.09.2018 – 20.09.2018	Виконано
5	Пошук та попередня підготовка даних для виконання обчислювальних експериментів	22.09.2018 – 10.10.2018	Виконано
6	Виконання обчислювальних експериментів з метою аналізу проекту за пропонованим методом. Аналіз адекватності отриманих результатів.	11.10.2018 – 14.12.2018	Виконано
7	Аналіз можливостей подальшого покращення ранжування задач та генерування назви	15.12.2018 – 30.12.2018	Виконано
8	Розробка плану запуску стартап-проекту	10.01.2019 – 28.02.2019	Виконано

Студент	А.А. Лєднікова
Науковий керівник дисертації	П.І. Бідюк

РЕФЕРАТ

Магістерська дисертація: 125 с., 21 рис., 29 табл., 5 додатки, 24 джерела.

Об’єктом дослідження є проектні ризики. Предметом дослідження є методи аналізу проектних ризиків і коментарів в системі управління проектами jira.

Мета дослідження:

1) дослідження питання ризиків проектів сфери IT та методів їх виявлення;

2) дослідження існуючих методів та алгоритмів для інтелектуального аналізу тексту на предмет тригерів ризиків;

3) розробка методології використання інтелектуального аналізу тексту для ідентифікації та аналізу ризиків проекту;

4) розробка ПЗ для проведення експериментів за даною методологією;

5) аналіз результатів та рекомендації щодо подальших досліджень.

Теоретичною та методологічною основою дослідження є праці закордонних вчених в галузі управління проектів, управління ризиками проекту, інтелектуальної обробки текстових даних, сентиментального та емоційного аналізу текста, а також моделей для побудови тем та графічного представлення результатів.

В ході дипломної роботи було розроблено методологію та створено програмний продукт для визначення ризиків проекту, базуючись на комунікації розробників, а також представлено результати роботи програми на даних реального проекту CASSANDRA компанії Apache Software Foundation.

Методологія реалізована на основі вже відомих алгоритмів визначення емоційних складових у тексті VAD та матричних методів аналізу ризиків

проекту з використанням власних розробок, що дозволяти з’єднати ці різні підходи.

Програмний продукт реалізовано за допомогою мови програмування Python та пакетами для роботи з текстом gensim та spacy. У кінці роботи надано рекомендації до подальших досліджень.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТЕКСТОВИХ ДАНИХ, ВЕЛИКІ ДАНІ, АНАЛІЗ РИЗИКІВ ПРОЕКТУ, ЕМОЦІЙНИЙ АНАЛІЗ, ТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ХМАРИ СЛІВ.

ABSTRACT

Thesis work: 125 pp., 21 fig., 29 tabl., 5 applications, 24 sources.

The object of the research is project risks. It is planned to study the methodological calculations of project risks and comments on the project management system.

The aim of the study:

1) research into the risks of IT projects and detection and detection methods;

2) the study of existing methods and algorithms for the intellectual analysis of the text on the subject of risk triggers;

3) development of a text mining methodology for project identification and risk analysis;

4) software development for conducting experiments on this methodology;

5) analysis of the results and recommendations for further research.

The theoretical and methodological basis of the research is the work of scientists in the field of project management, project risk management, intellectual processing of textual data, and sentimental and emotional analysis of text.

In the course of the thesis, a methodology was developed and a software product was created for project risk assessment based on developer communications, as well as the results of the program's work on the data of the real project CASSANDRA of Apache Software Foundation were presented.

The methodology is implemented on the basis of the already known algorithms for determining the emotional components in the VAD text and the matrix methods of project risk analysis using our own developments, allowing to combine these different approaches.

The software product is implemented using the Python programming language and the framework for working with Apache Spark big data. Recommendations for further research are given.

TEXT DATA MINING, BIG DATA, PROJECT RISK ANALYSIS, EMOTIONAL ANALYSIS, TOPIC MODELING, WORD CLOUDS.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ВЗД (Валентність Збудження Домінування)

DM (Data Mining) - інтелектуальний аналіз даних

IT (Information technology) - інформаційні технології

LDA (Latent Dirichlet Allocation) - розподілу прихованих дирихле

PM (Project Management) - управління проектами

ВСТУП

Робота складається з п’ятьох розділів.

У першому розділі розглядається постановка задачі текстового аналізу коментарів для виявлення та оцінки ризиків проекту, проводиться огляд основних понять, моделей та підходів, що використовуються при вирішенні такого роду задач.

У другому розділу наведені: виявлення емоційних складових та процес їх перетворення у показники ризиків проекту, формалізація моделі отримання теми текстів, перетворення їх у назви ризиків та критеріїв оцінки отриманих результатів.

У третьому розділі розглянуто програмне забезпечення, що використовувалось, а також наведено код з поясненнями та прикладами, які відтворюють процес виявлення та перетворення емоційних складових коментарів у ризики. Також наведемо аналіз ризиків реального проекту CASSANDRA компанії Apache Software Foundation

Четвертий розділ містить аналіз результатів та маркетингове дослідження можливого продукту на основі запропонованого методу, надані рекомендації для подальших досліджень.

РОЗДІЛ 1 ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ КОМЕНТАРІВ ДЛЯ ОЦІНКИ РИЗИКІВ ПРОЕКТУ

1.1 Актуальність проблеми

Сфера інформаційних технологій (IT) розвивається дуже швидко та стрімко, так само як її представники. Компанії хутко ростуть та все більше стають децентралізованими. Але чим більше компанія та команда, тим більша виникає потреба у її менеджменті.

Найпопулярнішим інструментом для управління IT проектами на даний момент є Jira - це система для відстеження помилок і проблем, яка надає функції управління проектами для компаній і розробників програмного забезпечення. Зазвичай проекти приватні, але деякі компанії, які надають безкоштовні програмні продукти, мають відкриті репозиторії, завдяки яким кожен має можливість відстежувати стан проекту, створювати завдання, коментувати та допомагати у розробці. Прикладом такої компанії є Apache Software Foundation [16].

В управлінні проектами є декілька складових, одна з яких найбільш важлива та й той же самий час найбільш трудомістка - це аналіз ризиків, метою якого є забезпечення виконання задач проекту у певний час та при наявності певної кількості ресурсів. Даний тип управління повинен здійснюватися протягом всього часу існування проекту.

В контексті системи Jira дані коментарів та журналів задач є цінною інформацією для визначення поточного стану. Аналізуючи їх на предмет висловлюваних емоцій або певних ключових слів, можливо створити матрицю

ризиків для виявлення проблем, пов'язаних з людським фактором та комунікаціями. Побудова тематичних моделей дає змогу дати швидке та коротке представлення про сутність проблеми. Використання даного підходу на історичних даних допоможе отримати цінні уроки минулого для більш ефективній роботи у майбутньому.

Тож у цій роботі ми вивчаємо емоції, виражені в системі відстеження проблем Apache Jira. Зокрема, ми зацікавлені в їх перетворенні у ризики й надання інструментів для подальшої можливої розробки системи підтримки прийняття рішень для управління проектним ризиком, пов'язаних з людськими і соціальними факторами.

Дана тема актуальна, оскільки дозволяє створити інструменти для визначення та відслідковування ризиків проекту в автоматичному режимі за допомогою методів інтелектуального аналізу даних.

При розв’язанні цієї задачі можуть з'являтися такі проблемі:

- Складності з аналізом суто технічних коментарів вузької області знань
- Пошук найбільш адекватного перетворення значень емоційних складових текстів у показники ризику
- Визначення релевантної назви ризику
- Технічні обмеження для обчислень

Постановка задачі така: аналізуючи коментарі задач проекту на предмет наявності емоції визначити показники ризику невиконання даної задачі та виявити ключові слова для опису ризиків пов’язаних з даною задачею.

1.2 Управління проектами розробки програмного забезпечення

Перші асоціації з управлінням проектами - PMBOK і його дев'ять галузей знань, такі як управління часом, управління якістю, тощо. Це не рідкість, коли практики, описані в PMBOK, вважаються марними і ідеалістичними, неможливим до застосування у реальному світі. Причина цього полягає в збільшенні кількості невеликих і динамічних проектів, коли більшість процедур PMBOK не використовуються, що заощаджує час і не викликає великих втрат, як у будівництві або промисловості.

Одним з місць концентрації таких динамічних проектів є IT-сфера. Існують ще одна категорії методів управління проектами в розробці програмного забезпечення - Agile.

Agile Software Development (ASD) стає найпоширенішою технікою управління проектами: організації шукають способи бути більш гнучкими, тоді як 71% організацій вже повідомляють про використання цих підходів для своїх проектів. [1] Найбільш поширеними методами Agile є Scrum, XP і Kanban.

PMBOK, розроблений Інститутом управління проектами, структурований навколо п'яти груп процесів (ініціювання, планування, виконання, контролю та закриття) і дев'яти областей знань (управління інтеграцією, управління сферою, управління часом, управління витратами, управління якістю, управління людськими ресурсами, управління комунікаціями, управління ризиками, управління закупівлями). З іншого боку, гнучке управління проектами програмного забезпечення базується на наступних принципах: прийняти зміни, зосередження на споживчій цінні, поставка частин функціональності поступово, співпрацювання, безперервне відображення та навчання. [2]

Agile методи роблять акцент у наступних областях знань:

- Менеджмент сфери управління, оскільки акцент робиться в управлінських вимогах
- Управління людськими ресурсами, оскільки акцент робиться на командну роботу
- Керування якістю, навіть не формально визначене, сприяє використанню стандартів, тестуванню та частому перегляду

З іншого боку, гнучкі методи не повністю враховують наступне:

- Ризик не керується явно
- Управління витратами не є частиною гнучких методологій
- Управління закупівель взагалі не розглядається

У даній роботи створена спроба врахувати перший недолік.

1.3 Ризики людського фактора та комунікації

Враховуючи, що інженерія програмного забезпечення є інтенсивною діяльністю людського капіталу; важливість управління емоціями в професії програмного забезпечення очевидна. Емоції є ключовими проблемами в поведінці людей. [3] Чим більше методологія приймає до уваги людські фактори, тим успішніше вона стає в реальному світі. Це тому, що людські та соціальні фактори мають дуже сильний вплив на успіх розробки програмного забезпечення та остаточної системи. [4]

Інтелектуальний аналіз емоцій, що застосовується до звітів розробників, може бути корисною для ідентифікації та контролю настрою команди розробників, що дозволяє керівнику проекту передбачати та вирішувати

потенційні загрози в своїй команді, а також виявляти та стимулювати фактори, які приносять спокій і продуктивність спільноти. [5]

За словами Atlassian, найбільшою проблемою, з якою стикаються команди сьогодні, є спілкування.[6] Коли робота в команді зроблена правильно, переваги очевидні:

- 50% більш мотивовані успіхом команди, ніж компанії (27%) або індивідуальним успіхом (23%).
- 43% вважають, що вони мають великий особистий вплив на місію своєї основної команди проти 33% на місію компанії в цілому.
- 56% почувають впевненіше працювати в команді, ніж індивідуально.

Тож інструмент для аналізу розмов розробників може допомогти не лише у більш швидкому визначення проблем у розробці, а й у підтриманні здорової атмосфері у команді.

1.4 Аналіз ризиків

Ризик – це можливість чи загроза відхилення результатів конкретних дій від очікуваних [2].

Проектні ризики - сукупність ризиків, що загрожують реалізації інвестиційного проекту чи можуть знизити його ефективність (комерційну, економічну, бюджетну, соціальну, екологічну тощо); сукупність обставин за яких ймовірність завершення поставлених цілей проекту зменшується або виключається; сукупність ризиків, які зумовлюють загрозу економічній

ефективності проекту, що виражається в негативному впливі різних чинників на грошові потоки

Важливість ризику (risk exposure) — показник, який може використовуватися в процесі ухвалення рішення і як механізм контролю за ризиками в проекті.

Ймовірність ризику (risk probability) — це міра можливості того, що наслідок (дія) ризику дійсно буде мати місце.

Загроза ризику (risk impact) — міра серйозності негативних наслідків, рівень збитків або оцінка потенційних можливостей, пов'язаних з ризиком.

Для моніторингу поточного стану ризиків можна визначити показники ризиків. Ці показники можуть бути кількісними (ймовірність виникнення або зусиль та витрат на контрольні заходи) або якісні (оцінка мотивації персоналу проекту). Інший метод моніторингу полягає у використанні тригерів, які є пороговими значеннями для показників, які запускають заходи, коли вони досягнуті.

Якісний аналіз проектних ризиків — це процес надання якісного аналізу ідентифікації ризиків і визначення ризиків, що вимагають швидкого реагування. Така оцінка ризиків визначає ступінь важливості ризику й обирає спосіб реагування. Доступність супровідної інформації допомагає легше розставити пріоритети для різних категорій ризиків.

Якісний аналіз ризиків включає в себе розстановку пріоритетів для ідентифікованих ризиків, результати якої використовуються згодом, наприклад, в ході кількісного аналізу ризиків або планування реагування на ризики.

Основними результатами якісного аналізу ризиків є:

- Ранжування загального ризику проекту.

- Список ризиків по пріоритету, що можуть бути розбиті за пріоритетом та різною кількістю критеріїв.
- Список ризиків для додаткового подальшого аналізу та управління, тобто ті, що потрапили в категорію високих або середніх.

Якісний аналіз ризиків передбачає швидке і малозатратне встановлення пріоритетів виявлених ризиків, виходячи з ймовірністю їх появи та відповідного їх впливу на цілі проекту в разі, якщо ризики виникають.

Кількісний аналіз проектних ризиків – визначає ймовірність їх виникнення і вплив наслідків ризиків на проект, що допомагає групі менеджменту проекту правильно приймати рішення і уникати невизначеностей.

Кількісна оцінка ризиків дозволяє визначити:

- Ймовірність досягнення кінцевої мети проекту
- Ступінь впливу ризику на проект й обсяги непередбачених витрат і матеріалів, які можуть знадобитися
- Ризики, що вимагають якнайшвидшого реагування й більшої уваги, а також вплив їхніх наслідків на проект
- Фактичні витрати, передбачені строки закінчення.

Кількісна оцінка ризиків часто супроводжує якісну оцінку й також вимагає процесу ідентифікації ризиків. Кількісна й кількісна оцінки ризиків можуть використатися окремо або разом, залежно від наявного часу й бюджету, необхідності в кількісній і якісній оцінці ризиків.

1.5 Комунікації розробників та їх оцінка

Система відстеження проблем є сховищем, в якому розміщені всі завдання розробки для організації та підтримки програмного забезпечення та еволюції. Atlassian Jira є найпопулярнішим Agile-специфічним інструментом, що використовується більш ніж 50 мільйонами користувачів [7]. Він забезпечує спільне середовище, де члени команди можуть подавати і обговорювати питання, просити поради і ділитися думками, корисними для заходів з підтримки або дизайнерських рішень [5].

Звіт про проблему характеризується стандартними полями, корисними для його обробки (наприклад, пріоритет, статус тощо). Крім того, він містить хронологію коментарів та вкладень розробників, що представляють щоденну комунікацію про поточну роботу, невдачі та успіхи, і «як члени комітету відчувають помилку, функцію, проект або навіть інших членів спільноти» [7], Що дає цінність вплив на соціальні взаємодії розробників.

Добування емоції з тексту набуває все більшої популярності для розподілених команд, коли особиста взаємодія майже виключена. Таким чином, існує потреба в управлінні, щоб бути в курсі емоцій розробників, щоб якнайшвидше запобігти ризикам.

Одним з перших досліджень у цьому напрямку визначення емоцій розробників, а не їх поведінки було проведено Murgia et al. хто поставив питання про відсутність досліджень у цій галузі [5]. Автори позначили коментарі як «одне повідомлення - одна емоція», використовуючи рамки Парротта (любов, радість, здивування, гнів, смуток, страх), щоб виміряти людську згоду щодо їхньої присутності у звітах про проблеми.

Дослідження, проведене M. Ortu, G. Destefanis, B. Adams et al. також показав, що «коментарі розробників містять не тільки технічну інформацію, але й цінну інформацію про почуття та емоції» [8]. Jongeling et al. [9] використовував цей набір даних, щоб перевірити, чи знаходяться інструменти

для навчання машин для емоцій, отримані від соціальних даних, у згоді з даними, позначеними вручну. Пізніше наявність цього сховища призвела до подальших досліджень та експериментів з емоцій розробників: виявлення вигорання та продуктивності [10][11], вимірювання афективності та ефективності [10], моделювання напрямку емоції гніву та аналізу ввічливості.

Mäntylä, M. та ін [10] використовували цей набір даних у зв'язку з VAD-лексиконом (Valence, Arousal, Dominance) з 13 915 англійських слів [12] для аналізу VAD у звітах про задачі, оскільки вони вважають, що «використання вимірного підходу є більш вигідним ніж використання дискретного підходу, оскільки розмір може бути пов'язаний з вигоранням і продуктивністю».

Взагалі кажучи, емоції вивчалися в психології або з використанням дискретного підходу, або з використанням розмірного підходу. Дискретний підхід представляє емоції як сукупність основних афективних станів, які можна відрізнити однозначно, такі як гнів, радість, смуток і любов. Розмірний підхід (запропонований ще в 1897 р.) групує афективні стани в меншому наборі основних розмірів, наприклад, VAD. До цих пір, минулі дослідження з видобутку впливають на сховища програмного забезпечення, наприклад, зосереджені майже виключно на використанні дискретних емоційних теорій.

1.5 Визначення тематики тексту

Тематичне моделювання є одним з найпопулярніших імовірнісних алгоритмів кластеризації, який останнім часом набуває все більшої уваги. Основною ідеєю моделювання тематики є створення імовірнісної генеративної

моделі для корпусу текстових документів. У тематичних моделях документи являють собою суміш тем, де тема - розподіл ймовірностей над словами.

Дві основні моделі теми - імовірнісний латентний семантичний аналіз (pLSA) [17] і модель латентного розподілу Диріхле (LDA) [18]. Hofmann [17] ввів pLSA для моделювання документів, але вона не забезпечує жодної ймовірнісної моделі на рівні документа, що ускладнює її узагальнення для моделювання нових невідомих документів. Blei et al. [18] розширили її, ввівши розподіл Дирихле у якості ваг суміші тем для кожного документа, і назвали її модель латентного розподілу Діріхле (LDA).

Модель латентного розподілу Діріхле є сучасною технікою “без вчителя” для вилучення тематичної інформації (тем) збірки документів. Основна ідея полягає в тому, що документи представлені у вигляді випадкової суміші прихованих тем, де кожна тема є розподілом ймовірностей над словами.

1.6 Системи підтримки прийняття рішень для управління ризиками

У jira marketplace можна знайти додатки для керування ризиками [19][20], але їх недоліток у тому, что потрібно власноруч визначати вігорідність та значущість кожного ризику. Приклад підсумку для проекту для цих додатків наведені у рис.1.1 та 1.2.

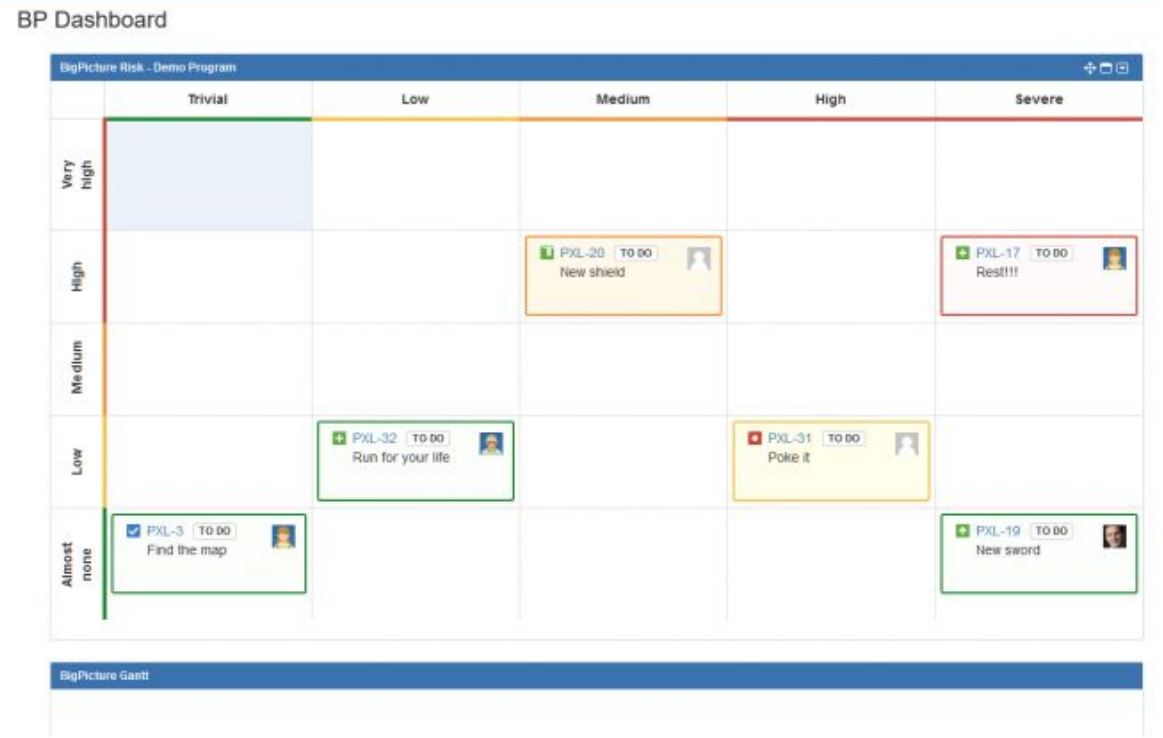


Рис 1.1 – Big Picture

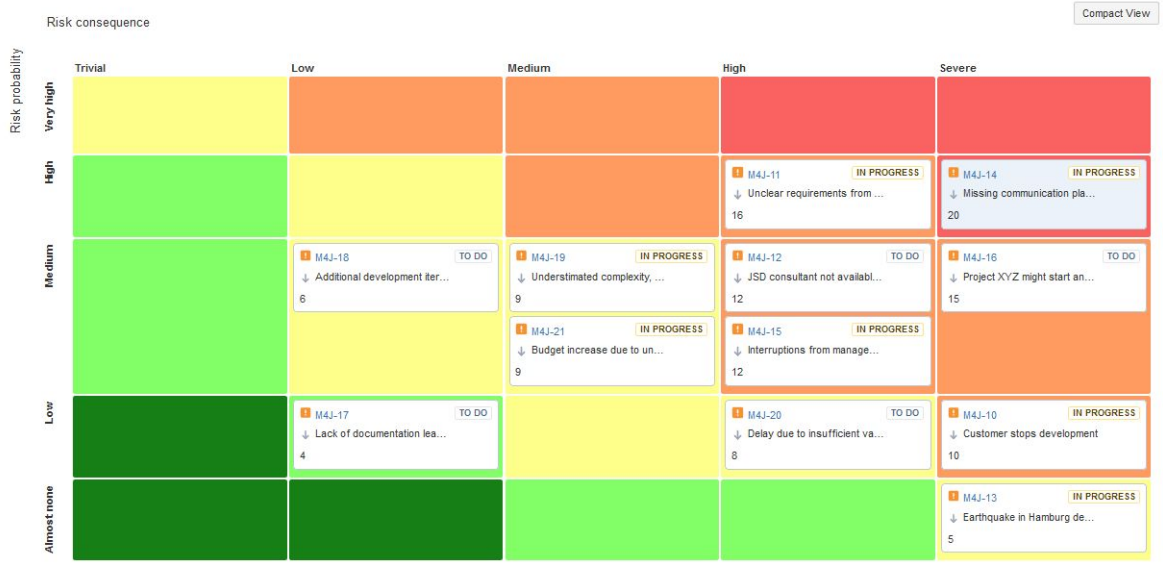


Рис 1.2 – catworks

Emitza Guzman у своїй роботі [13] описала прототип візуалізації, який дає огляд емоційного клімату проекту на основі текстової інформації, як-от пошти та артефакти. Вона складається з двох основних частин: вилучення емоцій з SentiStrength і моделювання тематики з LDA. Перший виражений в кольорах

(зелений - позитивний, жовтий - нейтральний і пурпуровий для негативних) і розмір кола, а другий - в хмарах слів. На рис. 1.3 представлен короткий огляд оцінки комунікацій проектів.

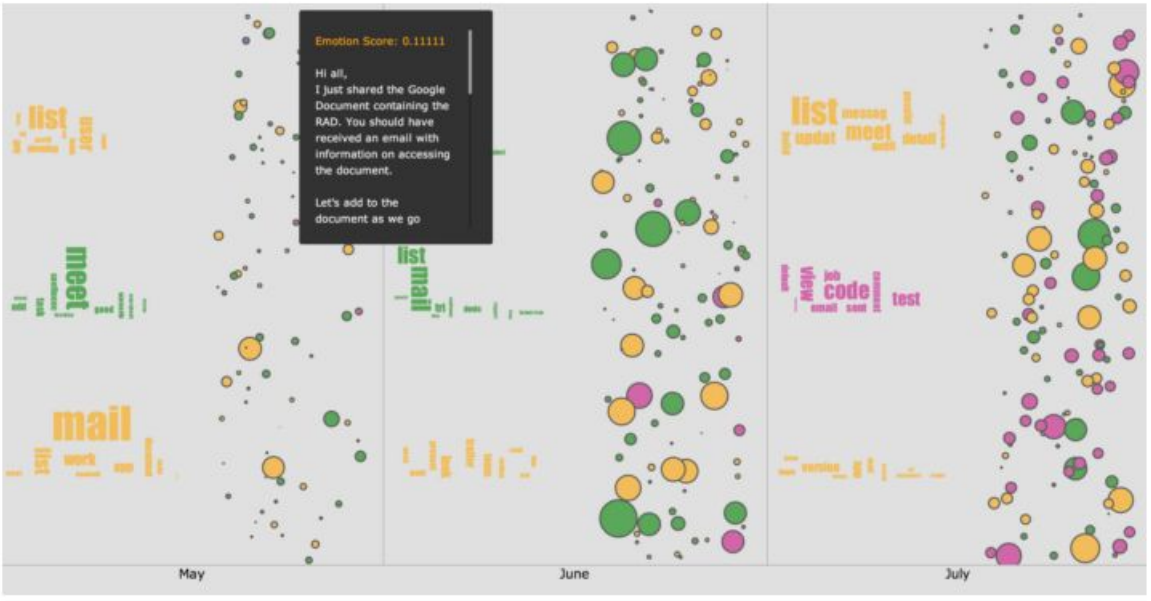


Рис 1.3 – Огляд прототипу візуалізації Guzman

A. Aslam et al. [14] зробили систематичний огляд літератури 80 статей, щоб відповісти на запитання з питань оцінки ризиків, стратегій управління ризиками, факторів, правил та питань, що використовуються для прийняття рішень у сфері розподіленого програмного забезпечення. Вони склали таблицю, яка включає аспекти, які впливають на оцінку ризику та стратегію управління, що відображається на організаційній моделі Leavitt [15] (Завдання, структура, актори та технології). Вони також розробили веб-додаток, що включає анкети, зібрані з різних досліджень, радарні символи, створені на основі відповідей і табличні стратегії управління на основі бази знань. Ця СППР є активною (рекомендує рішення для осіб, які приймають рішення) та співпрацюють (дозволяють особам, які приймають рішення, налаштовувати пропозиції щодо прийняття рішень).

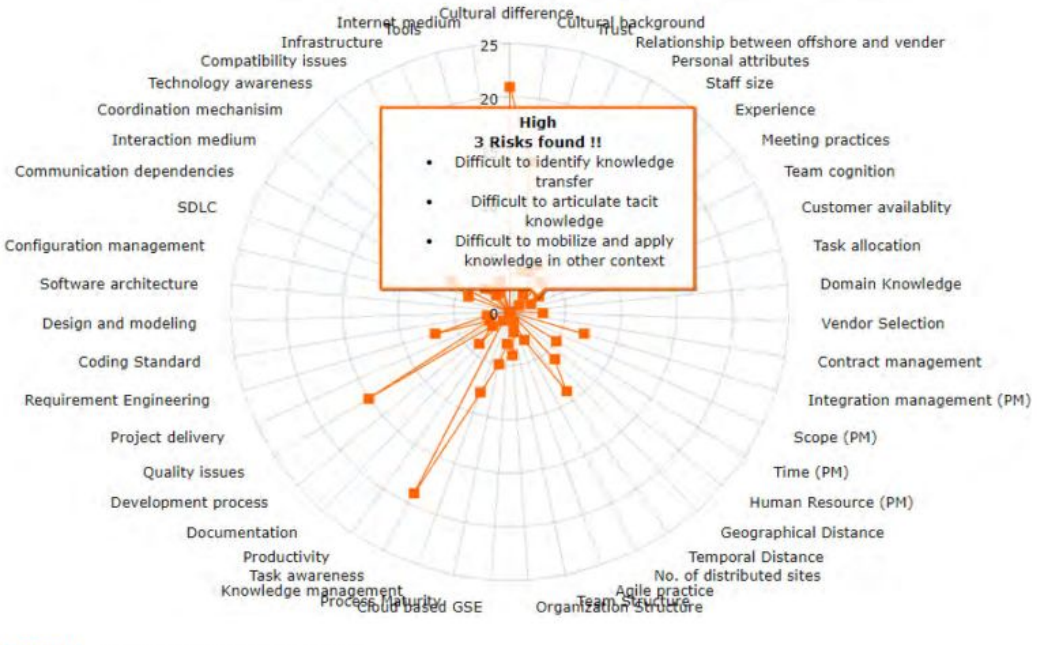


Рис. 1.4 – Радарна діаграма для досвіду Siemens

Q.ID	Question	Answers	Dimension	Avoidance	Mitigation	Transfer	Acceptance
1	What can be the cultural difference between distributed sites ?	High section 4.1 [92]	Actors				
2	What is the cultural background of different distributed sites ?	Same organization section 5.0 [92], [97]	Actors				
3	What is trust level among employees in distributed sites	Medium	Actors				
4	What is the degree of relationship between clients and vendors?	High section 4.1 [92]	Actors				
5	What are the employees skill levels in distributed sites	Medium section 4.3 [92]	Actors				
6	Is your staff fully competent with you or organizations?	Agree section 4.3 [92]	Actors				
7	What are the experience level of employees in your organization?	Medium	Actors				
8	When higher authorities and project manager meeting about issues?	Regularly	Actors				
9	What is the team cognition level between distributed team?	High section 4.1 [92]	Actors				
10	How much customer available during development process?	High section 3.0 [92]	Actors				
11	Is task divided into sub task allocated, and then allocated to distributed sites ?	Agree section 4.1 [92],[98]	Actors				
12	What is the degree of domain knowledge on particular site?	Low section 1.1 [92]	Actors				
13	Is vendor is capable for developing required product?	Medium section 5.0 [92]	Actors				
14	Is contract between sites or parties are concise and clear?	partially agree section 3.0 [92]	Actors				

Рис 1.5 – Табличні стратегії управління Philips

Висновки до розділу 1

У першому розділі розглянута актуальність поставленої задачі та базові поняття, пов’язані з нею, а саме - управління проектами, аналіз ризиків, комунікація в проекті та їх важливість для IT проектів, інтелектуальний аналіз тексту на предмет емоцій.

Існує багато різних підходів для визначення емоцій, хоча цей напрямок дослідження з'явився досить недавно - лише в 2014 році. Поштовхом для такого швидкого розвинення стала поява датасета з коментарями розробників відкритих jira-проектів. Саме цей датасет і буде використовуватися при побудові експериментів. Для обчислення буде використовуватися вимірні показники VAD.

Зроблений огляд основних методів та алгоритмів визначення тематики текстів та графічного представлення. Далі у роботі буде використовуватися модель LDA для відносення ризиків до певної категорії та word clouds для графічного представлення.

РОЗДІЛ 2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ТЕКСТІВ ТА РИЗИКІВ

2.1 Визначення емоційних показників

Спочатку кожний коментар позбавляється від пунктуації та чисел, а потім розбивається на токени, у результаті отримаємо представлення у вигляді “мішка слів” (Bag-Of-Words).

Наприклад:

Коментар: “I'm afraid that you have to install the pg_bulkload manually on all nodes at the moment. There isn't possibility to distribute pg_bulkload with the job. But I believe it's interesting idea for sqoop 2 (and for direct connectors in general).”

Токени: ['i', 'm', 'afraid', 'that', 'you', 'have', 'to', 'install', 'the', 'pg', 'bulkload', 'manually', 'on', 'all', 'nodes', 'at', 'the', 'moment.', 'there', 'isn', 't', 'possibility', 'to', 'distribute', 'pg', 'bulkload', 'with', 'the', 'job', 'but', 'i', 'believe', "it's", 'interesting', 'idea', 'for', 'sqoop', 'and', 'for', 'direct', 'connectors', 'in', 'general']

Далі ми лематизуємо кожний токен, тобто приводимо його до початкової форми. Наприклад, 'interesting' стає 'interest'.

Далі для кожного токену коментаря ми визначаємо показники валентності, збудження і домінування (ВЗД) за таблицею оцінок Warriner та ін. [12] з 13 915 англійських слів, залишаючи для подальшого розрахунку лише максимальні та мінімальні значення показників з усіх слів. Тож єдиним можливим варіантом токенізації є розбиття на уніграми через те, що словник містить лише слова, а не фрази.

Перед прикладом, розглянемо значення кожної зі складових [10].

Валентність (Valence) - це емоційний вимір, пов'язаний з привабливістю (або несприятливістю) події, об'єкта або ситуації. Термін означає напрямок поведінкової активації до стимулу (апетитною мотивацією) або відхиленням від нього (аверсивна мотивація).

Збудження (Arousal) - це розмірність, що представляє рівень емоційної активації. Вона має різні фізіологічні та психологічні реакції, наприклад, підвищену частоту серцевих скорочень і настороженість до відповідей, і вона сприймається як відчуття реактивності до подразників і психічного пробудження. Збудження також підсилює задоволення або незадоволення, що описується валентним виміром, наприклад, розчарування може змінити гнів і мирне щастя може змінитися в захват, коли збудження збільшується.

Домінантність (Dominance) являє собою зміну відчуття контролю над стимулом (або ситуацією).

Розглянемо декілька прикладів у таблиці 2.1. Кохання та радість мають більшу валентність як позитивні емоції, смуток має більш пасивну природу, тож отримує низькі показники збудження та домінування.

Таблиця 2.1 – Представлення слів у просторі ВЗД

Слово	Валентність	Збудження	Домінування
Anger / гнів	2.50	5.93	5.14
Joy / радість	8.21	5.55	7.00
Sadness / смуток	2.40	2.81	3.84
Love / кохання	8.00	5.36	5.92
Середнє	5.06	4.21	5.19

Для особливих випадків, коли максимум нижче середнього значення або коли мінімум вище, ми встановлюємо max або min до середнього значення всіх слів лексики. Далі ми використовуємо дані значення для розрахунку відносних показників за наступною формулою:

$$Range(\bar{w}) = \begin{cases} max(\bar{w}) - avg(\bar{W}), \ if \ min(\bar{w}) > avg(\bar{W}) \\ avg(\bar{W}) - min(\bar{w}), \ if \ max(\bar{w}) < avg(\bar{W}) \\ max(\bar{w}) - min(\bar{w}), \ if \ min(\bar{w}) \leqslant avg(\bar{W}) \leqslant max(\bar{w}) \end{cases} \quad (2.1)$$

Наприклад, якщо коментар буде містити всі слова, наведені в таблиці 2.1, він отримує оцінку валентності 5,81 (8,21-2,40, див. третій випадок у формулі). Чим вище значення, тим більш екстремальні бали ВЗД.

Таким чином дані показники визначають значущість наявності цих емоційних станів та ступінь їх відмінності від середніх значень.

2.2 Аналіз ризиків задачі

Для того, щоб бути здатними оцінити ризики всього проекту, спочатку треба перейти від оцінок одного коментаря для оцінок всієї задачі.

Перше, що треба брати до уваги - це час та відповідні до нього ваги. Задача може мати багато коментарів, включаючи й негативні, але якщо останні позитивні та вирішують проблему, то їх вага повинна бути більшою ніж попередніх. Таким чином за точку відліку можна взяти час першого коментаря (0), а за верхню межу (1) - поточний час, нормування часу коментаря у цьому проміжку дає нам вагу актуальності коментаря.

Обчислення ваги відповідно до часу першого повідомлення також допоможе отримати уявлення про інтенсивність обговорення.

У подальшому, маючи історичні дані, аналіз часових рядів може бути застосований для передбачення інтенсивності комунікацій та зміни емоційний показників.

Таким чином для кожного коментаря маємо:

- Валентність
- Збудження
- Домінування
- Актуальність

Та для задачі отримаємо зважені оцінки формулу 2.2:

$$S = \frac{1}{n} \sum_{i=1}^{n} w_i s_i \tag{2.2}$$

де S - загальна оцінка задачі (валентність / збудження / домінування),

s_i - оцінка (валентність / збудження / домінування) коментаря

w_i - актуальність коментаря

2.3 Перехід до матриці ризиків

Таблиці оцінки ризиків дають можливість організаторам подій розподіляти рейтинги ризиків на всі небезпеки, щоб вони могли визначати пріоритети та систематично вирішувати їх.

Маючи підхід для оцінки ризику по емоціям за коментарем, треба масштабувати його для всього проекту. За основу можна обрати підходи, описані у розділі 1.4. При побудові матриці ризиків для проекту, помістемо задачі або їх кількість в клітини.

CONSEQUENCE					
LIKELIHOOD*	Insignificant 1	Minor 2	Moderate 3	Major 4	Catastrophe 5
A (Almost certain)	H	H	E	E	E
B (Likely)	M	H	H	E	E
C (Possible)	L	M	H	E	E
D (Unlikely)	L	L	M	H	E
E (Rare)	L	L	M	H	H

Рис 2.1 – Матриця ризиків

За даною таблицею маємо наступні типи ризиків:

- E=Екстремальний: необхідні негайні дії
- H=Високий ризик: необхідна увага старшого керівництва
- M=Помірний: відповідальність керівництва повинна бути визначена
- L=Низький: управління за допомогою рутинних процедур

Кожен тип ризику є результатом поєднання двох його властивостей - ймовірності (див. таб. 2.2) та значущості наслідків (див. таб. 2.3).

Таблиця 2.2 – Ймовірність наслідків.

Рівень	**Значення**	**Опис**	**Приклад детального опису**
A	5	Безумовно	Очікується, що це відбудеться в більшості випадків
B	4	Імовірно	Ймовірно, відбудеться в більшості випадків
C	3	Можливо	Можуть відбутися на певний час
D	2	Навряд чи	Може статися через деякий час
E	1	Рідко	Може відбуватися, але тільки за виняткових обставин

У таблиці нижче наведені рівні значущості та відповідні пріоритети, що використовуються в JIRA.

Таблиця 2.3 – Значущість наслідків

Рівень	**Опис**	**тип Jira**	**Приклад детального опису**
1	Мізерні	Trivial	косметична проблема, як помилкові слова або змішаний текст
2	Незначні	Minor	незначна втрата функції або інші проблеми, де легко обходиться
3	Помірні	Major	велика втрата функціональності
4	Значні	Critical	аварії, втрата даних, витік пам'яті
5	Катастрофа	Blocker	блокує розробку та / або роботу з тестування, виробництво не може працювати

Таким чином інтегрований показник ВЗД може виступати в якості вірогідності, у той час як важливість зазвичай визначається менеджером.

Також важливість може бути збагачена вагами для типу завдання (баг, фікс, нова фіча), тривалістю виконання задачі (чим довше, тим гірше), а найголовніше - структурою проекту, тобто скільки завдань можуть бути в очікуванні через поточну, скільки виконавців у поточних та супутніх завдань.

Вірогідність ризику оцінюємо за таблицею 2.4.

Таблиця 2.4 – Ймовірність як інтегрований показник ВЗД

Рівень	**Значення**	**Опис**	**Інтегрований показник ВЗД**
A	5	Безумовно	>=10
B	4	Імовірно	(10, 8]
C	3	Можливо	(7, 5]
D	2	Навряд чи	(5, 2]
E	1	Рідко	<2

Далі визначаємо загальну оцінку ризику як добуток вірогідності на значущість.

$$VR = A * q \tag{2.3}$$

де: *VR* — важливість ризику;

A — загроза (наслідок, дія) ризику (небажаної події);

q — ймовірність її настання.

2.4 Визначення теми ризику

Для швидкої оцінки ситуації потрібно розуміти, що саме пішло не так. У цьому можуть допомогти кілька ключових слів або тематика проблеми.

Одним з найбільш поширених методів побудови тематичних моделей є Latent Dirichlet Allocation (LDA), який моделює документ як розподіл тем і тему як розподіл слів. Тут документ - це коментар.

Розглянемо іграшну модель LDA, що виробляє наступні теми:

Тема 0: '0.075*"patch" + 0.040*"fix" + 0.039*"cassandra_num_" + 0.020*"trunk" + 0.020*"attach" + 0.017*"v_num_" + 0.015*"apply" + 0.013*"change" + 0.011*"version" + 0.011*"issue"'

Тема 1: '0.018*"thrift" + 0.017*"table" + 0.016*"make" + 0.015*"change" + 0.014*"cql_num_" + 0.014*"use" + 0.013*"patch" + 0.013*"bq" + 0.012*"would" + 0.012*"think"'

Тема 2: '0.043*"cql" + 0.028*"id" + 0.020*"select" + 0.020*"_num_e_num_" + 0.016*"make" + 0.016*"would" + 0.010*"loop" + 0.009*"pprop" + 0.009*"eentid" + 0.009*"python"'

Тема 3: '0.036*"flush" + 0.017*"write" + 0.017*"call" + 0.012*"memtable" + 0.012*"segment" + 0.011*"thread" + 0.011*"replay" + 0.011*"get" + 0.010*"new" + 0.009*"need"'

Тоді представлення речення “moves strategy creation into Table instantiation so it can't be out of sync” в цьому просторі буде [(0, 0.1), (1, 0.50), (2, 0.16), (3, 0.24)].

Алгоритм LDA заснований на попередньому розподілі Діріхле і передбачає модель «мішок слів» - модель для аналізу текстів, яка враховує тільки частоту слів, але не їх порядок. Ця модель добре підходить для тематичного моделювання, оскільки вона дозволяє виявляти неявні зв'язки між словами. Метод LDA виконує м'яку кластеризацію і припускає, що кожне слово у реченні генерується деякою прихованою темою, яка визначається розподілом ймовірностей на множині всіх слів тексту.

Маючи корпус D, що складається з M документів, для документа d, що має N_d слів ($d \in \{1, ..., M\}$), LDA моделює D згідно з наступним генеративним процесом [18]:

1) Вибір полиномиального розподілу φ_t для теми t ($t \in \{1, ..., T\}$) з розподілу Діріхле з параметром β.
2) Вибір поліноміального розподілу θ_d для документа d ($d \in \{1, ..., M\}$) з розподілу Діріхле з параметром α.
3) Для кожного слова w_n ($n \in \{1, ..., N_d\}$) в документі d:
    a) Вибрати тему z_n з θ_d.
    b) Виберіть слово w_n з z_n.

У вищезгаданому генеративному процесі слова в документах є єдиними спостережуваними змінними, тоді як інші - латентні змінні (і θ) і гіперпараметри (α і β). Для того, щоб зробити висновок про приховані змінні і гіперпараметри, ймовірність спостережуваних даних D обчислюється і максимізується наступним чином:

$$p(D|\alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha)\left(\sum_{n=1}^{N_d} p(z_{dn}|\theta_d)p(w_{d_n}|z_{dn},\varphi)P(\varphi|\beta)\right)d\theta_d d\varphi \quad (2.4)$$

Внаслідок зв'язку між θ і φ в підінтегральній функції у формулі. (2.4), точний висновок у LDA є нерозв'язним. Різні наближувальні алгоритми, такі як

варіаційний висновок або ланцюг Маркова Монте-Карло (MCMC), зазвичай використовуються для виведення в LDA.

У цій роботи ми будемо використовувати пакет gensim, який має реалізацію online LDA. Цей алгоритм використовує стохастичну оптимізацію, щоб максимізувати варіаційну цільову функцію для моделі тематичного розподілу прихованих дирихле (LDA). Він тільки дивиться на підмножину загального корпусу документів кожної ітерації і тим самим здатний швидко знайти локально оптимальне налаштування варіаційного апостера над темами [21].

Для визначення теми, агрегуємо усі коментарі та застосуємо модель LDA, роблячи припущення, що кількість тем відповідає кількості задач.

Дві важливі методи, які використовуються для оцінки моделей теми:

- розгубленість (може бути не такою доброю мірою)
- когерентність теми

З огляду на підготовлену модель, розгубленість намагається виміряти, як ця модель дивується, коли їй дається новий набір даних. Це вимірюється як нормалізована логарифмічна ймовірність витриманого тестового набору. Чим нижче розгубленість, тим краще модель.

$$L(D') = \frac{\sum_{D} log_2 p(w_d;\Theta)}{N} \tag{2.5}$$

$$perplexity(D') = 2^{-L(D')} \tag{2.6}$$

w_d - невидимі дані в утриманному наборі

Θ - вивчені параметри моделі

N - кількість токенів

Перше рівняння обчислює логарифмічну ймовірність; ймовірність спостереження за деякими невидимими даними з урахуванням моделі, отриманої раніше. Це перевіряє, чи фіксує модель розподіл витриманого

набору. Якщо цього не відбудеться, то розгубленість дуже висока припускаючи, що модель поганою.

Для визначення когерентність теми існує дві основні метрики - C_v та C_{umass}.

C_V базується на ковзному вікні, однокомпонентній сегментації топ-слів і непрямій мірі підтвердження, що використовує нормалізовану точкову взаємну інформацію (NPMI) і схожість за косинусом. Ця міра когерентності отримує кількість спів зустрічання для заданих слів за допомогою ковзного вікна і розміру вікна 110. Підрахунки використовуються для обчислення NPMI кожного топ-слова для кожного іншого топ-слова, таким чином, що призводить до набору векторів - для кожного топ-слова.

$$NPMI(w,w) = \frac{log\frac{P(w_i,w_j)+e}{P(w_j)}}{log(P(w_i,w_j)+e)} \quad (2.7)$$

Однокомпонентна сегментація топ-слів призводить до розрахунку подібності вектора кожного топ-слова і суми векторів всіх топ-слів. Як міра подібності використовується косинус. Когерентність - це середнє арифметичне з цих подібностей. [22]

C_{umass} була запропонована Mimno et al. [23], ця метрика бере до уваги упорядкування серед топ-слів теми та маэ наступний вигляд:

$$C_{UMass} = \frac{2}{N*(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} log\frac{P(w_i,w_j)+e}{P(w_j)} \quad (2.8)$$

Де N - кількість топ-слів узятих для аналізу

Оскільки при будуванні моделі враховується дуже багато коментарів, то при визначенні теми документу топ-словами можуть слова, не притаманні даній проблеми, але дуже близькі. Для цього краще зробити перетин слів та поза вагою слова у теми, урахувати вагу (вірогідність) самої теми. Тоді алгоритм визначення теми ризиків задачі виглядає наступним чином:

1. Створити порожню таблицю T для слів та ваг задачі
2. Для кожного коментаря C_i обраної задачі:
    a. Визначити список слів W коментаря
    b. Визначити теми T даного коментаря
    c. Для кожної теми T_i та ваги цієї теми *topic_weight* для коментаря C_i:
        i. Визначити топ-N слів W_i^t з вагами *word_weigh* кожного слова
        ii. Для кожного слова теми:
            ➢ Якщо слово присутнє в коментарі:
                ✓ T[word] += *word_weigh* * *topic_weight*

Найбільш зручним представленням теми є хмара слів (word clouds). Оскільки після моделювання ми отримаємо набір пар слово-вага, то можемо створити зображення, де розмір слова буде пропорційний його вагі.

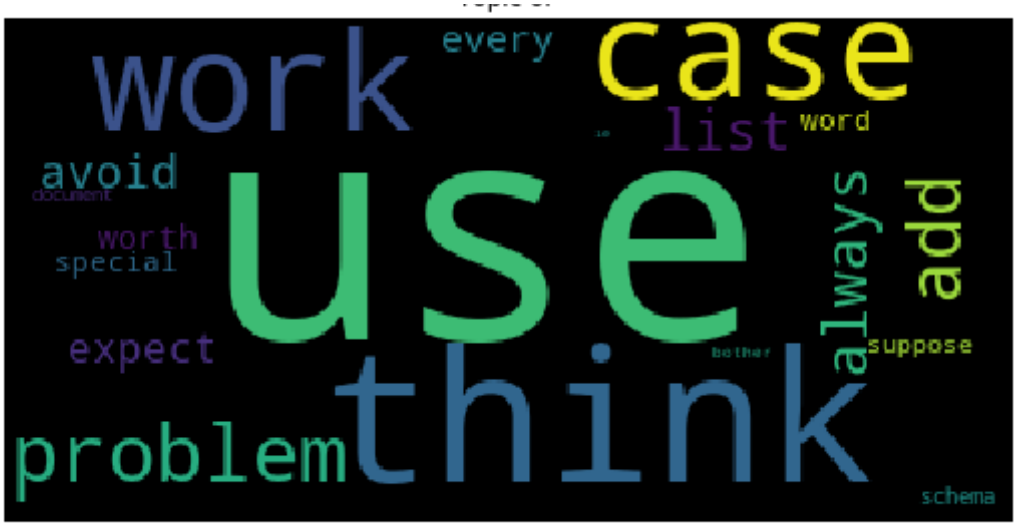


Рис 2.2 – Приклад хмари слів

Висновки до розділу 2

У другому розділі наведено метод визначення емоційних складових тексту, їх перетворення до ймовірності ризику, обчислення актуальності коменаря та зваженої ймовірності ризику з урахуванням останнього.

Далі описана модель LDA, критерії обрання оптимальних параметрів. Та алгоритм переходу від визначення тем коментарів задачі до побудови її назви з ключових слів з подальшим представленням у вигляді хмари слів.

РОЗДІЛ 3 ТЕСТУВАННЯ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДІВ АНАЛІЗУ РИЗИКІВ ІЗ ЗАСТОСУВАННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір програмних засобів, які можуть бути використані для розв’язання задачі

Лідерами серед мов програмування в сфері аналізу даних є Python та R. Обрано було мову Python3 як більш зручну для машинного навчання та за її розмаїття бібліотек, в т.ч.:

- pandas
- numpy
- sklearn
- matplotlib
- spacy
- gensim
- wordcloud

Середою розробки було обрано Jupyter через зручність роботи з комірками. Оскільки коментарі були наведені у вигляді дампу реляційної базі даних, то був використан postgresql, то використовувався відповідний конектор - psycopg2. Для обробки тексту спочатку використовувався NLTK, але у подальшому планується використання spacy та gensim для лематизації, вилучення ключових слів та побудови тем корпусу.

3.2 Опис датасету

Датасет jira_emotion [24] представляє собою набір даних, витягнутих з Jira ITS чотирьох популярних екосистем з відкритим вихідним кодом (а також інструментів та інфраструктури, що використовуються для видобутку) спільноти Apache Software Foundation, Spring, JBoss та CodeHaus. У наборі даних розміщено більше 1К проектів, які містять понад 700 тис. звітів про задачі і більше 2 млн. коментарів. Використовуючи ці дані, автори змогли глибоко вивчити процес спілкування між розробниками, і як цей аспект впливає на процес розвитку. Крім того, коментарі розробників містять не тільки технічну інформацію, але й цінну інформацію про почуття та емоції. Повний набір даних (включаючи проекти Apache) містить 3516 завдань та 25306 коментарів 1375 авторів.

Задачі в Jira поділяються на такі категорії, як помилки, поліпшення, запити на функції або завдання. Наведений набір даних містить задачі, що належать до всіх трьох категорій.

У нашому випадку ми використовували лише поля *comment, issue_report_id* та *updateDate* з таблиці *jira_issue_comment*, а також поля *id, priority* та *project* з таблиці *jira_issue_report*.

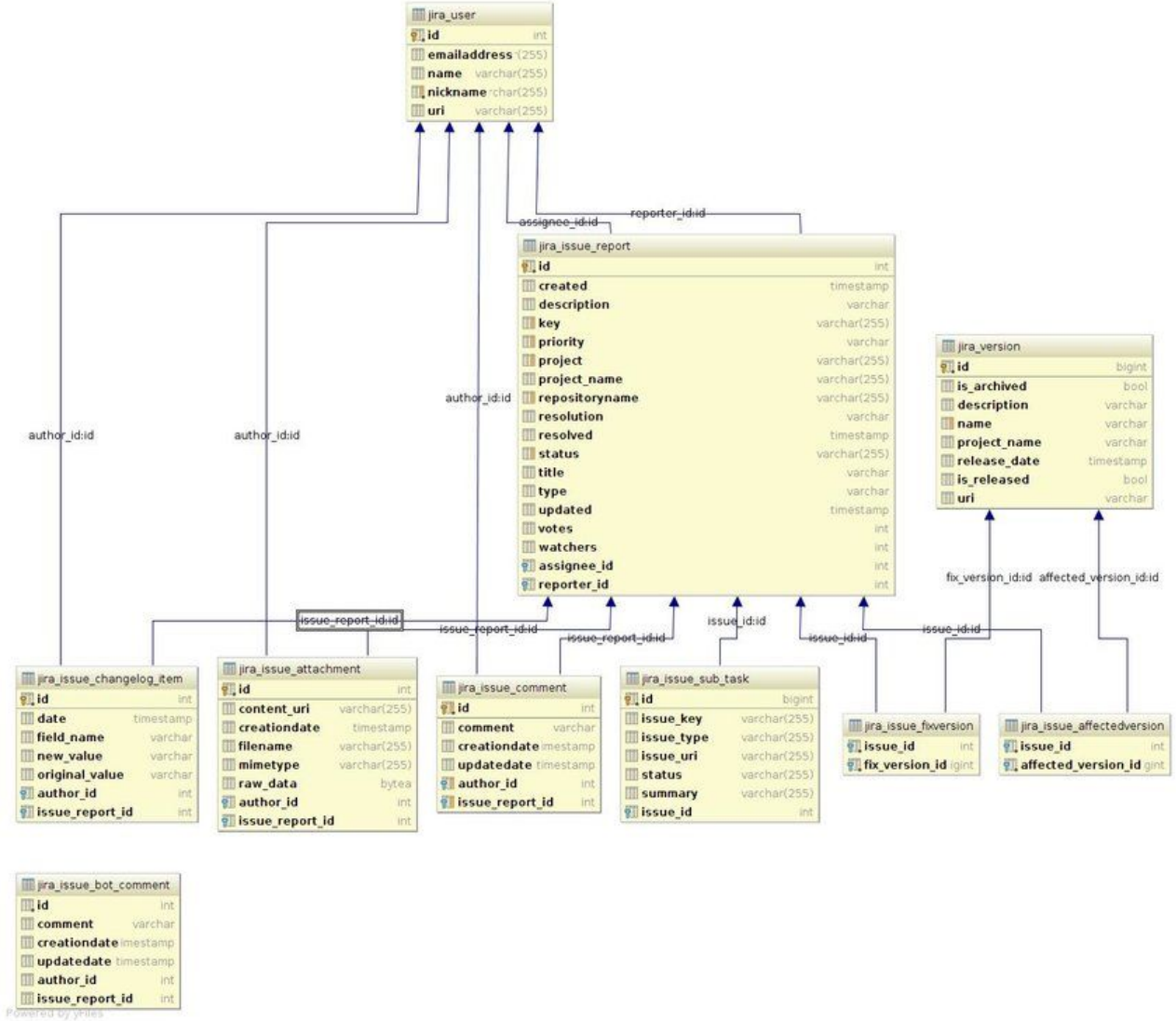


Рис. 3.1 – Схема бази даних

3.3 Предобробка даних

У даному випадку ми працюємо з коментарем. Кожний коментар розбивається на токени незалежно від речень, усі токени лематизуються, у результаті отримаємо BagOfWords представлення. Єдиним можливим

варіантом токенізації є розбиття на уніграми через те, що словник містить лише слова, а не фрази.

В якості прикладу було обрано достатньо відомий проект CASSANDRA компанії Apache Software Foundation. Тож у подальших розділах розглядаються задачі та коментарі, що стосуються лише цього проекту. Всього 41966 коментарів у 6271 задачах з 2009-03-07 по 2013-12-18.

3.4 Визначення емоційних показників

Спочатку кожний коментар токенізувався за допомогою word_tokenize з пакету NLTK, далі токени лематизувалися (приводились до початкової форми) за допомогою SpaCy та фільтрувалися за наявністю в словнику оцінок.

comment	clean_tokens
Attaching screenshot from debugger.	[]
I wouldn't say it's a bug. "id" wouldn't be a list, but some single value, so "in(id)" for variadic IN's is there for a reason, to distinguish between the two. The java-driver might need some modifications though.	[would, say, bug, would, would, list, single, value, reason, distinguish, two, java, driver, may, need, modification]
Aleksey is right, I'm sorry for the brain fart, I mixed it up while responding to the original email, thinking we had inverted when we should have returned 'in(id)' but no. This is working as designed.	[right, sorry, brain, fart, mix, respond, original, email, think, return, work, design]
+1	[]
Committed, thanks.	[thank]
Most likely the problem is https://datastax-oss.atlassian.net/browse/JAVA-213	[likely, problem]
Introduced by CASSANDRA-2524	[]
Committed, thanks.	[thank]
Drivers are not part of the Apache project.	[part, apache, project]

Рис 3.2 - Приклад коментарів та токенів після фільтрації

Далі обчислюємо оцінки ВЗД за формулою range вказаної у розділі 2.1.

comment	clean_tokens	VAD
This is intentional. So long as you are a valid user, you can see the schema, if auth is enabled (that and some other system stuff that our tools require).\r\n\r\nThere is no practical way to limit this, so we don't.	[intentional, valid, user, see, enable, system, stuff, tool, require, practical, way, limit]	[2.6, 1.43, 2.27]
can compression and compaction parameter play a role in that problem?	[compression, play, role, problem]	[4.03, 1.0, 1.52]
Converted the map in question to CHM. Thanks for the report!	[map, question, thanks, report]	[2.96, 0.691, 1.78]
Should clarify, version is 1.2.11-SNAPSHOT as of this weekend.	[clarify, version, weekend]	[2.106, 2.96, 1.925]

Рис 3.3 – Приклад коментарів та їх оцінок

Далі обчислюємо актуальність коментаря. Для цього переведемо дату в формат unix та пронормуємо для кожної задачі окремо, розділивши на максимум, тобто на дату-час останнього коментаря.

comment	date	relevance
Is this different from CASSANDRA-1016?	2010-07-23 23:14:02.861	0.473519
The implementation guarantees that triggers will be executed at least once even if the update is...	2010-12-28 18:39:56.774	0.554297
Probably makes more sense to keep the trigger at the table level and pass it key + CF instance, ...	2012-11-10 13:35:52.249	0.903796
like to know by when this trigger feature will be available?	2013-02-26 17:08:07.477	0.959153
Hi Jonathan,\r\n\r\nRemoved LinkedList allocation in v3 and pushed to https://github.com/Vijay2w...	2013-05-16 13:27:50.511	0.999512
I'm going to have to object one more time to storing a jar file in the file system. With large s...	2013-05-17 12:20:01.381	1.000000

Рис 3.4 – Приклад значення актуальності для коментарів задачі 334549

Далі обчислюємо спочатку інтегроване, як суму складових та множимо на актуальність, щоб отримати зважене значення.

comment	VAD	Integral_value	relevance	Integral_value_weighed
Is this different from CASSANDRA-1016?	[0.846, 0.261, 1.285]	2.392	0.473519	1.132658
The implementation guarantees that triggers will be executed at least once even if the update is...	[5.56, 3.38, 3.44]	12.380	0.554297	6.862199
Probably makes more sense to keep the trigger at the table level and pass it key + CF instance, ...	[1.47, 3.7, 1.97]	7.140	0.903796	6.453101
like to know by when this trigger feature will be available?	[2.59, 2.61, 1.405]	6.605	0.959153	6.335205
Hi Jonathan,\r\n\r\nRemoved LinkedList allocation in v3 and pushed to https://github.com /Vijay2w...	[4.04, 3.55, 3.15]	10.740	0.999512	10.734761
I'm going to have to object one more time to storing a jar file in the file system. With large s...	[4.61, 3.68, 2.94]	11.230	1.000000	11.230000

Рис 3.5 – Приклад інтегрованого та інтегрованого зваженого значення

Для того, щоб отримати вірогідність ризиків у завданні, обчислемо для кожного коментаря інтегрований показник ВЗД та їх середня для завдання. Далі за шкалою, вказаною у розділі методології, призначено рівень вірогідності. Додамо значення пріоритету та обчислимо результуючу важливість ризику. Результати наведені у рис.3.6.

Id	Integral_value	Integral_value_weighed	likelihood	priority	priority_value	rate
333428	11.390000	11.390000	5	Blocker	5	25
329678	11.130000	11.130000	5	Blocker	5	25
333669	10.755556	10.737776	5	Blocker	5	25
331283	10.720000	10.720000	5	Blocker	5	25
329510	10.184286	10.184142	5	Blocker	5	25
333505	11.730000	11.730000	5	Critical	4	20
333436	11.518667	11.514989	5	Critical	4	20
330706	10.750000	10.750000	5	Critical	4	20

Рис 3.6 – Кінцевий показник та вірогідність для задач

3.5 Визначення назви ризику

Для визначення тем корпусу, ми використали лемматизовані токени коментарів для створення словника (42869 слів) та побудови корпусу у представленні “мішка слів” (bow - bag-of-words).

Наприклад,

comment: (as noted on the mailing list, this does not affect 0.7)

bow: [(20, 1), (187, 1), (188, 1), (189, 1), (190, 1)]

Оскільки проект пов'язаний з розробкою, то зустрічається багато специфічних слів або посилань, або назв класів. Інколи це може допомогти, але якщо назва класу зустрічається лише декілька разів, то краще позбутися їх перед побудовою моделі.

Встановивши достатньо м’які обмеження (слово повинно зустрічатися хоча би у 7 коментарях, тобто середня довжина обговорення задачі, та не більше ніж у 70% від загальної кількості) вдалося зменшити розмір словника з 42869 слів до 7939.

Далі для побудови тематичної моделі треба обрати оптимальну кількість тем. Згідно з розіділом 2.4 були пораховані кофіцієнти model_list, cv_coherence_values, perplexity_values, umass_coherence_values, bounds_values для кількості тем в інтервалі (100, 1000) з кроком 100

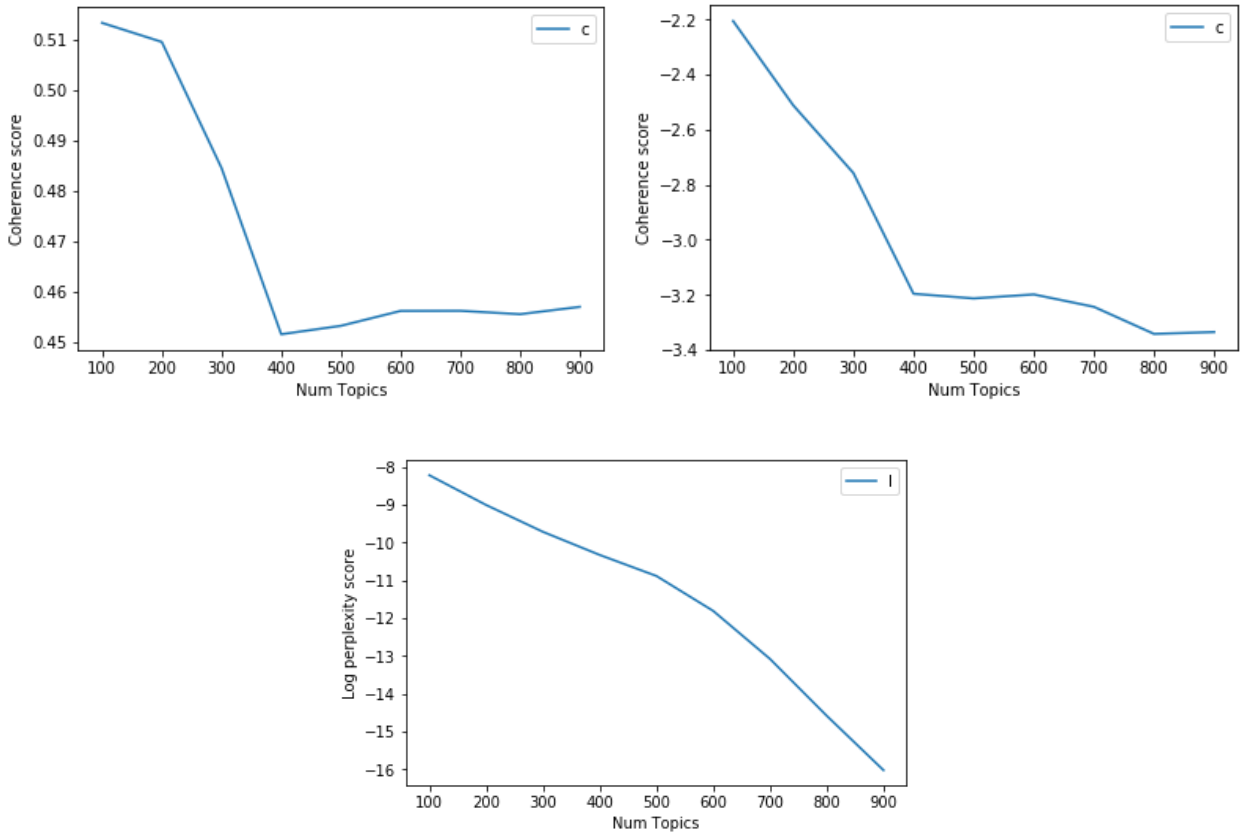


Рис 3.7 – Значення C_v , C_{UMass} та perplexity для кількості тем в інтервалі (100, 1000) з кроком 100

Як бачимо, оцінки стабільно падають, тож оптимальна кількість тем менше 100, а більше 400 немає сенсу розглядати. Повторимо процедуру але вже в інтервалі (10, 100) з кроком 10.

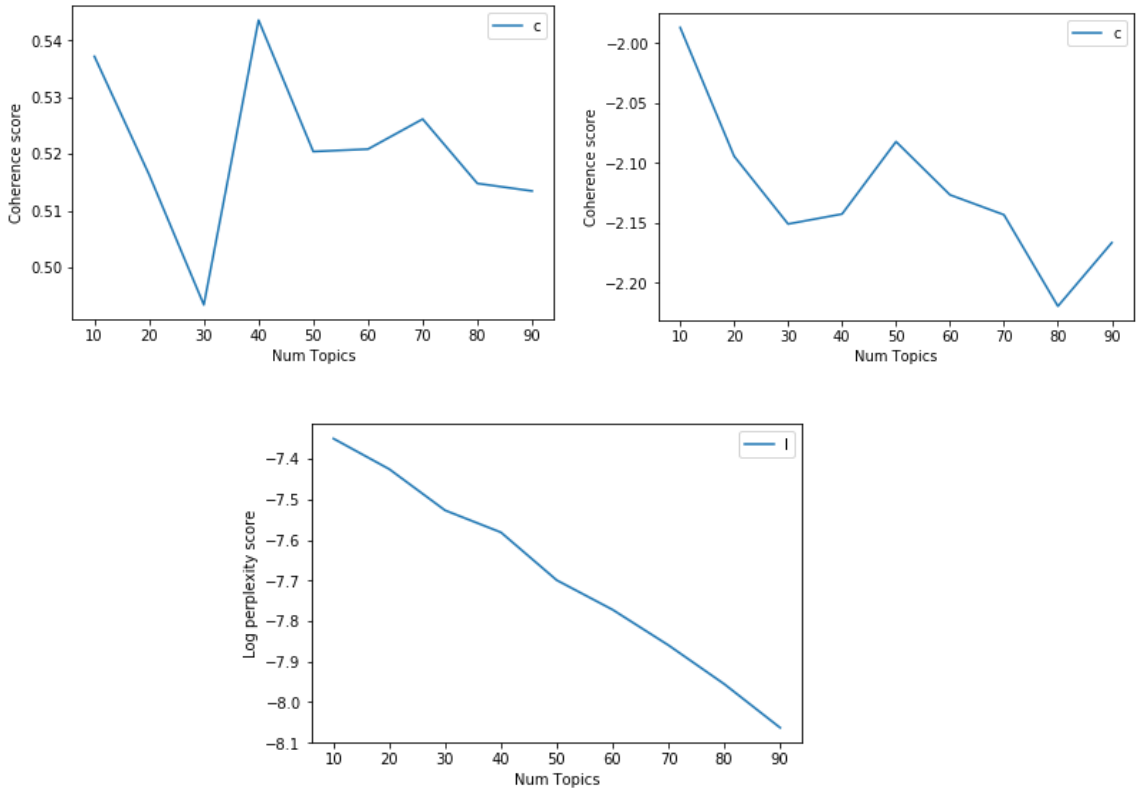


Рис 3.8 – Значення C_v , C_{UMass} та perplexity для кількості тем в інтервалі (10, 100) з кроком 10

Отже, за графіками отримаємо оптимальне значення кількості тем - 40.

Далі, щоб визначити тему коментаря, отримаємо для нього розподіл по темам. Наприклад, для коментаря “moves strategy creation into Table instantiation so it can't be out of sync”, маємо теми: [(0, 0.010958182), (4, 0.5013202), (11, 0.16128284), (25, 0.20744711)].

Для кожної теми маємо набір слів з вагами, завдяки яким можна вибрати ключові слова. Якщо слово несуттєве, то воно буде мати маленьку вагу в будь-якій темі, тобто буде вважатися стоп-словом, якщо є група документів, для яких це слово істотне, то значення ваги буде велике. Оскільки словник достатньо великий, краще зробити зберегти коєфіцієнти лише для перетин слів

коментаря і тем. Якщо слово зустрічається у декількох темах, то рахуємо зважену суму.

Наприклад:

Коментар: moves strategy creation into Table instantiation so it can't be out of sync

Теми: [(0, 0.010958182), (4, 0.5013202), (11, 0.16128284), (25, 0.20744711)]

Тема 4: '0.018*"thrift" + 0.017*"table" + 0.016*"make" + 0.015*"change" + 0.014*"cql_num_" + 0.014*"use" + 0.013*"patch" + 0.013*"bq" + 0.012*"would" + 0.012*"think"'

Тема 25: '0.036*"flush" + 0.017*"write" + 0.017*"call" + 0.012*"memtable" + 0.012*"segment" + 0.011*"thread" + 0.011*"replay" + 0.011*"get" + 0.010*"new" + 0.009*"need"'

Тема 11: '0.043*"cql" + 0.028*"id" + 0.020*"select" + 0.020*"_num_e_num_" + 0.016*"make" + 0.016*"would" + 0.010*"loop" + 0.009*"pprop" + 0.009*"eentid" + 0.009*"python"'

Тема 0: '0.075*"patch" + 0.040*"fix" + 0.039*"cassandra_num_" + 0.020*"trunk" + 0.020*"attach" + 0.017*"v_num_" + 0.015*"apply" + 0.013*"change" + 0.011*"version" + 0.011*"issue"'

За замовчування, модель видає лише топ-10 слів для кожної теми, в такому випадку для коментаря маємо лише:

{'table': 0.0082770735}

Якщо розришиті до топ-100 слів, то отримаємо:

Тема 0: {}

Тема 4: {'table': 0.008272701, 'strategy': 0.003277583}

Тема 11: {'instantiation': 0.0003548313}

Тема 25: {'move': 0.00050161354, 'sync': 0.0008123086}

Та отримаємо наступну хмару слів.

Рис 3.9 – Хмара слів для коментаря з прикладу

Ідея визначення теми обговорень задачі полягає в тому, щоб визначити ключові слова кожного коментаря та скласти їх ваги. Таким чином, якщо деякі слова притаманні майже всій нитці розмови, то вони отримують найбільші значення, якщо якіїсь слова властиві лише одному коментарю, то їх ваги стануть несуттєвими на фоні суми інших.

Наприклад для задачі 334211:

Коментар 1: moves strategy creation into Table instantiation so it can't be out of sync

Ключові слова та ваги:

{'table': 0.008272392,
'move': 0.00050160155,
'sync': 0.0008122892,
'strategy': 0.0032774606,
'instantiation': 0.00035488367}

Коментар 2: An addition to test/system/test_thrift_server.py which makes sure queries to the system keyspace can be made

Ключові слова та ваги:

{'sure': 0.00025436038,
'test': 0.0010942077,

'make': 0.0005898921,

'system': 0.029845346,

'keyspace': 0.001419298,

'query': 0.0017181913}

Коментар 3: test passes both with and without this patch, so the problem must be subtle, but I still think this patch has a good chance of stopping it. committed.

Ключові слова та ваги:

{'commit': 0.019157412,

'still': 0.0006146954,

'patch': 0.0044995123,

'think': 0.00043833553,

'test': 0.017650967,

'good': 0.0010508497,

'problem': 0.0004379859,

'pass': 0.003106538,

'without': 0.00032517817,

'stop': 0.0007060827,}

В цілому для задачі отримаємо хмару слів для задачі:

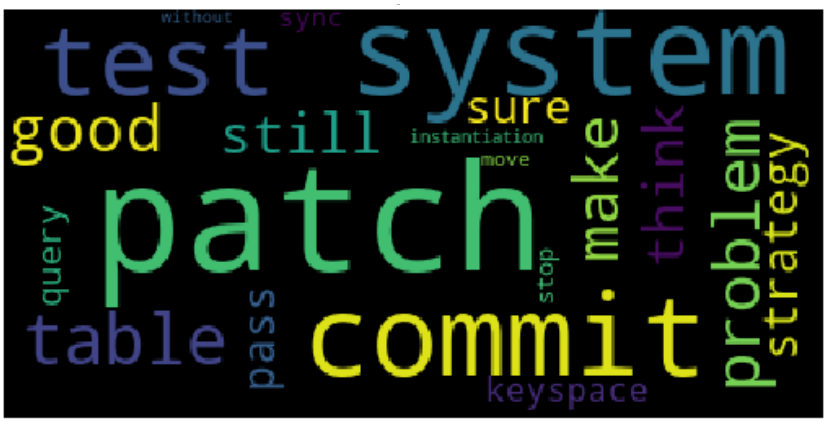


Рис 3.10 – Хмара слів для обраної задачі

3.6 Аналіз ризиків проекту

В результаті для кожної задачі маємо:

- Вірогідність як зважена сума ВЗД коментарів
- Значущість, що визначена менеджером проекту, у даному випадку поле *priority* з таблиці *jira_issue_report*
- Ключові слова та ваги, з яких можна отримати хмару слів

Id	integral_value	integral_value_weighed	likelihood	priority	priority_value	rate
333428	11.390000	11.390000	5	Blocker	5	25
329678	11.130000	11.130000	5	Blocker	5	25
333669	10.755556	10.737776	5	Blocker	5	25
331283	10.720000	10.720000	5	Blocker	5	25
329510	10.184286	10.184142	5	Blocker	5	25
333505	11.730000	11.730000	5	Critical	4	20
333436	11.518667	11.514989	5	Critical	4	20
330706	10.750000	10.750000	5	Critical	4	20

Рис 3.11 – Кінцевий показник та вірогідність для задач

З наступних рисунків бачимо приклади коментарів з найвискоми та найнижчими оцінками. Візьмемо перший же запис - задача 333428, вона має лише один коментар наступного змісту зі значенням ВЗД 4.62, 3.26, 3.51:

default_validation_class means "all data that isn't explicitly in column_metadata conforms to this data type." So you've violated that. You have two options:

- set d_v_c to BytesType (the default)

- leave the column definition alone, but only drop the index part (maybe this is what you were trying to do, but you changed from "colour" to "color")

More generally, note that best practice is to only use d_v_c in CFs with dynamic column names. I.e., if you know what the columns are going to be in the CF ahead of time as you do here, you shouldn't use d_v_c.

Автор надав розгорнуту відповідь, але з контексту не зрозуміло, вирішує цю проблему чи ні. Особливо беручи до уваги речення “So you've violated that.” (Так що ви порушили це.), задача вимагає хоча б уваги, тож отримане маркування має сенс. Розглянемо ще декілька прикладів.

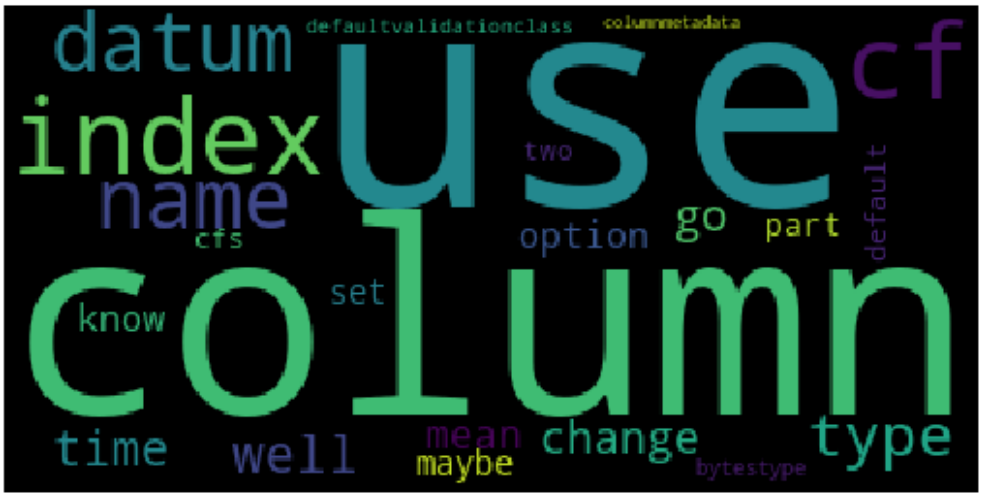


Рис 3.12 – Кінцевий показник та вірогідність для задач

Розглянемо коментарі задачі 329510. Їх досить багато та лише з наявності логів помилки можна зрозуміти, що проблема є та вона потребує вирішення.

Таблиця 3.1 – Коментарі задачі 329510

Коментар	ВЗД	∑
Those fail for me too, but that should be an easy bisect.	[5.14, 1.68, 2.05]	8.87
Looking at TokenMedataTest, it was just assuming the last test in the file was actually running last and apparently that wasn't happening on my box. Ninja-fixed that one in commit 0a5a766 to not depend on the tests execution order.	[4.13, 3.33, 3.45]	10.91
Looks pretty similar to what I did in 7de6f9666 to fix them in 1.1, except I used the	[4.2, 3.12,	9.12

shotgun method :) So if that fixes it, then +1	1.8]	

Продовження таблиці 3.1

Alright, ScrubTest was another instance of tests expecting to run in a particular order (don't know why my box don't run them in the order they are declared but well, expecting a particular order is a bad idea in any case) so ninja-fixed that. I've also committed the 2 patches attached above. This fixes the failure I'm saying, except for the pig tests, but as those are clearly a setup thing I don't want to block 1.2.12 for that and I've open CASSANDRA-6376 to deal with them. Closing this issue.	[5.74, 2.94, 3.97]	12.65
Btw, also got all pig tests to fail with the following exceptions: {noformat} [junit] Testcase: org.apache.cassandra.pig.CqlTableDataTypeTest: Caused an ERROR [junit] <error log> I wouldn't block a release because of pig tests, but if it does not just fail for me, it would be nice to fix it too.	[4.62, 1.97, 2.66]	9.25
For LeaveAndBootstrapTest, this bisects to CASSANDRA-6244. So I think this is just a case of "we've made things asynchronous so we now check the expected result before the computation is done". Tried adding a few calls to PRCS.blockUntilFinished in the few places that were failing for me and that seems to fix the test. Attaching the resulting patch. [~brandon.williams] can you check it's not entirely stupid?	[4.35, 2.66, 2.76]	9.77
Regarding the ConcurrentModificationException, it seems that the only reason this could get triggered is due to TMD.clearUnsafe(). As this is called by tests, this is not a real problem, but what about making it grab the writeLock like any good citizen to avoid getting scarry stack traces (and don't discard a real bug later on because we've grown used to discarding such stack)? Attaching patch to do that.	[4.57, 3.46, 2.69]	10.72

Читати всі коментарі достатньо важко, особливо, якщо брати до уваги, що вони суто технічні та наповнені термінологією. Але завдяки інструменту побудови назви ризику, можно швидко зрозуміти, що проблеми пов'язані помилками з junit, patch та якоюсь іншою задачею проекту (рис . ..).

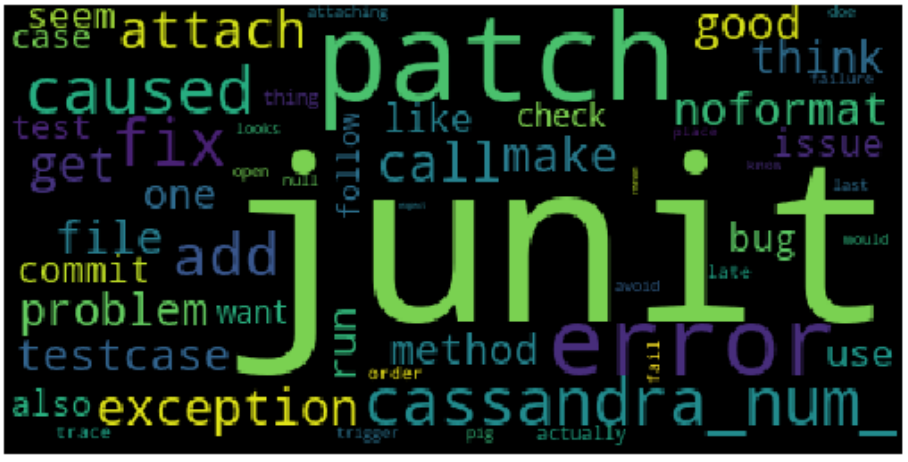


Рис 3.13 – Кінцевий показник та вірогідність для задач

Розглянемо декілька задач з кінця списку, тобто з важливістю - 1 (табл. 3.2). З тексту коментарів бачимо, що задачі дійсно не потребують додаткової уваги.

Таблиця 3.2 – Коментарі задач з низькою важливістю

задача	коментар	ВЗД	Σ
331618	bah, just realised you can use comparator= 'CompositeType(UTF8Type, UTF8Type)'	[0.116, 0.021, 0.185]	0.322
332691	duplicate of CASSANDRA-3164	[0.364, 0.289, 0.315]	0.968
330393	Resolving now that it's in trunk.	[0.044, 0.701, 0.275]	1.02
333721	done as part of CASSANDRA-2521	[0.294, 0.851, 0.025]	1.17
329561	{{ECHO OFF}}	[0.136, 0.601, 0.595]	1.33
	(this should be ninja-d)	[0.076, 1.399, 0.045]	1.52

Розглянемо, як розподілені задачі залежно від важливості. Як бачимо на рис. 3.6.4 коментарів, які дійсно потребують уваги небагато, тож це важливо відокремити їх від інших, щоб реагувати більш миттєво.

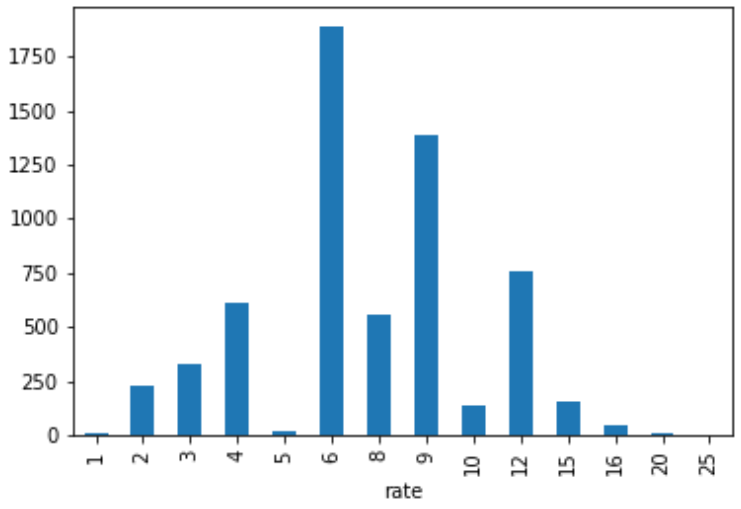


Рис 3.14 – Кінцевий показник та вірогідність для задач

Наостанок, розглянемо приклад задачі 334028 з найбільш частотною важливістю ризику - 6, чиї коментарі наведені у таблиці 3.6.1. Як бачимо зі змісту коментарів проблема була, але вона успішно вирішена, тож більше не потребує пильної уваги. Однак, перші два коментарі мають велике значення інтегрованого показника, тож на той момент задача мала би ризик великої вірогідності, а отже й важливості.

Таблиця 3.3 – Коментарі задачі 334028

коментар	ВЗД	∑
I did a bit more tests and here are some results which might help: 1. JMX port set to 9090 in cassandra-env.sh 2. On the machine where another service running on 8080 we get exception above 3. On the machine where no service running on 8080 we don't get any exception and MX4J runs on port 9090 Seems like something checks for port 8080 even though it is configured to run on 9090.	[3.92, 2.29, 2.83]	9.04
I think there's a confusion. There are two ports in business, one is the JMX port (default is 8080) and one is the MX4J port (default 8081) If the JMX port is used when cassandra starts you see the following exception, which is different from what's pasted in this bug report: <error log> So the problem in this case. I believe was that mx4j's port was bound to a different process. To control the port used by mx4j use -Dmx4jport=8082. See https://issues.apache.org/jira/browse/CASSANDRA-1068 for more details. I think this is not a bug and recommend to close it as such. I will, however, attach a patch for trunk to make this more obvious and add -Dmx4jport=8081 to conf/cassandra-env.sh	[5.05, 4.17, 4.06]	13.28
Patch that adds the variables MX4J_ADDRESS and MX4J_PORT to conf/cassandra-env.sh make configuration for mx4j obvious.	[1.65, 1.371, 2.39]	5.411
You right Ran, I checked this machine again and I have another service listening on 8081. For some reason I thought that MX4J uses same port. With config options we can close it now.	[2.32, 2.05, 2.49]	6.86

committed, thanks!	[2.706, 0.881, 1.705]	5.292

Якщо побудуємо хмару слів, то серед найважливіших побачимо “thank” (дякую), то також може бути натяком на розв'язану проблему.

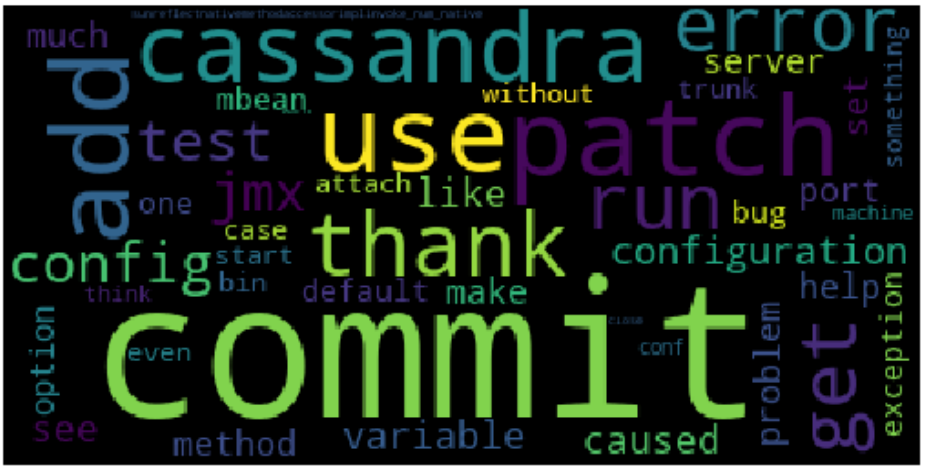


Рис 3.15 – Кінцевий показник та вірогідність для задач

Висновки до розділу 3

В даному розділі аргументовано використання обраного програмного забезпечення, опис використовуваних даних, а також надані деталі реалізації вилучення емоційних складових з тексту, деталізований процес з прикладами щодо переходу від цих складових до показників ризику (ймовірність та важливість) та методи перетворення їх у матрицю ризиків.

Також було описано процес визначення тематики коментарів та використання отриманої моделі для визначення назви ризику, що, як показано

було на прикладах, значно спрощує розуміння дискусії, особливо коли коментарів багато та вони насичені технічними термінами.

В якості демонстраційного прикладу був обран проект CASSANDRA компанії Apache Software Foundation, та базуючись на отриманих показниках ймовірності й важливості ризиків задач та побудованих темах було розглянуто декілька різноманітних прикладів, які підтвердили адекватність запропонованої методології.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї стартап-проекту

В межах даного підрозділу послідовно проаналізовано та подано у вигляді таблиць наступні пункти:

- зміст ідеї;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямом застосування);
- чим відрізняється від існуючих аналогів та замінників;

Перші три пункти подано у вигляді таблиці (Таблиця 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 – Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Надання додатка для управління ризиками проекту з автоматичною ідентифікацією та аналізом можливих ризиків.	Ранжування задач за важливістю можливих ризиків	Швидке розподілення пріоритетів усередині проекту
	Отримання інформації про потенційні ризики конкретної задачі	Стислий опис задачі та коментарів

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів, проектів-конкурентів, товарів-замінників чи товарів-аналогів, що вже існують на ринку;
- збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів.

Відповідно до визначеного вище переліку проводиться порівняльний аналіз показників: гірші значення (W, слабкі); аналогічні (N, нейтральні) значення; кращі значення (S, сильні).

Визначення сильних, слабких та нейтральних характеристик ідеї стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.2.

Таблиця 4.2 – Сильні, слабкі та нейтральні характеристики ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				Х-ка
		Мій проект	Конк. 1	Конк 2	Конк 3	
1	Форма виконання	Додаток jira	Додаток jira	Додаток jira	Додаток jira	N
2	Собівартість	Низька	Низька	Низька	Низька	N
3	Точність результатів	Велика	Середня	Низьа	Низька	S
4	Наявність інтернету	Так	Так	Так	Так	N
5	Кросплатформеність	Так	Так	Так	Так	N
6	Складність використання/ автономність	Ні	Так	Ні	Так	S

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2 Технологічний аудит ідеї стартап-проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можливо реалізувати ідею проекту (технології створення товару).

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових:

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проекту?

Технологічну здійсненність ідеї стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї стартап-проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення додатку для marketplace Atlassian для керування ризиками	Java	Наявна	Безкоштовна, доступна
		R	Наявна	Безкоштовна, доступна
		Python Gensim	Наявна	Безкоштовна, доступна

Обрана технологія реалізації ідеї проекту: для створення додатку для marketplace Atlassian обрана технологія Python Gensim, яка є безкоштовною та якою володіють розробники.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (Таблиця 4.4).

В ході таких досліджень вивчаються особливості і перспективи розвитку попиту на конкретні товари, позиції конкурентів на ринку, їх сильні і слабкі сторони, динаміку цін і т.д. Стартап-проекту важливо знати, чи буде обсяг продажів його товарів достатнім для компенсації зусиль щодо виходу на ринок, тому важливою характеристикою ринку є його ємність, під якою розуміють максимально можливий обсяг продажу певного товару протягом року, виражений в натуральних і вартісних одиницях.

Попит на більшість товару, який визначає місткість ринку, характеризується нестабільністю. Тому кожне підприємство прагне мати достовірний прогноз попиту на свій товар. З метою стимулювання збільшення

попиту на товар необхідно вивчити і проаналізувати думки і потреби споживачів певного товару.

Попередню характеристику потенційного ринку стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	3
Загальний обсяг продаж, грн/ум.од	150000
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	Немає
Специфічні вимоги до стандартизації та сертифікації	Немає
Середня норма рентабельності в галузі (або по ринку), %	R=20%

Отже, проаналізовано наявність попиту, обсяг, динаміку розвитку ринку. Обмеження для входу на ринок відсутні, динаміка ринку зростає, галузь є рентабельною.

Далі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи.

Щоб краще оцінити доцільність розробки даного додатку та існуючий попит серед його потенційних користувачів, було проведено маркетингове дослідження. Анкету можна знайти у додатку Г, а результати дослідження - у додатку Ґ.

В опитуванні взяло участь 20 чоловік. На основі їх відповідей, можна зробити висновок, що велика частина компаній працює або в невеликих: (21-80 чоловік), або в досить великих (800+). Як правило, це продуктова компанія (50%) або outsource (30%).

80% користуються jira і практикують scrum. У більшості випадків оцінюється терміни виконання завдань (70%), але не завжди (57%) перевіряється їхня точність. Даний функціонал реалізований в стандартному тарифі jira.

75% респондентів заявили про відсутність готового списку ризиків. При цьому, майже одноголосно погодившись з необхідністю управління ризиками для успішного ведення проекту.

Найбільш неприємними проблемами в управлінні ризиками були названі:

- Складність охоплення всіх ризиків (81%)
- Витрачається час (31)
- Оформлення документації (31)

Те, що хотілося б змінити: "підхід до перевірки завдань на різних її стадіях".

У більшості випадків (56%) спілкування частішає ближче до дедлайнів. Але як правило коментарі недостатньо емоційні, хоча в них часто обговорюються поточні проблеми. Так що з відповідей видно, що колеги як правило ввічливі.

Отже можна зробити висновок, що:

- варто зробити базовий список ризиків для IT-проектів, для цього можна застосувати запропоновану методологію на історичних даних запропонованих проектів або проектів конкретної компанії, щоб отримати теми ризиків та потім відкоригувати їх для резерву для майбутніх проектів.

- при аналізі коментарів робити акцент не на емоційній складовій, а на ключових словах, які б характеризували б завдання / проект, що було вже запропоновано у даній роботі.
- базові шаблони для оформлення документації, автоматизація заповнення, що треба буде додати, але це вже більше задачі для розробників.

Таким чином нашим сегментом ринку будуть компанії:

Таблиця 4.5 – Потенційні сегменти споживачів

Характеристика	Сегмент 1	Сегмент 2	Сегмент 3	Сегмент 4
Тип компанії	product	outsource	product	outsource
Розмір	800+	800+	21-80	21-80
Основний інструмент	jira			
Методологія УП	scrum			
Сфера	IT			
Ємність	6 000	4 000	20 000	120 000

Характеристику потенційних клієнтів стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.6.

Таблиця 4.6. – Характеристика потенційних клієнтів стартап-проекту

№ п/п	**Потреба, що формує ринок**	**Цільова аудиторія (цільові сегменти ринку)**	**Відмінності у поведінці різних потенційних цільових груп клієнтів**	**Вимоги споживачі в до товару**
1	Витрачати менше часу на аналіз ризиків проекту	Аудиторія: менеджери ИТ-проектів. Сегменти: індивідуальні користувачі, маленькі підприємства, великі підприємства.	Для сегменту дрібних користувачів більш характерні додатки з базовим набором функцій. Великі підприємства зацікавлені у додатках з широким спектром функцій та високою якістю результатів, у постійному оновленні та підтримці.	Усім споживача м важлива можливість швидко ідентифікув ати ризики та витрачати менше часу на їх аналіз

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають. Фактори в таблицях подають в порядку зменшення значущості.

Ринкові можливості – це сприятливі обставини, які підприємство може використовувати для отримання переваг. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які можна використовувати.

Ринкові загрози – події, настання яких може несприятливо вплинути на підприємство.

Фактори загроз стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.7. Фактори можливостей стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.8.

Таблиця 4.7. - Фактори загроз

№ п/п	**Фактор**	**Зміст загрози**	**Можлива реакція компанії**
1	Зростаюча конкуренція	Зі зростанням попиту на розробку додатків для аналізу ризиків зросла і пропозиція.	Розробляти додаток високої якості та з додатковими унікальними функціями.
2	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створюваного ПЗ

Таблиця 4.8 – Фактори можливостей

№ п/п	**Фактор**	**Зміст можливості**	**Можлива реакція компанії**
1	Зростаючий попит	Збільшення попиту на додаток.	Надавати високоякісні рішення, займати нішу ринку.
2	Оптимізація швидкості завантаження	Оптимізація швидкості завантаження додатка.	Оптимізація швидкості завантаження за рахунок рефакторингу, асинхронності, мінімізації файлів кінцевого веб-застосування та оптимізації стиснення зображень.
3	Зниження довіри до конкурента 1	У ПЗ конкурента №1 нещодавно була знайдена помилка, завдяки якій дані сервісів клієнтів стали доступні в інтернеті для всіх користувачів	При виході на ринок звертати увагу покупців на безпеку нашого ПЗ та авторитетність компанії

Ступеневий аналіз конкуренції на ринку стартап-проекту “Додатку для marketplace Atlassian для керування ризиками з їх автоматичної ідентифікацією” наведено у Таблиці 4.9.

Таблиця 4.9 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства**
1. Вказати тип конкуренції - досконала	Існує 3 компанії-конкуренти на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
2. За рівнем конкурентної боротьби - міжнародний	Всі компанії з інших країн	Використовувати мови міжнародного користування
3. За галузевою ознакою - внутрішньогалузева	Конкуренти мають ПЗ, яке використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях та додавати нові модулі в існуюче
4. Конкуренція за видами товарів: - товарно-видова	Види товарів є однаковими, а саме – додаток marketplace Atlassian	Створити ПЗ, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПЗ, щоб собівартість була нижчою	Використання менш дорогих технологій для розробки, ніж використовують конкуренти
6. За інтенсивністю - марочна	Бренди присутні	-

Після аналізу конкуренції проведено більш детальний аналіз умов конкуренції в галузі (табл. 4.10).

Таблиця 4.10 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу		**Висновки**
Прямі конкуренти в галузі	Навести перелік прямих конкурентів	Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 3, так як його рішення також представлене у вигляді ПЗ.
Потенційні конкуренти	Визначити бар'єри входження в ринок	Так, можливості для входу на ринок є, бо наше рішення покращує та пришвидшує роботу спеціаліста
Постачальники	Визначити фактори сили постачальників	Постачальники відсутні.
Клієнти	Визначити фактори сили споживачів	Важливим для користувача є кросплатформеність ПЗ та якість його роботи.
Товари-замінники	Фактори загроз з боку замінників	Товари-замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару.

За результатами аналізу таблиці зроблено висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. Також зроблено висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку. Другий висновок враховується при формулюванні переліку факторів конкурентоспроможності. На основі аналізу конкуренції, проведеного в табл.4.10, а також із урахуванням

характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.6) та факторів маркетингового середовища (табл. 4.7 – табл. 4.8) визначено та обґрунтовано перелік факторів конкурентоспроможності. Аналіз оформляється за табл. 4.11.

Таблиця 4.11 - Обґрунтування факторів конкурентоспроможності

№	**Фактор конкуренто-спроможності**	**Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)**
1	Виконання ПЗ у вигляді додаток marketplace Atlassian	Це рішення дозволяє швидко встановлювати використовувати ПЗ всередині проекту
2	Простота інтерфейсу користувача	Інтерфейс користувача зроблений таким чином, що користувачу необхідно лише заповнити необхідні поля.
3	Наявність моделей ШІ	Це дозволить надати користувачеві інформацію, яка може спростити його работу

За визначеними факторами конкурентоспроможності (табл. 4.11) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.12).

Таблиця 4.12 - Порівняльний аналіз сильних та слабких сторін проекту

N	**Фактор конкуренто-спроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів у порівнянні з нашим підприємством**						
			-3	-2	-1	0	1	2	3
1	Автоматична ідентифікація ризиків	20			+				
2	Простота інтерфейсу користувача	15	+						

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.13) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.12). Перелік ринкових загроз та ринкових можливостей складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.13 - SWOT- аналіз стартап-проекту

Сильні сторони: простий інтерфейс користувача, інтеграція с jira,наявність списку можливих ризиків, аналіз великих даних	**Слабкі сторони:** доступно тільки англійською мовою, список можливих ризиків неспеціалізований, неточності у визначенні ключових слів
Можливості: зростання популярності управління ризиками серед маленьких компаній, розширення базового списку ризиків за рахунок зростання користувачів, поява нових моделей для обробки даних	**Загрози**: конкуренція, зміна потреб користувачів, поява подібного функціоналу в jira

На основі SWOT-аналізу розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл.4.10, аналіз потенційних конкурентів). Визначені альтернативи проаналізовано з точки зору строків та ймовірності отримання ресурсів (табл.4.14).

Таблиця 4.14 - Альтернативи ринкового впровадження стартап-проекту

№	**Альтернатива (орієнтовний комплекс заходів) ринкової поведінки**	**Ймовірність отримання ресурсів**	**Строки реалізації**
1	Створення додатку marketplace Atlassian за технологією Python Gensim для керування ризиками з їх автоматичної ідентифікацією та сторення попередньо визначеного реєстру потенційних ризиків	80%	7 місяці
2	Створення додатку marketplace Atlassian за технологією Python Gensim для керування ризиками з їх автоматичної ідентифікацією без реєстру	40%	6 місяці

Обираємо альтернативу 1.

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – не набагато більші. Враховуючи, що наявність веб-застосунку збільшить ймовірність отримання ресурсів, то обираємо перший варіант.

4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.15).

Таблиця 4.15 – Вибір цільових груп потенційних споживачів

№ п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Великі продуктові компанії.	Середня: велика конкуренція і можливість власних веб-відділів.	Високий.	Велика.	Легко.
2	Великі аутсорсові компанії	Середня.	Високий.	Велика.	Середня
3	Маленькі продуктові компанії.	Середня.	Середній.	Середня.	Середня.
4	Маленькі аутсорсові компанії	Низька. Приватні особи воліють продукт за найменшу ціну і не обов’язково якісний.	Низький.	Середня.	Важко.

Як цільові групи обрано усі три варіанти.

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку:

- якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;
- якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;
- якщо компанія працює з усім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг. Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (табл. 4.16)

Таблиця 4.16 - Визначення базової стратегії розвитку

№	**Обрана альтернатива розвитку проекту**	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку**
1	Створення додатку marketplace Atlassian за технологією Python Gensim для керування ризиками з їх автоматичної ідентифікацією	Ринкове позиціювання	Простота інтерфейсу, пришвидшення роботи, легкість вбудовуємості, можливість налаштування параметрів	Диференціації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.17).

Таблиця 4.17 - Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Так	Так: базові функції керування ризиками	Зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. Табл. 4.6), а також в залежності від обраної базової стратегії розвитку (табл. 4.16) та стратегії конкурентної поведінки (табл. 4.17) розробляється стратегія позиціонування (табл. 4.18). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.18 - Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Швидкість роботи, відповідність результатів	Диференціації	автоматична ідентифікація ризиків, реєстр попередньо визначених ризиків	Швидкість легкість, точність, великі дані, аналітика

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

4.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримує споживач. Для цього у табл. 4.19 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.19 - Визначення ключових переваг концепції потенційного товару

№	**Потреба**	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити**
1	Витрачати менше часу на визначення ризиків	Автоматична ідентифікація ризиків з коментарів	Економія часу та зусиль
2	Можливість передбачити більше ризиків	Наявність попередньо визначених ризиків	Користувач може лише обрати релевантні до проекту ризики

Розроблена трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.20).

Таблиця 4.20 - Опис трьох рівнів моделі товару

Рівні товару	**Сутність та складові**		
I. Товар за задумом	Товар допомагає користувачам автоматично ідентифікувати та аналізувати ризики задач проекту та ранжувати їх. В результаті це дозволяє пришвидшити процес виявлення та розуміння існуючих проблем та пошук рішення.		
II. Товар у реальному виконанні	Хар-ки	М/Нм	Вр/Тх /Тл/Е/Ор
	1) додаток marketplace Atlassian; 2) Простота у використанні; 3) Можливість розширення	-	-
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє.		
III. Товар із підкріпленням	Безкоштовна версія з урізаним функціоналом		
	Постійна підтримка для користувачів		

1-й рівень - При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням

технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень - Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень - Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін)

За рахунок чого потенційний товар буде захищено від копіювання: ноу-хау.

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару. Наступним кроком визначено цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 4.21). Аналіз проводився експертним методом.

Таблиця 4.21 - Визначення меж встановлення ціни

№	**Рівень цін на товари-замінники**	**Рівень цін на товари-аналоги**	**Рівень доходів цільової групи споживачів**	**Верхня та нижня межі встановлення ціни на товар/послугу**
1	1000	1500	200000	500

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.22):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 4.22 - Формування системи збуту

№	**Специфіка закупівельної поведінки цільових клієнтів**	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна система збуту**
1	Купують ПЗ та роблять щорічні внески для подовження ліцензії	Продаж	1(через посередника)	Власна та через посередників

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.23).

Таблиця 4.23 - Концепція маркетингових комунікацій

№	**Специфіка поведінки цільових клієнтів**	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
1	Купівля	Інтернет	автоматична	Показати	Демо-ролик

	ліцензій на використання через інтернет повної версії		ідентифікація ризиків, реєстр ризиків	переваги ПЗ, у тому числі і перед конкурентами	із використанням

Результатом пункту 5 є ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки

Висновок до розділу 4

Згідно до проведених досліджень:

- існує можливість ринкової комерціалізації проекту;
- існують перспективи впровадження з огляду на потенційні групи клієнтів, бар’єри входження високі, але проект має одну значну перевагу перед конкурентами;
- необхідно реалізувати додатку marketplace Atlassian за технологією Python Gensim для керування ризиками з їх автоматичної ідентифікацією для доступу;
- подальша імплементація є доцільною.

ВИСНОВКИ ПО РОБОТІ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

В ході роботи було вивчено джерела для визначення актуального напряму розвитку роботи та методів, що використовуються в даній сфері. На основі отриманої інформації був створен огляд літератури, складена методологія роботи, розроблені інструменти для її використання і проведені експерименти для перевірки адекватності методу.

Для експерименту використовувалися коментарі відкритого jira-сховища Apache, потім вони токенізіровалісь, лематизувалися і для кожного з них обчислювалися характеристики VAD. Імовірність появи ризиків в даній задачі розраховуються як середнє інтегроване взажене значення цих показників, де вагами виступає актуальність коментаря. В результаті коментарі з більш яскравим емоційним забарвленням дійсно мають великі показники, що може служити сигналом для менеджерів, що варто передчасно звернути увагу на дане завдання.

У подальшому можно розглянути та застосувати інші емоційні фреймворки та показники, наприклад, визначати рівень ввічливості або агресивності. Також, можна застосувати сентиментальний аналіз та використати його оцінку як окремий вимір.

С точки зору семантики покращити результати можна за рахунок врахування особливостей сучасної комунікації, а саме: сленг, смайли, які можуть знижувати або підвищувати показник ризику, позначення типу “+1” повинні мати позитивні властивості. З точки зору структури текстів, варто дослідити вплив довжини коментарю на значення результуючого показника.

Далі був запропонований підхід для визначення назви потенційних ризиків задачі з її коментарів на основі побудови моделі LDA та використання отриманих коефіцієнтів для побудови хмари слів. У подальшому варто зробити акцент на автоматизацію обрання оптимальних параметрів (кількість тем та слів, що формують тему), які можуть змінюватися від проекту до проекту.

Оскільки назва ризику задачі формується зі слів, що наявні в коментарях, маємо релевантні результати, але тим не менш можна дослідити як буде змінюватися відображення назви зі зміною кількості тем та топ-N для врахування.

Якщо дані інструменти будуть використовуватися в реальних проектах, треба подумати про автоматизацію обрання оптимальних параметрів за замовчуванням. З іншого боку вони можуть встановлюватися менеджером у налаштуваннях, так сама як періодичність оновлення моделі новими коментарями.

У подальшому, якщо надати користувачам інструмент для перейменування ризиків з ключових слів, можна використовувати моделі для генерування більш природних назв.

Також варто обрати до уваги, що одна задача може містити у собі декілька ризиків, здатність виокремити їх з розмови розробників - інше складне завдання, що потребує дослідження.

Наостанок, зроблено маркетинговий аналіз потенційного продукту, що може бути створений на основі запропонованої методології. З результатів опитування можна зробити висновок, що майбутній додаток має містити в собі базовий функціонал та реєстр ризиків, що може зробити його більш привабливим для можливих користувачів. Чим більше буде користувачів та даних, тим більш краще уявлення про ризики у розробці програмних продуктів можна отримати, що відкриває перспективи для більш швидкого визначення

ризиків та можливих їх розв’язків. Застосування аналізу часових рядів для обраних показників може допомогти також і передбачати проблеми за змінами у листуванні, а можливо й лише з назви задачі.

ПЕРЕЛІК ПОСИЛАНЬ

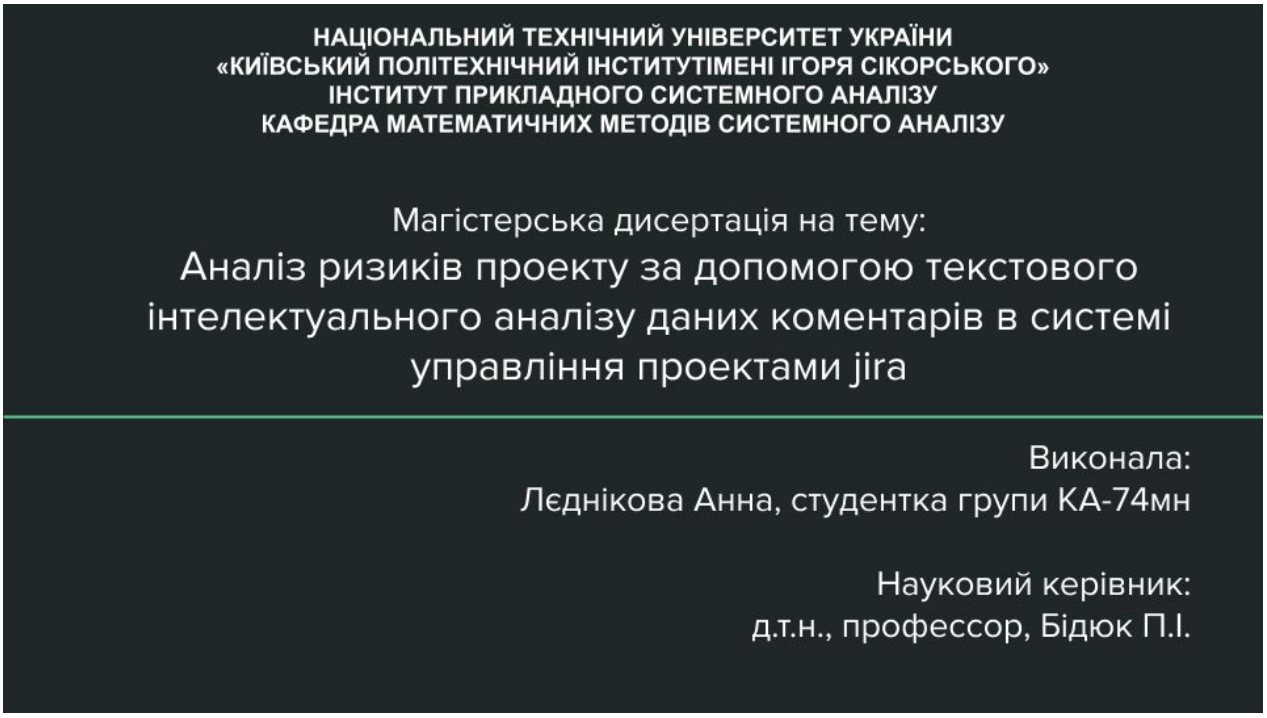
1. Success Rates Rise. Transforming the high cost of low performance., 2017. – (Project Management Institute). – (PMI’s Pulse of the Profession). – Режим доступу: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf . – Дата доступу : 13.05.2019.
2. Fitsilis P. Comparing PMBOK and Agile Project Management software development processes / Fitsilis. // Advances in Computer and Information Sciences and Engineering. – 2008. – pp. 378–383.
3. Using the Affect Grid to Measure Emotions in Software Requirements Engineering / R.Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, A. García-Crespo. // Journal of Universal Computer Science. – 2011. – №17. – pp. 1281–1298.
4. Lin J. Human Factors in Agile Software Development. [Електронний ресурс] / Jun Lin // arXiv. – 2015. – Режим доступу до ресурсу: https://arxiv.org/abs/1502.04170.
5. Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts. / A.Murgia, P. Tourani, B. Adams, M. Ortu. // Proceedings of the 11th Working Conference on Mining Software Repositories. – 2014. – pp. 262–271.
6. Teamwork [Електронний ресурс] // Atlassian. – 2018. – Режим доступу до ресурсу: https://www.atlassian.com/teamwork.
7. Project Management Software [Електронний ресурс] // 2018 – Режим доступу до ресурсу: https://www.capterra.com/project-management-software/#infographic.
8. Ortu, M., Destefanis, G., Adams, B., Murgia, A., Marchesi, M., & Tonelli, R. (2015). The JIRA Repository Dataset: Understanding Social Aspects of Software Development. Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering
9. On negative results when using sentiment analysis tools for software engineering research / R.Jongeling, P. Sarkar, S. Datta, A. Serebrenik. – 2017. – Empirical Software Engineering.
10. Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity? / [M. Mäntylä, A. Adams, G. Destefanis та ін.]. //

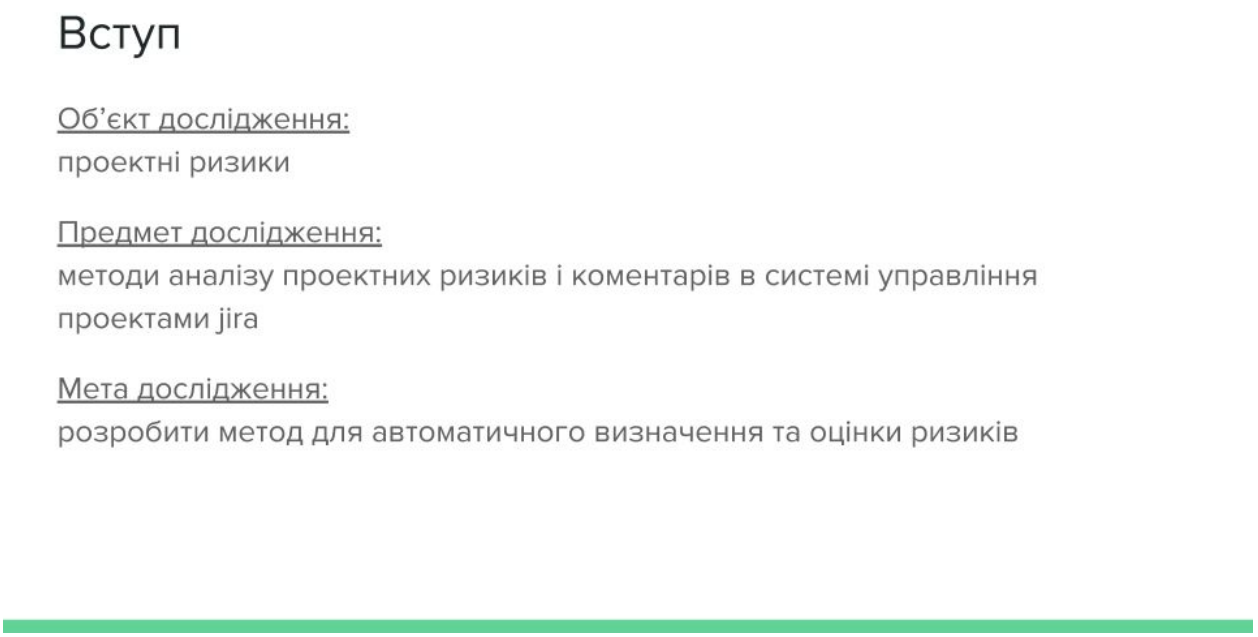
Proceedings of the 13th International Conference on Mining Software Repositories. – 2016. – pp. 247–258.

11. Arsonists or Firefighters? Affectiveness in Agile Software Development / [M. Ortu, G. Destefanis, S. Counsell та ін.] // Agile Processes, in Software Engineering, and Extreme Programming / [M. Ortu, G. Destefanis, S. Counsell та ін.]. – Cham: Springer, 2016. – (Lecture Notes in Business Information Processing). – (XP 2016; вип. 251).
12. Warriner A. B. Norms of valence, arousal, and dominance for 13,915 English lemmas / A. B. Warriner, V. Kuperman, M. Brysbaert. // Behavior Research Methods. – 2013. – №45. – pp. 1191–1207.
13. Guzman E. Visualizing emotions in software development projects / Guzman. // 1st IEEE Working Conference on Software Visualization - Proceedings of VISSOFT 2013. – 2013.
14. Decision Support System for Risk Assessment and Management Strategies in Distributed Software Development / [A. Aslam, N. Ahmad, T. Saba та ін.]. // IEEE Access. – 2017. – №5. – С. 20349–20373.
15. Leavitts H. Applied organisation change in industry: Structural, technical and human approaches / Leavitts., 1964. – (Handbook of Organizations). – (Handbook of Organizations).
16. System Dashboard [Електронний ресурс] // Apache Software Foundation. – 2018. – Режим доступу до ресурсу: https://issues.apache.org/jira/secure/Dashboard.jspa.
17. Hofmann T. Probabilistic latent semantic indexing / Hofmann. // Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. – 1999. – pp. 50–57.
18. Blei D. M. Latent Dirichlet Allocation / D. M. Blei, A. Y. Ng, M. I. Jordan. // Journal of Machine Learning Research. – 2003. – №3. – С. pp. 993–1022.
19. Стаття SoftwarePlant Team «Risk Management with BigPicture plugin for JIRA». – Режим доступу: https://softwareplant.com/risk-management-bigpicture-plugin-jira/ . – Дата доступу : 13.05.2019.

20. Стаття «Matrix for Jira» – Режим доступу: https://documentation.catworkx.com/matrix/latest/use-cases . – Дата доступу : 13.05.2019.
21. Hoffman M. D. Online Learning for Latent Dirichlet Allocation / M. D. Hoffman, D. M. Blei, F. Bach. // Advances in Neural Information Processing Systems. – 2010. – №23. – С. 856–864.
22. Röder M. Exploring the Space of Topic Coherence Measures / M. Röder, A. Both, A. Hinneburg. // In Proceedings of the eighth International Conference on Web Search and Data Mining. – 2015.
23. Optimizing semantic coherence in topicmodels / [D. Mimno, H. M. Wallach, E. Talley та ін.]. // InProc. of the Conf. on Empirical Methods in Natural Language Processing. – 2011. – pp 262–272.
24. Репозиторій з датасетом. – Режим доступу: https://github.com/marcoortu/jira-social-repository . – Дата доступу : 13.05.2019.

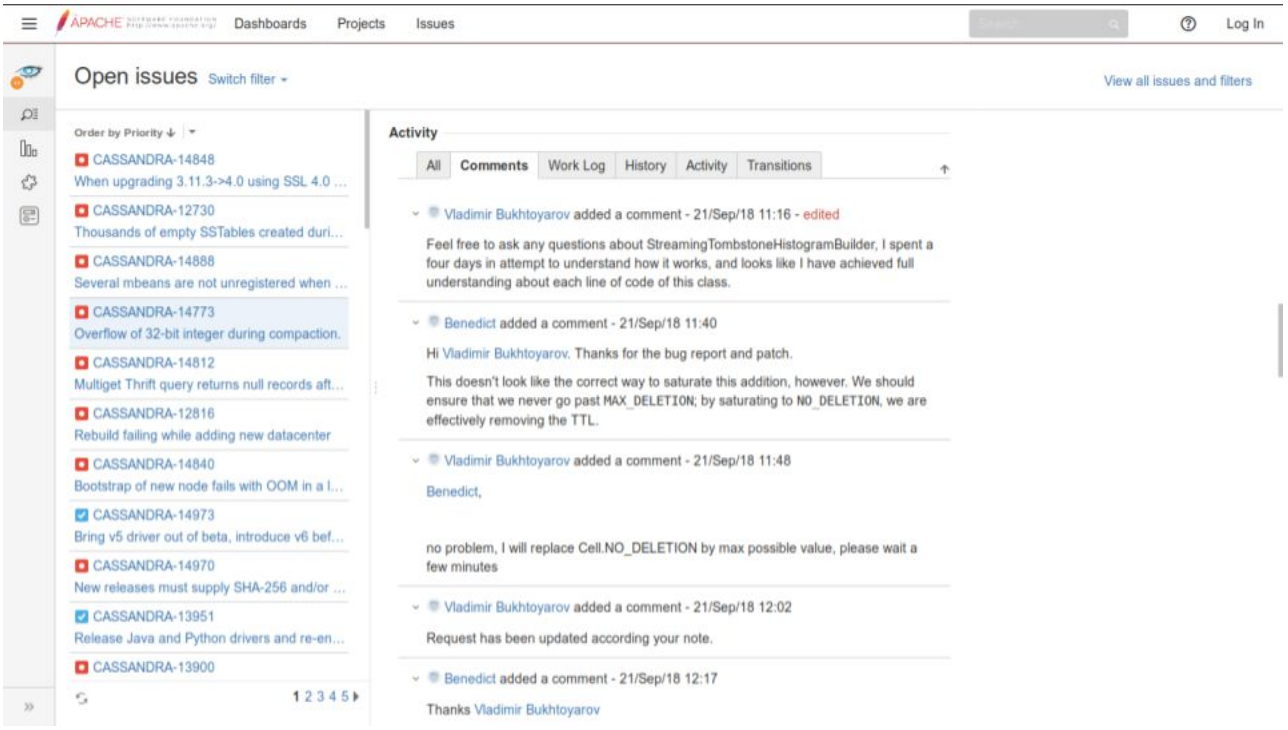
ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ





Актуальність роботи

- Компанії хутко ростуть та все більше стають децентралізованими
- Чим більше компанія та команда, тим більша потреба у її менеджменті
- Аналіз ризиків - це одна з найбільш важливих та трудомістких частин управління
- Jira - найпопулярніший інструмент управління IT проектами (50+M)
- Інженерія ПЗ є інтенсивною діяльністю людського капіталу
- Людські та соціальні фактори мають дуже сильний вплив на успіх розробки програмного забезпечення та остаточної системи
- Чим більше методологія приймає до уваги людські фактори, тим успішніше вона стає в реальному світі.



Постановка задачі

В наявності:

- датасет jira_emotion спільноти Apache Software Foundation, Spring, JBoss та CodeHaus (1K проектів, >700 тис. звітів і >2 млн. коментарів)

Треба:

- добути емоційні показники
- перетворити емоційні складові у ймовірність ризику задачі
- визначати назву ризику задачі

Методологія

Загальний план

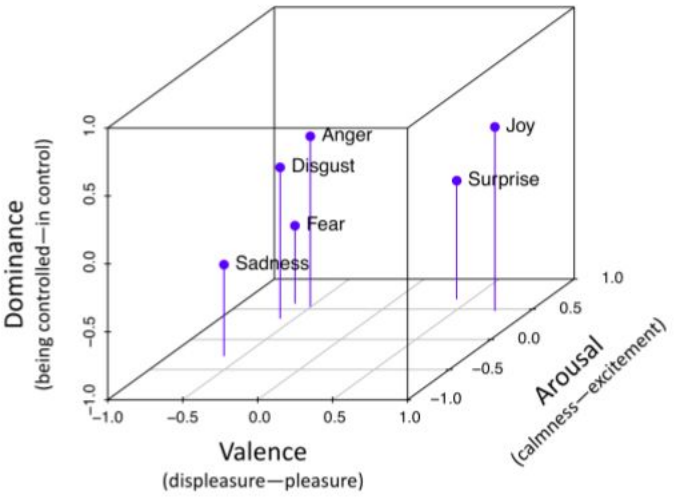
1. Визначити ймовірність ризиків задачі
    a. Для кожного коментаря порахувати ВЗД та актуальність
    b. Агрегувати ці значення в цілому для задачі
2. Визначити пріоритет задачі з датасету
3. Визначити назву ризиків задачі
    a. Побудовати тематичну модель
    b. Визначити ключові слова та ваги для кожного коментаря
    c. Агрегувати ці значення в цілому для задачі

Емоційні показники

Валентність (Valence) - це емоційний вимір, пов'язаний з привабливістю (або несприятливістю) події, об'єкта або ситуації.

Збудження (Arousal) - це розмірність, що представляє рівень емоційної активації.

Домінантність (Dominance) являє собою зміну відчуття контролю над стимулом (або ситуацією).



Визначення емоційних показників

- Показники валентності, збудження і домінування (ВЗД) для слова визначаються за таблицею оцінок Warriner з 13 915 англійських слів
- Для коментаря визначаємо показники за формулою:

Слово	Валентність	Збудження	Домінування
Anger / гнів	2.50	5.93	5.14
Joy / радість	8.21	5.55	7.00
Sadness / смуток	2.40	2.81	3.84
Love / кохання	8.00	5.36	5.92
Середнє	5.06	4.21	5.19

$$Range(\bar{w}) = \begin{cases} max(\bar{w}) - avg(\bar{W}),\ if\ min(\bar{w}) > avg(\bar{W}) \\ avg(\bar{W}) - min(\bar{w}),\ if\ max(\bar{w}) < avg(\bar{W}) \\ max(\bar{w}) - min(\bar{w}),\ if\ min(\bar{w}) \leqslant avg(\bar{W}) \leqslant max(\bar{w}) \end{cases}$$

Матриця ризиків

CONSEQUENCE					
LIKELIHOOD*	Insignificant 1	Minor 2	Moderate 3	Major 4	Catastrophe 5
A (Almost certain)	H	H	E	E	E
B (Likely)	M	H	H	E	E
C (Possible)	L	M	H	E	E
D (Unlikely)	L	L	M	H	E
E (Rare)	L	L	M	H	H

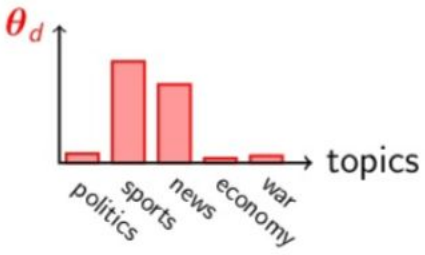
Рівень	Значення	Опис	ВЗД
A	5	Безумовно	>=10
B	4	Імовірно	(10, 8]
C	3	Можливо	(7, 5]
D	2	Навряд чи	(5, 2]
E	1	Рідко	<2

Рівень	Опис	тип Jira
1	Мізерні	Trivial
2	Незначні	Minor
3	Помірні	Major
4	Значні	Critical
5	Катастрофа	Blocker

Latent Dirichlet Allocation (LDA)

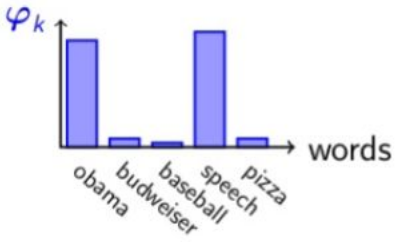
Моделює документ
як розподіл тем

Document d



Моделі теми
як розподіл слів

Topic k



http://nlpx.net/wp/wp-content/uploads/2016/01/LDA_image2.jpg

Визначення тем

Маючи корпус D, що складається з M документів, для документа d, що має N_d слів ($d \in \{1, ..., M\}$), LDA моделює D згідно з наступним генеративним процесом:

1. Вибір полиномиального розподілу φ_t для теми t ($t \in \{1, ..., T\}$) з розподілу Діріхле з параметром β.
2. Вибір поліноміального розподілу θ_d для документа d ($d \in \{1, ..., M\}$) з розподілу Діріхле з параметром α.
3. Для кожного слова w_n ($n \in \{1, ..., N_d\}$) з кожного документа d ($d \in \{1, ..., M\}$) призначте випадково одну тему t з T можливих;

Визначення тем

4. Для кожного слова w_n ($n \in \{1, ..., N_d\}$) документа ***d*** розрахувати:
   - ***p(t|d)*** - частку слів у документі, які присвоєні темі;
   - ***p(w|t)*** - частку цього слова у всіх документах, віднесених до теми;
   - призначити слову ***w*** ймовірність ***p(t|d)*** ×***p(w|t)***

$$p(d, w) = \sum_{t \in T} p(d) \cdot p(w \mid t) \cdot p(t \mid d),$$

де

- ***d*** - документт;
- ***t*** - тема;
- ***w*** - слово;
- ***T*** - набір тем;
- ***p(d)*** - апріорний розподіл набору документів;
- ***p(w|t)*** - умовний розподіл слова w в темі t;
- ***p(t|d)*** є умовним розподілом даних у документі.

Приклад

			moves strategy creation into Table instantiation so it can't be out of sync	table
t1	patch 0.075	fix 0.040	cassandra_num_ 0.039	trunk 0.020	0.1	0.13
t2	thrift 0.018	table 0.017	make 0.016	change 0.015	0.5	0.53
t3	cql 0.067	id 0.048	select 0.023	_num_e_num_ 0.021	0.16	0.27
t4	flush 0.040	write 0.030	call 0.025	memtable 0.020	0.24	0.07

Метрики якості тематичної моделі

Дві важливі методи, які використовуються для оцінки моделей теми:

- розгубленість
- когерентність теми

$$L(D') = \frac{\sum_{D} log_2 p(w_d;\Theta)}{N}$$

$$perplexity(D') = 2^{-L(D')}$$

Для визначення когерентність теми існує дві основні метрики - C_v та C_{umass}. CV базується на:

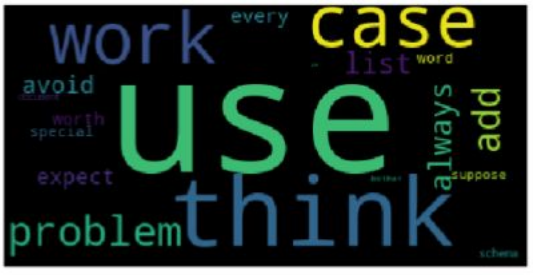
- ковзному вікні
- однокомпонентній сегментації топ-слів
- нормалізованій точковій взаємній інформації (NPMI)
- схожісті за косинусом

$$C_{UMass} = \frac{2}{N*(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} log \frac{P(w_i,w_j)+e}{P(w_j)}$$

$$NPMI(w,w) = \frac{log \frac{P(w_i,w_j)+e}{P(w_j)}}{log(P(w_i,w_j)+e)}$$

Перехід до назви та хмари слів

1. Створити порожню таблицю ***Tab*** для слів та ваг задачі
2. Для кожного коментаря C_i обраної задачі:
   a. Визначити список слів ***W*** коментаря
   b. Визначити теми ***T*** даного коментаря
   c. Для кожної теми T_i та ваги цієї теми topic_weight для коментаря C_i:
      i. Визначити топ-N слів W_i^t з вагами ***word_weigh*** кожного слова
      ii. ...
      i. ...
      ii. Для кожного слова ***word*** теми T_i:
         1. Якщо слово ***word*** присутнє в коментарі C_i:
            a. Tab[word] += word_weigh * topic_weight



Деталі реалізації

- Среда розробки - ***python*** та ***jupyter notebook***
- Робота з датасетом - база даних ***postgresql***, коннектор ***psycopg2*** та пакет для управління данними ***pandas***
- Обробка тексту (токенізація та лематизація) - ***spacy***
- Побудова тематичної моделі - ***gensim***
- Графічне представлення результатів - пакет ***word clouds***

Приклад проекту CASSANDRA

Токенизуємо та визначаємо ВЗД (VAD)

comment	clean_tokens	VAD
This is intentional. So long as you are a valid user, you can see the schema, if auth is enabled (that and some other system stuff that our tools require).\r\n\r\nThere is no practical way to limit this, so we don't.	[intentional, valid, user, see, enable, system, stuff, tool, require, practical, way, limit]	[2.6, 1.43, 2.27]
can compression and compaction parameter play a role in that problem?	[compression, play, role, problem]	[4.03, 1.0, 1.52]
Converted the map in question to CHM. Thanks for the report!	[map, question, thanks, report]	[2.96, 0.691, 1.78]
Should clarify, version is 1.2.11-SNAPSHOT as of this weekend.	[clarify, version, weekend]	[2.106, 2.96, 1.925]

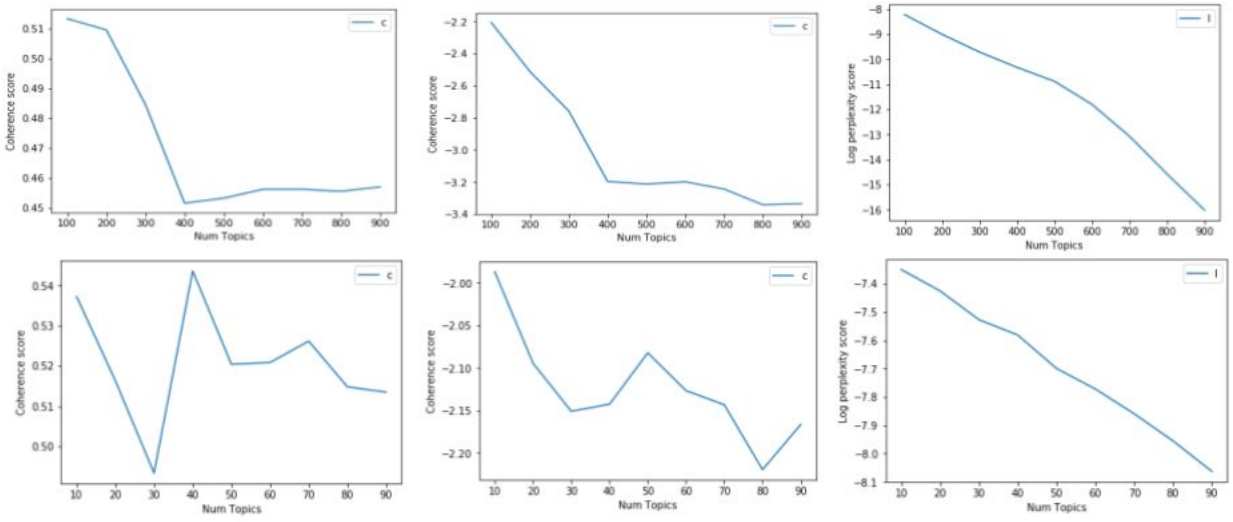
Визначаємо актуальність коментаря

comment	date	relevance
Is this different from CASSANDRA-1016?	2010-07-23 23:14:02.861	0.473519
The implementation guarantees that triggers will be executed at least once even if the update is...	2010-12-28 18:39:56.774	0.554297
Probably makes more sense to keep the trigger at the table level and pass it key + CF instance, ...	2012-11-10 13:35:52.249	0.903796
like to know by when this trigger feature will be available?	2013-02-26 17:08:07.477	0.959153
Hi Jonathan,\r\n\r\nRemoved LinkedList allocation in v3 and pushed to https://github.com/Vijay2w...	2013-05-16 13:27:50.511	0.999512
I'm going to have to object one more time to storing a jar file in the file system. With large s...	2013-05-17 12:20:01.381	1.000000

Розраховуємо інтегральний показник

comment	VAD	integral_value	relevance	integral_value_weighed
Is this different from CASSANDRA-1016?	[0.846, 0.261, 1.285]	2.392	0.473519	1.132658
The implementation guarantees that triggers will be executed at least once even if the update is...	[5.56, 3.38, 3.44]	12.380	0.554297	6.862199
Probably makes more sense to keep the trigger at the table level and pass it key + CF instance, ...	[1.47, 3.7, 1.97]	7.140	0.903796	6.453101
like to know by when this trigger feature will be available?	[2.59, 2.61, 1.405]	6.605	0.959153	6.335205
Hi Jonathan,\r\n\r\nRemoved LinkedList allocation in v3 and pushed to https://github.com/Vijay2w...	[4.04, 3.55, 3.15]	10.740	0.999512	10.734761
I'm going to have to object one more time to storing a jar file in the file system. With large s...	[4.61, 3.68, 2.94]	11.230	1.000000	11.230000

Визначення оптимальної кількості тем для моделі



Ранжуємо задачі проекту

id	integral_value	integral_value_weighed	likelihood	priority	priority_value	rate
333428	11.390000	11.390000	5	Blocker	5	25
329678	11.130000	11.130000	5	Blocker	5	25
333669	10.755556	10.737776	5	Blocker	5	25
331283	10.720000	10.720000	5	Blocker	5	25
329510	10.184286	10.184142	5	Blocker	5	25
333505	11.730000	11.730000	5	Critical	4	20
333436	11.518667	11.514989	5	Critical	4	20
330706	10.750000	10.750000	5	Critical	4	20

Ранжуємо задачі проекту

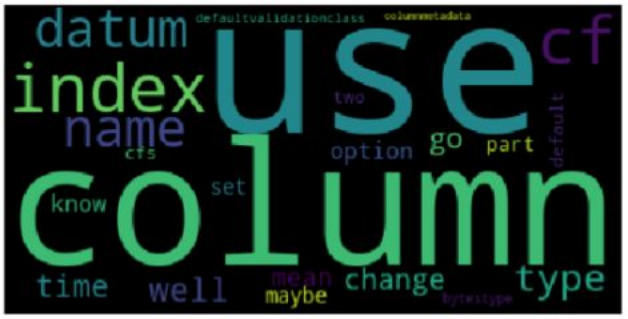
id	integral_value	integral_value_weighed	likelihood	priority	priority_value	rate
333428	11.390000	11.390000	5	Blocker	5	25
329678	11.130000	11.130000	5	Blocker	5	25
333669	10.755556	10.737776	5	Blocker	5	25
331283	10.720000	10.720000	5	Blocker	5	25
329510	10.184286	10.184142	5	Blocker	5	25
333505	11.730000	11.730000	5	Critical	4	20
333436	11.518667	11.514989	5	Critical	4	20
330706	10.750000	10.750000	5	Critical	4	20

Задача 333428

default_validation_class means "all data that isn't explicitly in **column_metadata** conforms to this data type." ***So you've violated that.*** You have two options:

- set d_v_c to BytesType (the default)

- leave the **column** definition alone, but only drop the index part (maybe this is what you were trying to do, but you changed from "colour" to "color")



More generally, note that best practice is to only use d_v_c in CFs with dynamic **column** names. I.e., if you know what the **columns** are going to be in the CF ahead of time as you do here, you shouldn't use d_v_c.

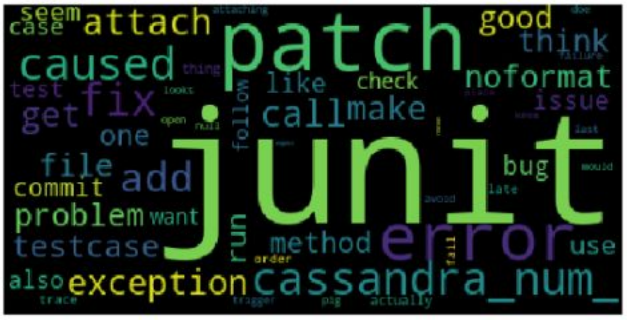
Ранжуємо задачі проекту

Id	Integral_value	Integral_value_weighed	likelihood	priority	priority_value	rate
333428	11.390000	11.390000	5	Blocker	5	25
329678	11.130000	11.130000	5	Blocker	5	25
333669	10.755556	10.737776	5	Blocker	5	25
331283	10.720000	10.720000	5	Blocker	5	25
329510	10.184286	10.184142	5	Blocker	5	25
333505	11.730000	11.730000	5	Critical	4	20
333436	11.518667	11.514989	5	Critical	4	20
330706	10.750000	10.750000	5	Critical	4	20

Коментар	ВЗД	
Those fail for me too, but that should be an easy bisect.	[5.14, 1.68, 2.05]	8.87
Looking at TokenMedataTest, it was just assuming the last test in the file was actually running last and apparently that wasn't happening on my box. Ninja-fixed that one in commit 0a5a766 to not depend on the tests execution order.	[4.13, 3.33, 3.45]	10.91
Looks pretty similar to what I did in 7de6f9666 to fix them in 1.1, except I used the shotgun method :) So if that fixes it, then +1	[4.2, 3.12, 1.8]	9.12
Alright, ScrubTest was another instance of tests expecting to run in a particular order (don't know why my box don't run them in the order they are declared but well, expecting a particular order is ***a bad idea in any case***) so ninja-fixed that. I've also committed the 2 **patches attached** above. This fixes the failure I'm saying, except for the pig tests, but as those are clearly a setup thing I don't want to block 1.2.12 for that and I've open **CASSANDRA-6376** to deal with them. Closing this issue.	[5.74, 2.94, 3.97]	12.65
Btw, also got all pig tests to fail with the following **exceptions**: {noformat} [junit] Testcase: org.apache.cassandra.pig.CqlTableDataTypeTest: **Caused** an **ERROR** **[junit] <error log>** I wouldn't block a release because of pig tests, but if it does not just fail for me, it would be nice to fix it too.	[4.62, 1.97, 2.66]	9.25
For LeaveAndBootstrapTest, this bisects to **CASSANDRA-6244**. So I think this is just a case of "we've made things asynchronous so we now check the expected result before the computation is done". Tried adding a few calls to PRCS.blockUntilFinished in the few places that were **failing** for me and that seems to fix the test. **Attaching** the resulting **patch**. [~brandon.williams] can you check it's not entirely ***stupid***?	[4.35, 2.66, 2.76]	9.77
Regarding the **ConcurrentModificationException**, it seems that the only reason this could get triggered is due to TMD.clearUnsafe(). As this is called by tests, this is not a real ***problem***, but what about making it grab the writeLock like any good citizen to avoid getting scarry stack traces (and don't discard a real bug later on because we've grown used to discarding such stack)? Attaching **patch** to do that.	[4.57, 3.46, 2.69]	10.72

Задача 329510

- Багато коментарів
- Багато специфічних термінів та назв
- Згадка інших задач та комітів
- Було зроблено багато патчів
- Головна проблема пов’язана з пакетом junit



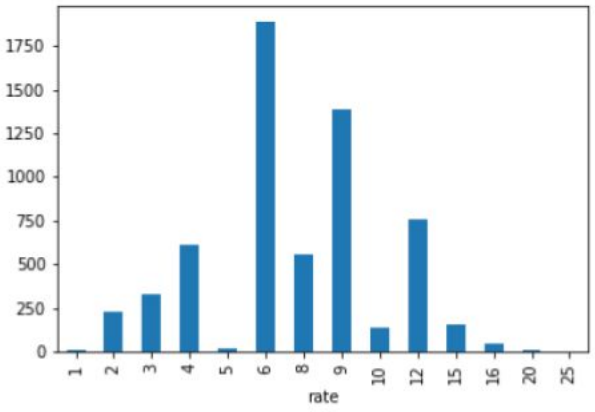
Коментарі задач з низькою важливістю

задача	коментар	ВЗД	
331618	bah, just realised you can use comparator= 'CompositeType(UTF8Type, UTF8Type)'	[0.116, 0.021, 0.185]	0.322
332691	duplicate of CASSANDRA-3164	[0.364, 0.289, 0.315]	0.968
330393	Resolving now that it's in trunk.	[0.044, 0.701, 0.275]	1.02
333721	done as part of CASSANDRA-2521	[0.294, 0.851, 0.025]	1.17
329561	{{ECHO OFF}}	[0.136, 0.601, 0.595]	1.33
	(this should be ninja-d)	[0.076, 1.399, 0.045]	1.52

Як бачимо з цих коментарів, задачі не потребують додаткової уваги та мають більш інформаційний характер

Розподіл задач за показником важливості

- Задачі, які потребують додаткової уваги (16-25), в меншості
- Найбільш велика група - задачі з важливістю 6
- З даного розподілу бачимо важливість ранжування для оптимізації ресурсів



коментар	ВЗД	
I did a bit more tests and here are some results which might help: 1. **JMX** port set to 9090 in cassandra-env.sh 2. On the machine where another service **running** on 8080 we get exception above 3. On the machine where no service **running** on 8080 we don't get any exception and MX4J **runs** on port 9090 Seems like something checks for port 8080 even though it is **configured** to **run** on 9090.	[3.92, 2.29, 2.83]	9.04
I think there's a confusion. There are two ports in business, one is the **JMX** port (default is 8080) and one is the MX4J port (default 8081) If the **JMX** port is used when cassandra starts you see the following exception, which is different from what's pasted in this bug report: **<error log>** So the problem in this case. I believe was that mx4j's port was bound to a different process. To control the port used by mx4j use -Dmx4jport=8082. See https://issues.apache.org/jira/browse/**CASSANDRA-1068** for more details. I think this is not a bug and recommend to close it as such. I will, however, attach a **patch** for trunk to make this more obvious and **add** -Dmx4jport=8081 to conf/cassandra-env.sh	[5.05, 4.17, 4.06]	**13.28**
Patch that **adds** the variables MX4J_ADDRESS and MX4J_PORT to conf/cassandra-env.sh make **configuration** for mx4j obvious.	[1.65, 1.371, 2.39]	5.411
You right Ran, I checked this machine again and I have another service listening on 8081. For some reason I thought that MX4J uses same port. With **config** options we can close it now.	[2.32, 2.05, 2.49]	6.86
committed, thanks!	[2.706, 0.881, 1.705]	5.292

Задача 334028

- Показник ВЗД
  - *збільшувався* відповідно до *інтенсивності обговорення проблеми* та
  - *зменшувався* відповідно до її *вирішення*
- Хмара слів має “дякую” як одно з ключових слів

Підведення підсумків та шляхи розвитку

Висновки

- використання лише цих трьох емоційних складових достатньо для ранжування коментарів та задач в залежності від наявності потенційних проблем та напруження у коментарях
- оскільки назва ризику задачі формується зі слів, що наявні в коментарях, маємо релевантні результати

Новизна:

- З'єднання двох різних підходів:
  - визначення емоційних складових у тексті
  - матричних методів аналізу ризиків проекту

Шляхи подальшого розвитку

Емоції - ризики

- розглянути та застосувати інші емоційні фреймворки та показники
  - наприклад, визначати рівень ввічливості або агресивності
- сентиментальний аналіз
- врахування особливостей сучасної комунікації, а саме
  - сленг, смайли, +1
- дослідити вплив довжини коментарю на значення результуючого показника

Назва ризику

- автоматизація обрання оптимальних параметрів
- генерування більш природних назв з ключових слів та вагів
- одна задача може містити у собі декілька ризиків, здатність виокремити їх з розмови розробників

Дякую за увагу!

ДОДАТОК Б ЗРАЗКИ ДАНИХ

Останні 10 коментарів з таблиці jira_issue_comment:

{41956: {'id': 335857,
'comment': 'Yay! Nice work. \r\n\r\nSomething seems to be up with the wiki configuration, here\'s what I get when I click on the "wiki" link:\r\n\r\nMoinMoin Configuration Error\r\nCould not find a match for url: "wiki.apache.org/cassandra/".\r\nCheck your URL regular expressions in the "wikis" list in "farmconfig.py".\r\nsunos5 (posix) Python 2.4.4 (/usr/bin/python) MoinMoin release 1.3.4 (revision 1.3.4 release)\r\n',
'date': Timestamp('2009-03-27 04:30:00.226000')},
41957: {'id': 335857,
'comment': 'what do we need to do to get a publicly editable wiki?',
'date': Timestamp('2009-03-27 04:35:41.615000')},
41958: {'id': 335857,
'comment': 'I believe we can just ask a mentor to set it up for us. I created CASSANDRA-15 to keep track of progress.',
'date': Timestamp('2009-03-27 05:16:53.664000')},
41959: {'id': 335857,
'comment': 'Closing since the site is up. I dont have access to set the assignee, should be Avinash.',
'date': Timestamp('2009-03-27 05:17:40.944000')},
41960: {'id': 335858,
'comment': "This patch makes changes that make remove support easier or possible:\r\n\r\nColumn:\r\n - add boolean isMarkedForDelete. If true, the timestamp field represents the deletion time\r\n - all fields are final (immutable). This avoids the need for Atomic* variables and makes whole classes of bugs impossible\r\n\r\nSuperColumn:\r\n - removed boolean isMarkedForDelete\r\n - long markedForDeleteAt added. If greater than MIN_VALUE, it is considered deleted at the given time\r\n - putColumn() and repair() combined; renamed to integrate()\r\n\r\nColumnFamily:\r\n - long markedForDeleteAt added, as in SuperColumn\r\n - isSuper() convenience method added\r\n - addColumn and createColumn methods combined; all are now overloads of addColumn. Note that addColumn(name, column) was removed in favor of simply addColumn(column) since the column already knows its name, and allowing a different one

to be specified could result in hard-to-find bugs\r\n - serializer always dumps + loads the Columns; trying to optimize by leaving them out causes bugs with remove\r\n - renamed serializer2 to serializerWithIndexes\r\n - renamed getColumnFamilies to getColumnFamilyMap. Added getColumnFamilies method returning only the CF collection (the map values).\r\n\r\nMemtable:\r\n - added SuperColumn support to forceFlush. Refactored flush methods slightly so that the only one who cares about fRecovery is Table. [everyone else just passed False.]\r\n\r\nNamesFilter:\r\n - makes a copy of the List it is passed. This fixes a bug that may not be specific to remove support.\r\n\r\nRow:\r\n - merge() removed (duplicate of Repair)\r\n\r\nRowMutation:\r\n - added makeRowMutationMessage()\r\n - added sanity checks to add()\r\n - added delete(columnFamilyColumn, timestamp) method\r\n - cleaned up duplicate code in apply() overloads\r\n\r\nMessage:\r\n - Changed constructor from Object[] body to Object... body. This allows (but does not require) single Objects to be passed without explicitly wrapping in a new Object[] {}.\r\n\r\nGeneral:\r\n - old-style remove/delete support removed, since it's going to be rewritten in the next patch\r\n\r\nThe other changes are just dealing with the consequences of the above, particularly the getColumnFamilyMap rename and the CF.addColumn parameter change.",
 'date': Timestamp('2009-03-07 11:38:38.982000')},
41961: {'id': 335858,
 'comment': 'This patch provides the actual remove support internally. Thrift API support is not yet included.\r\n\r\nColumnComparatorFactory:\r\n - fix exception when comparing two SuperColumns with the same name\r\n\r\nColumnFamilyStore:\r\n - Split resolve() into resolve(), which combines ColumnFamilies, and removeDeleted(), which takes a single ColumnFamily and returns a new one with deleted IColumns removed. Keep deletion information around until removeDeleted is called so that deletion information can properly supress older IColumns.\r\n\r\nRowMutationVerbHandler:\r\n - send response back so blocking calls can work\r\n\r\nWriteResponseMessage: \r\n - Renamed to WriteResponse to avoid confusion with Message class\r\n\r\nStorageProxy:\r\n - added insertBlocking method for use by batch_insert_blocking, batch_insert_superColumn_blocking, and remove in blocking mode.\r\n\r\nCassandraServer:\r\n - added remove(String, String, String, long, int). Thrift needs to be modified to expose this and not the old remove (which is left in as a stub to keep the build happy).\r\n',
 'date': Timestamp('2009-03-07 11:39:49.299000')},

41962: {'id': 335858,
'comment': 'This fixes the CF deserialization in SequenceFile to know about the format change (boolean -> long).',
'date': Timestamp('2009-03-11 02:44:58.262000')},
41963: {'id': 335858,
'comment': "I've updated my patches to apply against current trunk and split into bite-sized pieces. Each piece corresponds to one of the steps in the larger patches described above. (Full description is in a Subject: line in the header for each patch.)",
'date': Timestamp('2009-03-24 23:08:49.294000')},
41964: {'id': 335858,
'comment': 'No. You cannot free up memory. It will be get garbage collected once they are no longer actively referenced which will be the case. Setting it to NULL (which is what the clear() does) is not going to force any GC anyways. Hence it is moot.',
'date': Timestamp('2009-03-25 12:10:57.224000')},
41965: {'id': 335858,
'comment': 'Committed in r758965 - r758983',
'date': Timestamp('2009-03-27 10:22:45.401000')}}

Останні 10 коментарів з таблиці jira_issue_report:

{335849: {'priority': 'Major'},
335850: {'priority': 'Major'},
335851: {'priority': 'Major'},
335852: {'priority': 'Critical'},
335853: {'priority': 'Minor'},
335854: {'priority': 'Major'},
335855: {'priority': 'Blocker'},
335856: {'priority': 'Major'},
335857: {'priority': 'Blocker'},
335858: {'priority': 'Major'}}

ДОДАТОК В ЛІСТИНГ ПРОГРАМИ

```
import psycopg2

conn = psycopg2.connect(database="jira_dataset",
                        user = "jiu",
                        password = "123456",
                        host = "127.0.0.1")#, port = "5432")
cur = conn.cursor()
PROJECT = 'CASSANDRA'

cur.execute("""
        SELECT issue_report_id, comment,
        updatedate,
        arousal_mean_sum, dominance_mean_sum, valence_mean_sum
        from jira_issue_comment
        where issue_report_id IN (select id from jira_issue_report where project='CASSANDRA')
        """)

comments = cur.fetchall()

cur.execute("""
        SELECT id, priority
        from jira_issue_report
        where project='CASSANDRA'
        """)

priorities = cur.fetchall()
priorities = pd.DataFrame(priorities, columns=[0, 'priority']).set_index(0)
priority_map = {
        'Trivial': 1,
        'Minor': 2,
        'Major': 3,
        'Critical':4,
        'Blocker':5
}

priorities['priority_value'] = priorities.priority.map(priority_map)

import pandas as pd
import numpy as np

rates = pd.read_csv('Ratings_Warriner_et_al.csv', index_col=0).set_index('Word')
mV, mA, mD = rates[['V.Mean.Sum', 'A.Mean.Sum', 'D.Mean.Sum']].mean().values

def get_VAD_range(sentence):
```

```
        minV, minA, minD = np.min(sentence, 0)
        maxV, maxA, maxD = np.max(sentence, 0)
        return [
        get_range_value(minV, maxV, mV),
        get_range_value(minA, maxA, mA),
        get_range_value(minD, maxD, mD),
        ]


def get_range_value(minW, maxW, meanW):
        result=0
        if minW > meanW:
        result = maxW - meanW
        elif maxW < meanW:
        result = meanW - minW
        else:
        result = maxW - minW
        return round(result,3)


def get_representation(tokens):
        representation = []
        for token in tokens:
        try:
        representation += [rates.loc[token][['V.Mean.Sum', 'A.Mean.Sum', 'D.Mean.Sum']].values]
        except: pass
        return representation


data = pd.DataFrame(comments, columns=['id', 'comment', 'date', 'A', 'D', 'V'])

import nltk, re
import spacy
nlp = spacy.blank('en')
nlp.add_pipe(nlp.create_pipe('sentencizer'))

def get_good_tokens(sentence):
        stopwords = nltk.corpus.stopwords.words('english')+['pron', '_num_']

        replaced_punctation = list(map(lambda token: re.sub('[^0-9A-Za-z]+', '', token), sentence))
        replaced_numbers = list(map(lambda token: re.sub('[0-9]+', '_num_', token),
replaced_punctation))
        removed_punctation = list(filter(lambda token: token, replaced_numbers))
        removed_stopwords = list(filter(lambda token: token not in stopwords,
removed_punctation))
        return removed_stopwords
```

```
data['clean_text'] = data.comment.apply(lambda post:
                                        get_good_tokens(list(map(lambda x: str(x.lemma_).lower(),
nlp(post)))))
data['clean_tokens'] = data['clean_text'].apply(lambda x: [word for word in x if word in rates.index])
data['len_tok'] = data.clean_tokens.apply(len)
new_data = data[data.len_tok>0]
new_data['date_int'] = (pd.to_datetime(new_data.date).astype(int) - 1200000000000000000)

new_data  = pd.merge(new_data,
         new_data[['id', 'date_int']].groupby('id').max().rename(columns={'date_int':
'date_max'}).reset_index()
         , how='left')

new_data['relevance'] = new_data['date_int'] / new_data['date_max']
new_data['VAD'] = new_data['clean_tokens'].apply(lambda x:
get_VAD_range(get_representation(x)))
new_data['integral_value'] = new_data.VAD.apply(lambda x: sum(x))
new_data['integral_value_weighed'] = new_data['integral_value'] * new_data['relevance']


def get_likelihood(rate):
         if rate>=10: return 5
         elif rate>=8: return 4
         elif rate>=5: return 3
         elif rate>=2: return 2
         else: return 1

tasks_risks = new_data.groupby('id').mean()[['integral_value', 'integral_value_weighed']]
tasks_risks['likelihood'] = tasks_risks.integral_value.apply(get_likelihood)
tasks_risks = tasks_risks.join(priorities).sort_values('likelihood')

tasks_risks['rate'] = tasks_risks['likelihood'] * tasks_risks['priority_value']

tasks_risks.sort_values(['rate', 'integral_value_weighed'], ascending=False)[tasks_risks.rate==6]

tasks_risks.groupby('rate').count()['integral_value'].plot(kind='bar')

# ## LDA

from gensim.corpora import Dictionary

dictionary = Dictionary(documents=data.clean_text.values)
print("Found {} words.".format(len(dictionary.values())))
```

```
dictionary.filter_extremes(no_below=7, no_above=0.7)
print("Found {} words.".format(len(dictionary.values())))

data['bow'] = list(map(lambda doc: dictionary.doc2bow(doc), data.clean_text))

from gensim.models import LdaModel, LdaMulticore

corpus = data.bow
num_topics = 40
#A multicore approach to decrease training time
LDAmodel = LdaMulticore(corpus=corpus,
                    id2word=dictionary,
                    num_topics=num_topics,
                    workers=4,
                    chunksize=4000,
                    passes=7,
                    alpha='asymmetric')

from gensim.models import CoherenceModel
import gensim

def compute_coherence_values(dictionary, corpus, texts, limit, start=2, step=3):
        perplexity_values = []
        cv_coherence_values = []
        umass_coherence_values = []
        bounds_values = []
        model_list = []
        f = open('logs.txt', 'w')
        for num_topics in range(start, limit, step):
        model = LDAmodel = LdaMulticore(corpus=corpus,
                    id2word=dictionary,
                    num_topics=num_topics,
                    workers=8,
                    alpha='asymmetric')
        model_list.append(model)
        perplexity_values.append(LDAmodel.log_perplexity(corpus))
        coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary,
coherence='c_v')
        cv_coherence_values.append(coherencemodel.get_coherence())
        coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary,
coherence='u_mass')
        umass_coherence_values.append(coherencemodel.get_coherence())
        bounds_values += [LDAmodel.bound(corpus)]
        print(num_topics, perplexity_values[-1], cv_coherence_values[-1],
umass_coherence_values[-1],
        bounds_values[-1])
```

```
        f.write("{0}\t{1}\t{2}\t{3}\t{4}".format(num_topics, perplexity_values[-1],
cv_coherence_values[-1],
                                                umass_coherence_values[-1], bounds_values[-1]))
        f.close()
        return model_list, cv_coherence_values, perplexity_values, umass_coherence_values,
bounds_values


model_list, cv_coherence_values, perplexity_values, umass_coherence_values, bounds_values =
compute_coherence_values(dictionary=dictionary, corpus=corpus,
                                                texts=data.clean_text, start=10, limit=100, step=10)

import matplotlib.pyplot as plt
limit=1000; start=100; step=100;
x = range(start, limit, step)


plt.plot(x, cv_coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()


plt.plot(x, umass_coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

plt.plot(x, log_perplexity_values)
plt.xlabel("Num Topics")
plt.ylabel("Log perplexity score")
plt.legend(("log preprexity"), loc='best')
plt.show()



total_total = dict()
for i in data[data.id==334028].index:
        ids = list(dict(data.loc[i].bow).keys())
        total = dict((dictionary.id2token[x],0) for x in ids)
        for topic, weight in LDAmodel.get_document_topics(data.loc[i].bow):
        topic_terms = dict(LDAmodel.get_topic_terms(topic,100))
        for id_ in ids:
        try:
```

```
                total[dictionary.id2token[id_]] += topic_terms[id_] * weight
        except KeyError:
                pass
        total_total.update(total)


sorted(filter(lambda x: x[1], total_total.items()), key=lambda x: -x[1])[:5]


import matplotlib.pyplot as plt
from wordcloud import WordCloud


plt.figure(figsize=(9, 9))
plt.imshow(WordCloud().fit_words(dict(filter(lambda x: x[1], total_total.items()))))
plt.axis("off")
plt.show()
```

ДОДАТОК Г АНКЕТА МАРКЕТИНГОВОГО ДОСЛІДЖЕННЯ

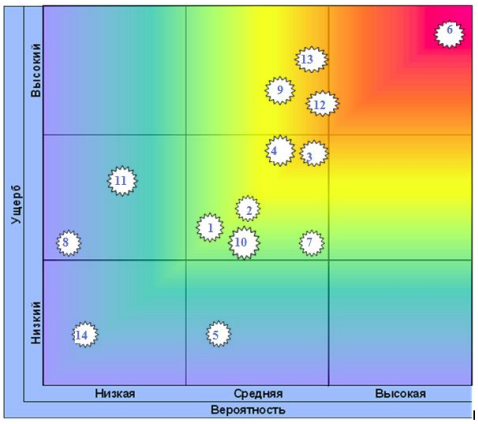
Уважаемый респондент!

Перед Вами анкета, которая содержит ряд вопросов посвященных теме управления проектными рисками и коммуникациям. Все данные конфиденциальны и будут использованы в обобщенном виде. Опрос займет не более 10 минут.

Благодарим за сотрудничество!

1. Вы работаете в ИТ компании?
    a. да
    b. нет
2. Ваша компания является …
    a. продуктовой
    b. outsource
    c. outstaff
    d. другое
3. Какой стиль управления используете?
    a. Scrum
    b. Kanban
    c. Другой
4. Какой инструмент используется для управления проектами?
    a. Jira
    b. Microsoft Project
    c. Redmine
    d. Git Projects
    e. Другой
5. Оцениваете ли Вы сроки выполнения задач?
    a. да
    b. нет
    c. не всегда
6. Анализируете ли Вы точность прогнозов по окончанию проекта?
    a. да
    b. нет
    c. не всегда

7. С чем ассоциируется у Вас данное изображение?



   a. ...
8. Осуществляется ли идентификация рисков проекта?
   a. да, всегда
   b. в большинстве проектов
   c. иногда
   d. редко, лишь в некоторых проектах
   e. нет
9. Считаете ли Вы этот процесс необходимым для успешного ведения проекта?
   a. да
   b. нет
   c. не уверен
10. Что самое неприятное в управлении рисками? (мультивыбор)
   a. Затрачиваемое время
   b. Затрачиваемые усилия
   c. Отсутствие готового списка для выбора рисков
   d. Оформление / документация
   e. Сложно охватить все риски / Что-то каждый раз упускается
   f. Другое
11. Есть ли в Вашей компании готовый список рисков?
   a. да
   b. нет
12. Оцените частоту следующих явлений

	Всегда	Часто	Иногда	Редко	Никогда
Часто ли Вы оставляете комментарии к задаче?	○	○	○	○	○
Часто ли Ваши коллеги оставляют комментарии к задаче?	○	○	○	○	○
Вежливы ли Ваши коллеги в комментариях?	○	○	○	○	○
Обсуждаются ли проблемы в комментариях?	○	○	○	○	○

a.

13. Как бы Вы оценили комментарии в проекте?

	1	2	3	4	5	
Нейтральные	○	○	○	○	○	Очень эмоциональные

a.

14. Как меняется общение ближе к дедлайнам?
    a. Учащается
    b. Не меняется
    c. Становится редким
    d. Не замечал(-а)
15. В процессе управления проектами в нашей компании мне бы хотелось изменить ...
16. Ваш пол?
    a. М
    b. Ж
17. Ваш возраст?
    a. 18-23
    b. 24-27
    c. 28-35
    d. 35+
18. Ваша должность больше всего близка к ...
    a. Developer
    b. Analyst
    c. Team Lead
    d. Project Manager

e. Head of ..
f. Другое

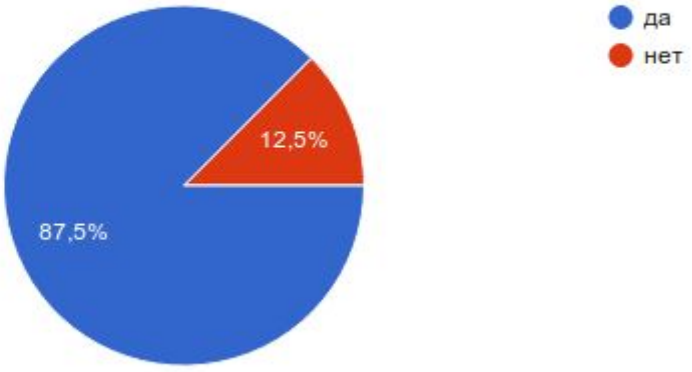
19. В Вашей компании работает …
    a. менее 11 человек
    b. 11-20 человек
    c. 21-80 человек
    d. 80-200 человек
    e. 200-800 человек
    f. 800 + человек

Спасибо за участие!

ДОДАТОК Ґ РЕЗУЛЬТАТИ МАРКЕТИНГОВОГО ДОСЛІДЖЕННЯ

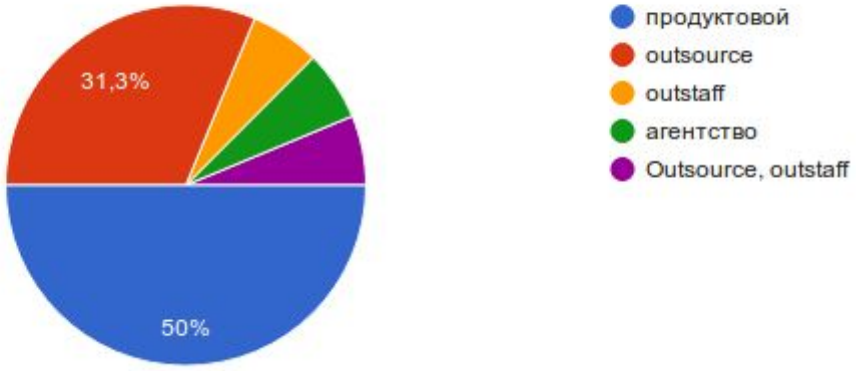
Вы работаете в ИТ компании?

16 ответов



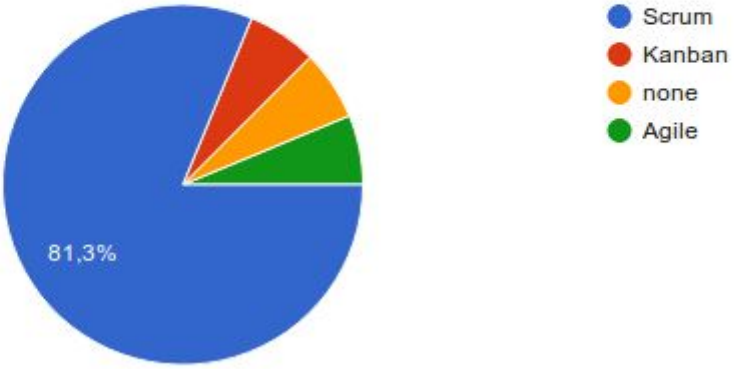
Ваша компания является ...

16 ответов



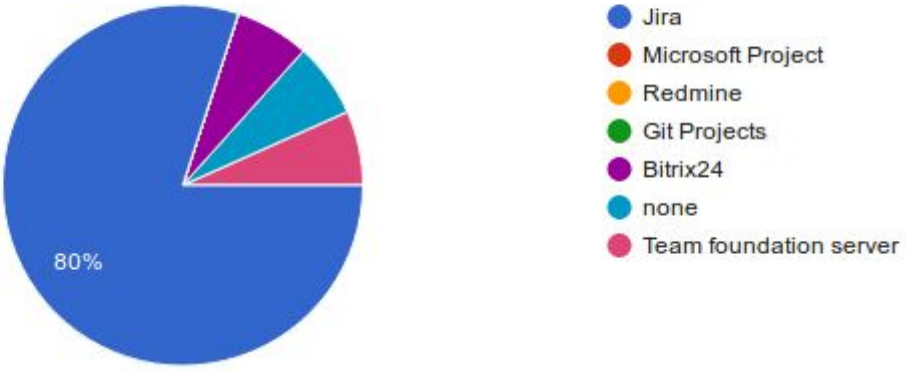
Какой подход управления проектами используется в Вашей компании? (в большинстве случаев)

16 ответов



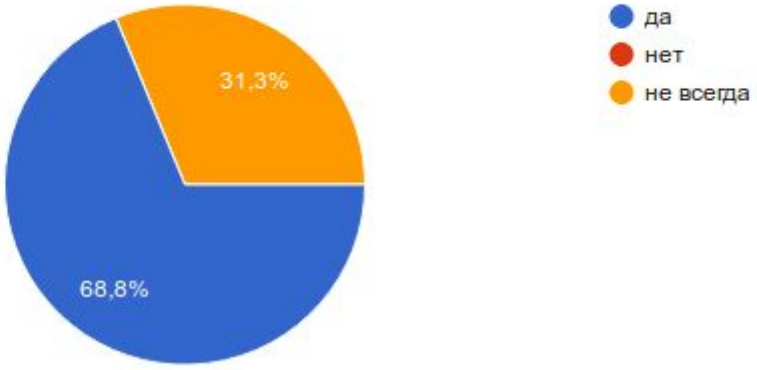
Какой инструмент используется для управления проектами?

15 ответов



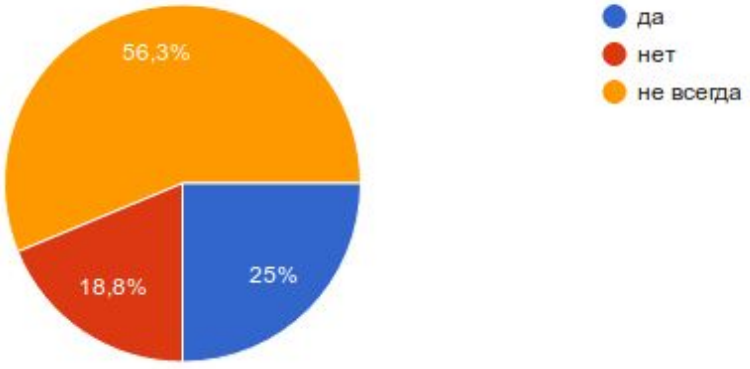
Оцениваете ли Вы сроки выполнения задач?

16 ответов



Анализируете ли Вы точность прогнозов этих оценок по окончанию проекта?

16 ответов



С чем ассоциируется у Вас данное изображение?

12 ответов

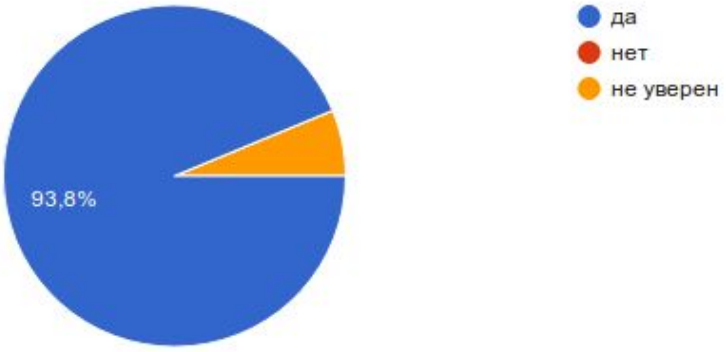
Нет
Риски
приоритетность задач
Метод наименьших квадратов
Судя по тематике вопросов, это матрица рисков
Ёжики
Скоринг
Оценка рисков
оценка успешности компании по количеству рискованных проектов
Risk assessment
критерии рисков
-

Осуществляется ли идентификация рисков в начале проекта в вашей компании?

15 ответов

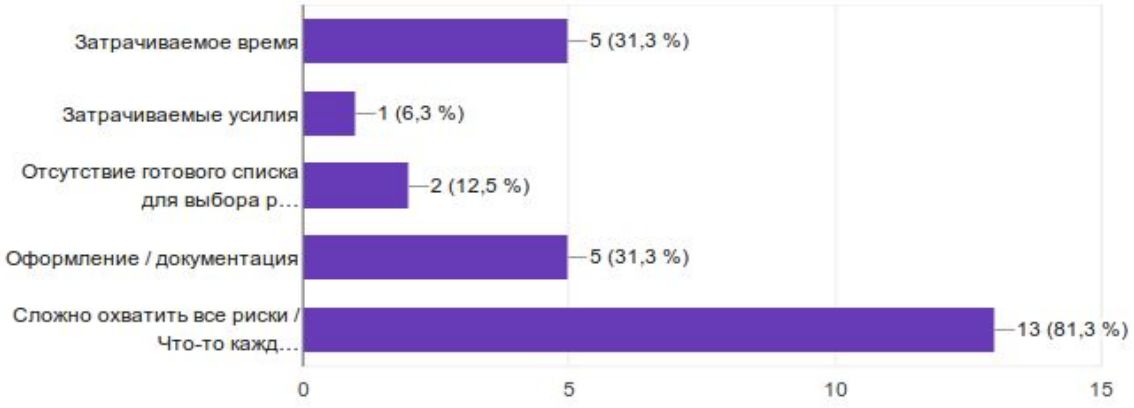
Как Вы думаете, является ли этот процесс необходимым для успешного ведения проекта?

16 ответов



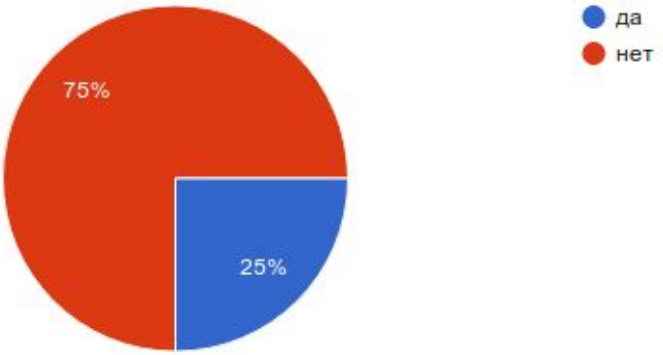
Что самое неприятное в управлении рисками?

16 ответов



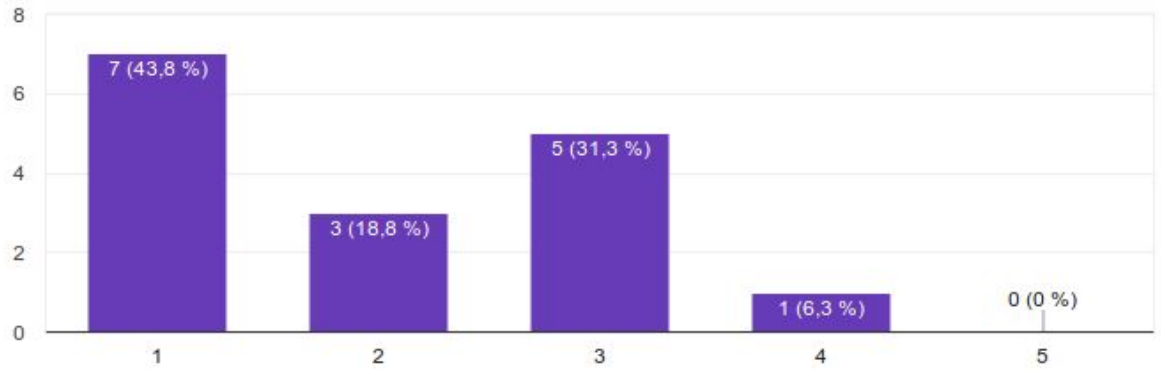
Есть ли в Вашей компании готовый список рисков?

16 ответов



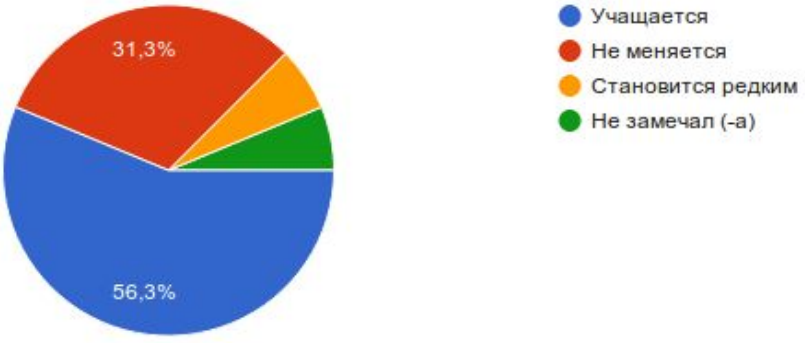
Как бы Вы оценили комментарии в проекте?

16 ответов



Как меняется общение ближе к дедлайнам?

16 ответов



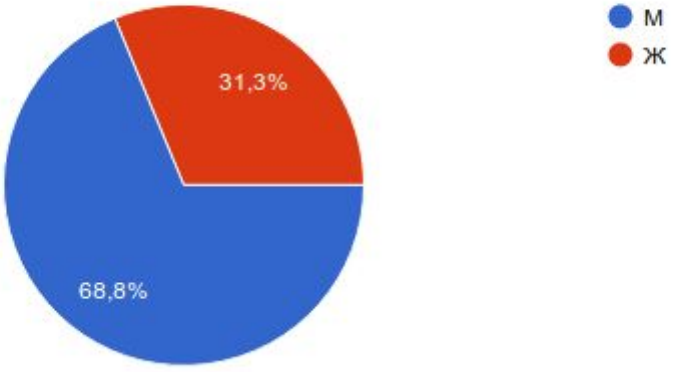
В процессе управления проектами в нашей компании мне бы хотелось изменить ...

8 ответов

подход к проверке задач на разных ее стадиях
Ничего
Собрать разработчиков в одном месте
ведення логів
Ничего
качество связи "заказчик-исполнитель"
и так сойдет
-

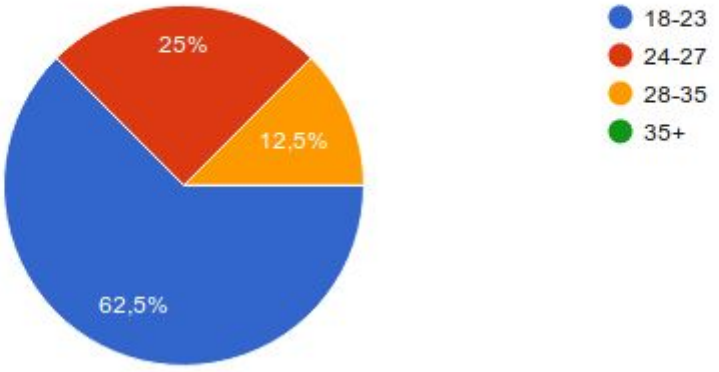
Ваш пол?

16 ответов



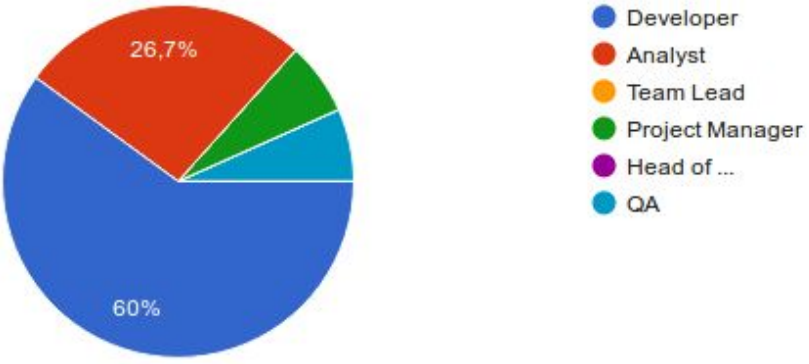
Ваш возраст?

16 ответов



Ваша должность больше всего близка к ...

15 ответов



В Вашей компании работает ...

16 ответов