

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

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

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: **Методика організації роботи розподіленої Agile**
команди у великих ІТ-проектах

Виконав(ла): студент(ка) 6 курсу, групи СНМ-61
спеціальності _____

112 Комп'ютерні науки

(шифр і назва спеціальності)

Вовк А.І.

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Гром'як Р.С.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Загородна Н.В.

(прізвище та ініціали)

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

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

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«21» вересня 2020 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

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

студенту Вовк Андрій Іванович
(прізвище, ім'я, по батькові)

1. Тема роботи Методика організації роботи розподіленої Agile команди у великих ІТ-проектах

Керівник роботи к.т.н., доц. Гром'як Р.С.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «06» листопада 2020 року № 4/7-825

2. Термін подання студентом завершеної роботи 23 грудня 2020 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

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

ВСТУП ;РОЗДІЛ 1 ОГЛЯД SCRUM-МЕТОДОЛОГІЇ ТА ПРОБЛЕМИ ПРИ РОБОТІ НАД ВЕЛИКИМИ ПРОЕКТАМИ ;1.1 Основні поняття гнучкої методології Scrum ;1.2 Основні задачі, які вирішуються керівництвом проекту для координації роботи багатьох команд; РОЗДІЛ 2 ОПИС ВІДБРАНИХ ДЛЯ АНАЛІЗУ ДАНИХ ;2.1 Опис проекту А ;2.2 Опис проекту В ;2.3 Збір даних ;РОЗДІЛ 3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ;3.1 Моделі зовнішньої координації команд ;3.2 Профілі команд та їх координаційні потреби ;3.3 Потреба команд у координації роботи ;РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ ; ВИСНОВКИ ;ПЕРЕЛІК ПОСИЛАНЬ ;ДОДАТКИ

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

1. Тема, мета роботи. 2. Наукова новизна отриманих результатів. 3. Постановка задачі.

4. Фреймворк Scrum. 5. Досліджувана предметна область. 6. Характеристики потреби координації команд. 7. Числові дані про координацію команд. 8. Візуалізація таблиці координації команд. 9. Групування команд відносно потреб їх координації. 10. Вплив координації на ефективність роботи команд. 11. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Дмитроца Л.П., доц.		
Безпека в надзвичайних ситуаціях	Стадник І.Я., проф.		

7. Дата видачі завдання 21 вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	21.09.20-27.09.20	<i>Виконано</i>
2.	Підбір наукових джерел по темі роботи	28.09.20-04.10.20	<i>Виконано</i>
3.	Переклад та опрацювання наукових джерел по темі кваліфікаційної роботи	05.10.20-11.10.20	<i>Виконано</i>
4.	Виконання дослідження щодо огляду атак на комп'ютерні системи	12.10.20-18.10.20	<i>Виконано</i>
5.	Оформлення першого розділу	19.10.20-25.10.20	<i>Виконано</i>
6.	Оформлення другого розділу	26.10.20-01.11.20	<i>Виконано</i>
7.	Оформлення третього розділу	02.11.20-08.11.20	<i>Виконано</i>
8.	Виконання завдання до підрозділу «Охорона праці»	09.11.20-15.11.20	<i>Виконано</i>
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	16.11.20-22.11.20	<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи	23.11.20-29.11.20	<i>Виконано</i>
11.	Нормоконтроль	30.11.20-05.12.20	<i>Виконано</i>
12.	Перевірка на плагіат	05.12.20	<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи	14.12.20	<i>Виконано</i>
14.	Захист кваліфікаційної роботи	23.12.2020	

Студент

_____ (підпис)

Вовк А.І.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Гром'як Р.С.

_____ (прізвище та ініціали)

АНОТАЦІЯ

"Методика організації роботи розподіленої Agile команди у великих ІТ-проектах" // Вовк Андрій Іванович // Тернопільський національний технічний університет ім. І. Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2020 // с. – , рис. – , табл. – , джерел – .

Ключові слова: ПРОГРАМНА ІНЖЕНЕРІЯ, ГНУЧКІ ТЕХНОЛОГІЇ, АЖІЛЕ, КООРДИНАЦІЯ, SCRUM.

Команди, що працюють у масштабних проектах, часто не мають усіх знань для виконання своїх завдань, які також є дуже взаємозалежними. Тому їм потрібно покладатися на координацію з іншими командами, експертами та допоміжними ролями. У цій роботі ми досліджуємо координаційні потреби команд та оцінюємо вплив задоволення цих потреб на ефективність роботи команди.

ANNOTATION

"The job management strategy of distributed Agile team in big IT-projects" // Diploma paper of Master degree level // Vovk Andrii Ivanovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department // Ternopil, 2020 // p. – , Fig. – , Tables – , Refence – .

Key words: SOFTWARE ENGINEERING, FLEXIBLE TECHNOLOGIES, AGILE, COORDINATION, SCRUM

Teams working on large-scale projects often do not have all the knowledge to perform their tasks, which are also highly interdependent. Therefore, they need to rely on coordination with other teams, experts and support roles. In this paper, we explore the coordination needs of teams and assess the impact of meeting these needs on team performance.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД SCRUM-МЕТОДОЛОГІЇ ТА ПРОБЛЕМИ ПРИ РОБОТІ НАД ВЕЛИКИМИ ПРОЄКТАМИ	10
1.1 Основні поняття гнучкої методології Scrum.....	10
1.2 Основні задачі, які вирішуються керівництвом проекту для координації роботи багатьох команд	16
1.3 Роль команди у великих гнучких Agile проектах.....	17
1.3.1 Загальне керівництво	19
1.3.2 Зовнішня координація.....	20
РОЗДІЛ 2 ОПИС ВІДБРАНИХ ДЛЯ АНАЛІЗУ ДАНИХ	22
2.1 Опис проекту А	22
2.2 Опис проекту В	23
2.3 Збір даних	24
2.3.2 Аналіз потреб в координації команд	25
2.3.3 Потреба в зовнішній координації команд.....	27
2.3.4 Робота команди.....	29
РОЗДІЛ 3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ	30
3.1 Моделі зовнішньої координації команд	30
3.2 Профілі команд та їх координаційні потреби	33
3.2.2 Група 1 (команда С і команда Н).....	36
3.2.3 Група 2 (команда D та команда E).....	37
3.2.4 Група 3 (команда F).....	37
3.2.5 Група 4 (команда А, команда В, команда G та команда I).....	38
3.3 Потреба команд у координації роботи	39
3.3.1 Група 1 (команда G та команда Н)	40
3.3.2 Група 2 (команда D, команда E та команда F)	40
3.3.3 Група 3 (команда В, команда С і команда I).....	41

3.3.4 Група 4 (команда А)	42
3.4 Вплив рівня зовнішньої координації на результатами роботи команд.....	43
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ...	46
4.1 Впровадження в Україні світового досвіду щодо покращення умов і безпеки праці в ІТ-компаніях	46
4.2 Вплив електромагнітного імпульсу (ЕМІ) ядерного вибуху на елементи виробництва та заходи захисту.....	48
ВИСНОВКИ	53
ПЕРЕЛІК ПОСИЛАНЬ	55
ДОДАТКИ	

ВСТУП

Актуальність теми. Масштабна розробка програмного забезпечення все частіше використовує гнучкі практики. У широкомасштабних проектах команда повинна співпрацювати з іншими командами та організаціями. Це загрожує автономії команди, що, в свою чергу, знижує швидкість реагування та гнучкість. Отже, Agile команди стикаються з проблемами в адаптації до масштабнішого розвитку.

Командам у масштабних проектах необхідно досягти згоди щодо багатьох рішень з експертами, менеджерами, зацікавленими сторонами та іншими командами [1]. Крім того, проблеми з якістю та необхідність частих та скоординованих релізів змушують компанії керувати, контролювати та стандартизувати зусилля з розробки декількох команд. Таким чином, гнучка команда, яка працює в великомасштабних середовищах, повинна бути узгоджена з іншими командами і організаціями. Якщо команда порушить якість або функціональність або запізниться, це вплине на інші команди та результати. Однак необхідність узгодження роботи, процесів та технологій та зовнішньої координації є загрозою для автономії команди, що є запорукою Agile.

У зв'язку з актуальністю теми та важливістю нею займаються ряд науковців, в тому числі і в ТНТУ. Зокрема серед робіт цієї тематики варто згадати такі: [2], [3], [4], [5], [6].

Мета роботи. Основною метою даної роботи є розуміння факторів, що сприяють чи перешкоджають автономії, та дослідження конфлікту між автономією на рівні команди та необхідністю організаційного контролю у масштабній гнучкій розробці програмного забезпечення, а також як зовнішньо-командна координація впливає на ефективність роботи команди. Тому наша одиниця аналізу – команда розробників програмного забезпечення у масштабному проекті.

Тому ставимо такі задачі дослідження:

– Які перепони та позитивні фактори для автономії команди у великому Agile проекті?

– Що характеризує координаційні потреби команд у масштабній розробці програмного забезпечення?

– Який взаємозв'язок між зовнішньою координацією та ефективністю команд у масштабній розробці програмного забезпечення?

Об'єкт дослідження: процеси координації роботи декількох команд розробників ПЗ, що працюють у відповідності до гнучких методів.

Предмет дослідження: ітераційна (гнучка) модель життєвого циклу SCRUM у великих проектах.

Методи дослідження: Для досягнення поставленої мети застосовувались такі методи:

- методи збору даних: інтерв'ю, анкетування.
- методи аналізу даних для визначення потреб у координації команд.

Наукова новизна отриманих результатів. Наукова новизна полягає у вирішенні задачі координації роботи команд у великих проектах з гнучкими методиками розробки. Під час виконання роботи були отримані результати:

- організації повинні підтримувати необхідну інфраструктуру для місцевої та міжвідомчої координації роботи команд;
- кваліфікаційна робота дозволяє оптимізувати управління проектом з метою максимізації ефективності роботи команд.

Практичне значення отриманих результатів. Команди, що працюють в одному проекті за гнучкими методиками розробки, часто матимуть різні потреби залежно від різних факторів, і тому до них слід ставитися по-різному. Це означає, наприклад, що обов'язкові координаційні або комунікаційні засоби та методи можуть бути не однаково корисними для всіх. Організації повинні, натомість, зосередитись на наданні командам можливості отримувати досвід та вирішувати залежності від роботи інших команд за допомогою відповідних контактів при необхідності. Крім того, для поліпшення підтримки команд організації повинні покращити обізнаність та доступність експертів та підтримку ролей команд.

Апробація результатів та особистий внесок здобувача. Основні положення роботи доповідались, розглядались та обговорювались на науковій конференції Тернопільського національного технічного університету. Результати кваліфікаційної роботи опубліковані у тезах студентської наукової конференції, яка проводилась у ТНТУ у грудні 2020 року.

РОЗДІЛ 1

ОГЛЯД SCRUM-МЕТОДОЛОГІЇ ТА ПРОБЛЕМИ ПРИ РОБОТІ НАД ВЕЛИКИМИ ПРОЄКТАМИ

1.1 Основні поняття гнучкої методології Scrum

Scrum стосується практично будь-якої організації та компанії, що займається розробкою ПЗ. Методи Scrum допомагають компаніям виводити свою продукцію на ринок швидше, ніж будь-коли раніше.

Структура Scrum виникла в 1995 році. Scrum є частиною всеохоплюючої системи Agile. Однак це, звичайно, не єдина частина.

Agile не призначає кроків, яких слід дотримуватися. Структура Scrum забезпечує легкий фреймворк, який можна застосувати до проекту. На додаток до Scrum, є й інші методології гнучкої розробки, такі як Feature Driven Development або Extreme Programming, які також можна застосувати.

Scrum не обмежується використанням лише програмного забезпечення. Це виявилось вигідним при застосуванні до проектів та бізнес-процесів поза розробкою програмного забезпечення, таких як HR, бухгалтерський облік, продажі, досвід користувачів та управління компаніями.

Перш ніж заглибитися в огляд Scrum, давайте почнемо з пояснення деяких термінів Scrum. У цій частині огляду Scrum будуть описані деякі основні концепції Scrum (див. рис. 1.1).

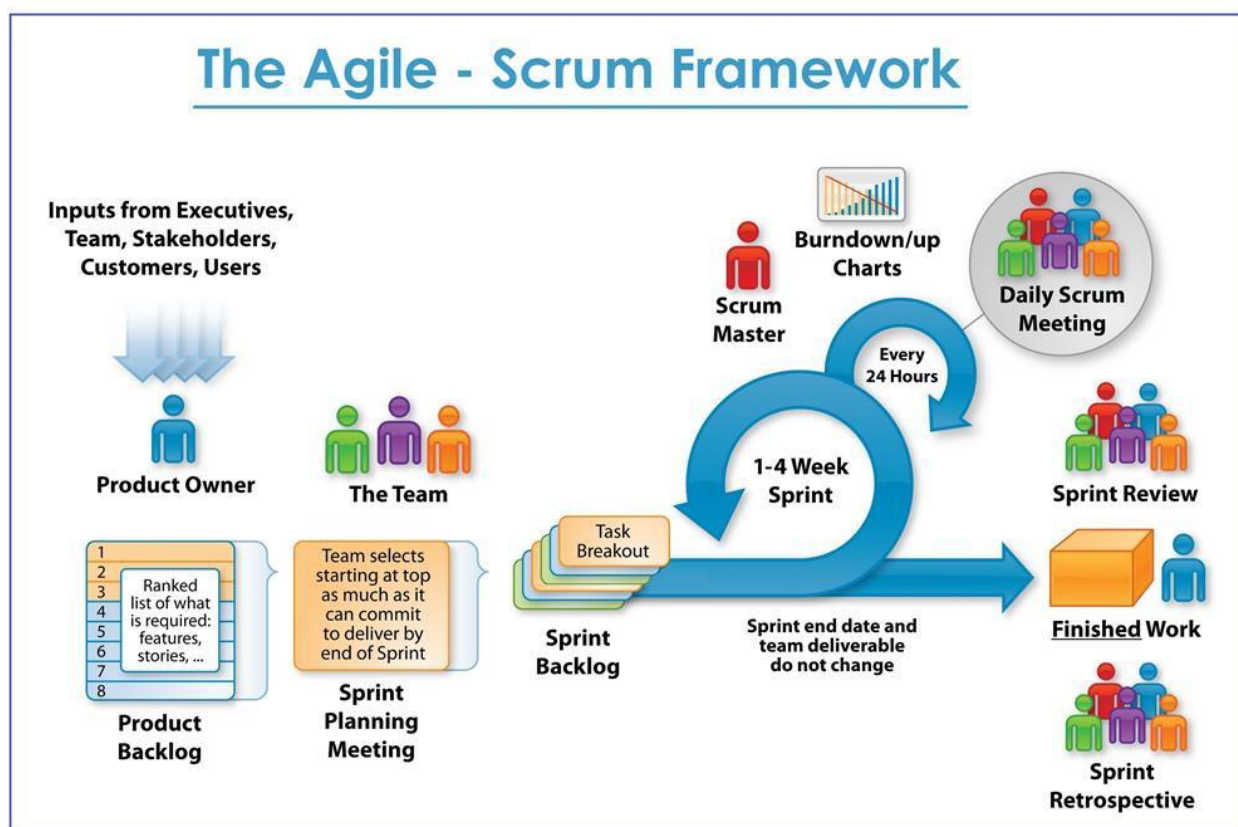


Рисунок 1.1 – Принцип роботи команди згідно Scrum та основні артефакти методології

Спринт (Sprint) – це подія, яка виконує роль контейнера для всіх інших подій у рамках Scrum. Спринт триває, як правило, від двох до чотирьох тижнів, причому тривалість встановлюється на початку Спринту. Кожен наступний спринт негайно слідує за попереднім спринтом.

Власник продукту (Product owner) є ключовим учасником проекту. Він представляє зацікавлені сторони (користувачів, клієнтів та інших), які беруть участь у процесі. Як правило, власник продукту походить із сфери управління товаром або маркетингу. Вони також могли бути або є ключовою зацікавленою стороною.

Існує лише один власник продукту, відповідальний за управління відставанням товару. Основна відповідальність Власника продукту полягає в тому, щоб бути максимізатором вартості для команди Scrum. Власник продукту виконує цю роль, керуючи беклогом та його вмістом. Управління беклогом складається з:

- Елементів, які чітко сформульовані і виражені.
- Пріоритезація елементів беклогу для максимізації вартості та досягнення цілей Власника продукту.
- Забезпечення прозорості т беклогу.
- Співпраця з командою розробників для роз'яснення та допомоги у розумінні питань стосовно беклогу.

Незважаючи на те, що власник продукту несе відповідальність, інші учасник команди можуть допомогти у розробці та обслуговуванні беклогу.

Scrum Master гарантує, що команда максимізує продуктивність. Scrum Master служить координатором, допомагаючи команді застосовувати структуру Scrum. Scrum Master – це керівник команди, який підпорядкований власнику продукту. Scrum Master часто є клеєм, який спочатку утримує команди разом під час навчання та наставництва Scrum команди з метою стати командою з високими показниками.

У рамках Scrum команда розробників повинна мати усі навички, необхідні для завершення проекту. Оптимально в команду входить від трьох до дев'яти людей, за винятком Scrum Master та Власника продукту. Проекти, що використовують Scrum, можуть масштабуватися, щоб включити більше команд.

Беклог продукту – це впорядкований перелік усіх бажаних характеристик товару або змін у продукті. Створення беклогу продукту зазвичай починається з того, що власник продукту створює перелік функцій, які вони хочуть включити до продукту. Власник продукту співпрацює з командою розробників, щоб замовити елементи у беклозі продукту. Беклог продукту є динамічним і розвивається з часом, коли більше дізнаються про товар, технологію, ризики, використання товару клієнтами, нормативні акти та сферу бізнесу.

Елементи, що містяться в беклозі продукту, можуть включати функціональні, вимоги, нефункціональні вимоги, дефекти чи помилки. Елементи, з якими, найімовірніше, незабаром працюватимуть, повинні бути в готовому стані і можуть бути завершені протягом Спринту.

Однією з практик щодо беклогів, яка може суттєво допомогти команді Scrum, є проведення нарад щодо уточнення беклогу. Мета зустрічі полягає в тому, щоб Власник продукту та команда розробників переглянули елементи, записані у верхній частині беклогу, щоб отримати краще розуміння, отримати додаткову ясність, оцінити складність та, за потреби, розділити елементи на менші предмети, які команда Scrum може зрозуміти та реалізувати за один спринт.

Беклог спринта (Sprint Backlog) містить план роботи команди та перелік завдань, які команда повинна виконати під час спринту. Команда розробників є власником Sprint Backlog. Як власник, команда розробників несе відповідальність за реалізацію всіх елементів протягом спринту.

На початку кожного спринту команда проводить збори з планування спринту. На цій зустрічі власник продукту переглядає найпопулярніші елементи беклогу. Члени команди Scrum шукають подальших роз'яснень щодо основних тем та вибирають роботу, яку, на їхню думку, вони можуть передбачити, виконавши під час спринту. Ці елементи беклогу продукту переміщуються у Sprint Backlog. Команда розробників починає визначати завдання та заходи, необхідні для завершення прогнозованої роботи.

Команда Scrum спільно повинна визначити мету Спринту. Мета Спринту – це міра, яку Команда Scrum застосовує до всього, що вони роблять, щоб визначити, що вони роблять і як вони це роблять.

Dayli Scrum meeting – це коротка зустріч до 15 хвилин, що відбувається щодня в один і той же час та під час спринту. Зустріч не є вирішенням проблеми або статусною нарадою. Мета – команда розробників перевіряє завершену та залишену роботу та адаптує план роботи, що міститься у Sprint Backlog, для досягнення цілі спринту. Основними є питання:

1. Що було завершено вчора?
2. Над чим команда буде працювати сьогодні?
3. Чи є перешкоди до завершення роботи?

Це задає контекст робочої доби, що допомагає команді залишатися зосередженою. Усі члени команди повинні відвідувати щоденні збори. Інші, які не належать до команди Scrum, можуть брати участь як спостерігачі.

Sprint Review. Наприкінці спринту команда Scrum та зацікавлені сторони перевіряють роботу, виконану під час проведення спринту на засіданні Sprint Review. Для багатьох Sprint Review – це неформальна демонстрація нових функцій, відзначення успіхів та перевірка невдач. Неформально команда повинна демонструвати робоче програмне забезпечення та уникати використання презентацій PowerPoint. Ідея полягає в тому, що сама презентація не повинна бути завданням, відволікаючи тим самим члена команди від процесу.

Однак Sprint Review – це також можливість для команди Scrum взаємодіяти із зацікавленими сторонами, збираючи відгуки та уявлення про:

1. Цінність, яку становить виконана робота.
2. Майбутнє продукту / проекту.
3. Поняття про бізнес-домен.
4. Поняття про ринкові зміни.
5. Нові / змінені нормативні вимоги.

Ще однією подією, яка відбувається в кінці спринту, є ретроспектива спринту. Це зустріч для всієї команди (власник продукту та Scrum Master), щоб обміркувати, наскільки успішно працював Sprint і як для них працює система Scrum. Теми, які команда повинна розглянути, можуть включати:

1. Як пройшов спринт стосовно людей, стосунків, процесу та інструментів.
2. Оцініть, що працювало, а що ні.
3. Для того, що не спрацювало, визначте способи, які можуть усунути проблеми.
4. Розробіть план впровадження вдосконалень.

Участь у засіданні обмежена членами команди Scrum.

Scrum – це фреймворк, а не методологія. Тим не менше, Scrum часто помилково класифікується як методологія.

Scrum складається з п'яти подій, трьох ролей та трьох артефактів. Невелика кількість правил пов'язує все це разом. Події Scrum:

- Планування спринту.
- Спринт.
- Щоденна зустріч.
- Sprint Review.
- Ретроспектива спринту.

Ролі в Scrum:

- Власник продукту.
- Scrum Master.
- Команда розробників.

Артефакти в Scrum:

- Product backlog.
- Sprint backlog.
- Приріст продукту.

Ключ до Scrum – це Sprint. Sprint – це контейнерна подія, в якій розміщуються всі інші події. На початку Спринту Команда Scrum проводить захід Планування Спринту. Команда перевіряє беклог, прогнозує, які з найбільш пріоритетних пунктів можна виконати під час спринту, а потім визначає, як заповнити кожен з них. Елементи переміщуються з беклогу продукту у беклог спринта. Команда повинна підтримувати спільний план та задавати запитання власнику товару та зацікавленим сторонам, якщо це необхідно.

Відразу після заходу з планування спринту команда починає працювати. Сам Спринт обмежений часом від двох до чотирьох тижнів. Під час спринту команда розробників повинна утримуватися від внесення змін, які можуть перешкодити їм вчасно виконати цілі. Команда може і повинна переглядати предметну область за необхідності, консультуючись із власником продукту за необхідності.

Оскільки команда розробників працює, у неї є щоденна подія, що називається Daily Scrum. Метою Щоденного скраму є перевірити прогрес, здійснений шляхом

реалізації беклогу спринта. Будь-які зміни, які потрібно внести до Sprint Backlog, зазвичай вносяться відразу після щоденної зустрічі. Команда повинна уникати спроб вирішити проблеми в цей час.

До кінця спринту команда Scrum прагне завершити всю роботу. Всі тестування, документування та огляди повинні бути завершені.

Останній день Спринту команда Scrum проводить огляд спринту та ретроспективу спринту. Огляд спринту – це можливість для команди Scrum та зацікавлених сторін переглянути завершену роботу, переглянути те, що очікується у майбутніх спринтах, та поглянути на інші фактори, такі як ринкові та регулятивні зміни, які можуть впливати на майбутні спринти.

Адаптованість Scrum є однією з найбільших переваг. На початку кожного Sprint нова робота вибирається у верхній частині впорядкованого відставання, дозволяючи змінювати напрямок продукту або проекту. Поєднання пристосованості Scrum з потужністю постійного вдосконалення Scrum може дозволити команді швидко перейти від команди з низькими показниками до команди з високими показниками.

1.2 Основні задачі, які вирішуються керівництвом проекту для координації роботи багатьох команд

Agile команди зазвичай складаються з багатьох – не обов'язково жорстко закріплених за членами команд – ролей. Це, як правило розробник/програміст, тестувальник, архітектор, дизайнер взаємодії та менеджер проекту/власник продукту. Щоб усі ці ролі працювали разом у команді, автономні Agile команди часто використовують певні командні практики чи церемонії. Для відстеження прогресу програмного забезпечення використовуються щоденні короткі мітинги, на яких члени команди діляться своїми робочими спостереженнями та можливими перешкодами.

Головними перешкодами успішної роботи команди в гнучких методологіях є відсутність чітких і спільних цілей, відсутність довіри, занадто велика залежність від

інших, відсутність тренерської та організаційної підтримки та різноманітність командних норм.

Впровадження спільної роботи автономних команд над одним проектом є складним, якщо на організаційному рівні є бар'єри. Одним із цих бар'єрів є організаційний контроль. У [7] зазначено, що певні форми детального контролю з боку керівництва гальмують автономність, оскільки вся справа в тому, що команди повинні контролювати себе. У [8] автори стверджують, що саме тут важлива роль керівника проекту, тобто бути координатором між організацією та командою, запобігаючи непотрібним перебоєм.

Дві інші проблеми на організаційному рівні – це спільні ресурси та культура спеціалістів [7]. Розподіл ресурсів між кількома проектами загрожує автономії. Культура спеціалістів – це результат того, що організації підтримують і стимулюють бути найкращими в тому, що робить людина, а не створюють загальних спеціалістів, які можуть виконувати функції один одного (там же).

1.3 Роль команди у великих гнучких Agile проектах

Зовнішня координація є проблемою для автономних Agile команд у великих Agile проектах. Жодна команда не володіє усіма знаннями, необхідними для вирішення складних завдань. Тому командам потрібно координувати роботу з іншими командами та експертами. Далі, коли команди координуються, продукти стають більш зрілими, а процес розвитку змінюється. Згідно [9], основною проблемою координації проекту є те, що такі зацікавлені сторони, як користувачі, замовники, команда розробників та керівництво, повинні бути одночасно задоволеними. А Agile практики не забезпечують комунікаційних механізмів у ситуаціях, коли багато команд беруть участь у одному процесі розробки. Scrum і Kanban призначені для одиночних команд і не призначені для міжкомандного спілкування. Як наслідок, вони не є інструментами для одночасної координації кількох команд чи проектів. Автори в [10] визначили 11 механізмів координації у

масштабному гнучкому проєкті та дійшли висновку, що найважливішими були спеціальні комунікації у вигляді зустрічей та обговорень.

У широкомасштабному середовищі найпоширенішою стратегією управління кількома командами є Scrum of Scrum. Scrum of Scrum – це запланована зустріч, на якій один із членів команди виступає в ролі «посла» для участі у щоденній зустрічі з послами інших команд. Однак Scrum of Scrum виявився неефективним у великих проєктах.

Через труднощі, пов'язані з координацією роботи, було створено ще ряд Agile методик, таких як Large-Scale Scrum (LeSS) [11] та Scaled Agile Framework (SAFe) [12]. У LeSS будь-яка команда або член команди повинен мати можливість і, як очікується, зв'язатися з іншою командою, якщо існує проблема, яку потрібно вирішити (як за допомогою запланованих, так і позапланових зустрічей). LeSS може бути трактована, як підхід знизу вгору до координації і дає автономній команді владу, щоб скоригувати свою діяльність.

The LeSS can be understood as a bottom-up approach to coordination and gives the autonomous agile team authority to adjust practices. The SAFe seems to create a structure with more organizational control, which might leave less flexibility for meetings to emerge and for teams to take the initiative for coordination.

SAFe здається більш формалізованою з більшим організаційним контролем, що зменшує можливості для появи незапланованих зустрічей і взяти на себе ініціативу координації для команд.

В результаті аналізу даних з'явилися дві теми, що описують виклики для автономних команд розробників у великих гнучких проєктах та те, як вони пов'язані з автономією: загальний напрямок та зовнішня координація. Загальний напрямок полягає у створенні спільного керівництва між членами команди та в тому, як встановлюються цілі завдань, що вони спричиняють та як вони повідомляються команді.

Зовнішня координація передбачає координацію команд зі своїм оточенням. Сюди входить те, як команда співвідноситься з іншими командами та компонентами,

як команда працює з додатковими завданнями (наприклад, завданнями, яких, немає у беклозі), як команди спілкуються та як команди координуються із зовнішнім клієнтом.

1.3.1 Загальне керівництво

Аналіз джерел підкреслив важливість створення спільного керівництва для команд у їх масштабному проекті. Пошук спільної сутності полягає також у розумінні завдань, які повинна виконувати команда, і порядку, в якому ці завдання повинні розроблятися.

Далі, щоб зрозуміти завдання та цілі команди, команда повинна зрозуміти потреби кінцевого користувача системи, що розробляється. Однак у великих гнучких проектах команда рідко контактує безпосередньо з кінцевим споживачем. Члени команди обговорювали потреби та проблеми кінцевих користувачів, не маючи на увазі однаковий тип кінцевого користувача, що спричинило непорозуміння в команді. Одна команда вирішила створити набір персон (опис різних типів кінцевих користувачів та їх поведінки), щоб мати можливість краще обговорити та зрозуміти потреби клієнтів у команді. Наявність належної компетенції, спілкування та досвіду в команді сприяє спільному керівництву. Досвідчені члени команди, маючи огляд роботи, можуть допомогти іншим членам команди та підтягнути команду в правильному напрямку.

Хоча розуміння напрямку роботи команди важливо, ми вважаємо, що так само важливо розуміти напрямок масштабного проекту, частиною якого є команда. В рамках конкретного масштабного проекту проект кожної команди складаються з однакових ролей. Однак часто члени одного масштабного проекту не мали однакового розуміння цих ролей.

Як пояснювалось вище, спільні цілі важливі для спільного спрямування. У досліджених масштабних проектах ми виявили, що цілі часто встановлюються керівництвом без залучення команд, цілі часто дорівнюють результатам та термінам,

і члени команди не завжди впевнені, які це цілі. Для кількох команд цілі виявилися еквівалентними термінам виконання.

Під час запитання, чи має команда будь-які цілі, крім конкретних термінів, більшість відповіли, що якщо існують додаткові цілі, вони не знайомі з командою. Одним із пояснень того, чому цілі сприймалися як незрозумілі для команд, може бути те, що їх часто визначав хтось поза командою, не залучаючи членів команди.

Хоча керівники часто не брали участі у визначенні цілей, прямий керівник тієї ж компанії висловив думку, що участь у визначенні цілей є важливою особливістю автономних команд. Хоча залучення було бажаним, члени команди визнали, що вони не можуть бути частиною всіх процесів, пов'язаних з цілями у великому проекті.

1.3.2 Зовнішня координація

У широкомасштабних умовах команди залежать від інших команд, проектів, підрозділів та / або систем всередині організації, і навпаки. Співпраця з командами, які мають однакові системні компоненти, корисна при вирішенні цього питання. Обговорення незрозумілих специфікацій та необхідність підтвердження рішень – це приклади того, коли команді потрібно встановити контакт перед тим, як рухатись далі.

Залежно від того, як інші люди виконують частину роботи, знижується автономія команди, оскільки прийняття рішень обмежене, команда не може повністю контролювати, як виконуються завдання, і їм потрібно адаптувати свої процеси до інших команд та суб'єктів.

Додаткові завдання (завдання, не визначені пріоритетами у sprint backlog або не визначені пріоритетами команди) затримують команди у виконанні їх початкової роботи. Ці не пов'язані між собою завдання можуть навіть відкласти цілі спринти. Тому звертали увагу на зовнішніх акторів, які намагаються змусити команду зробити такі додаткові запитання.

Щоб зменшити проблему, пов'язану із занадто великою кількістю додаткових завдань, більшість команд мають одного члена команди, відповідального за

спілкування з рештою проекту та організації. У більшості команд це керівник команди або технічний керівник. Його основне завдання в цьому випадку – захист команди від зовнішніх впливів.

РОЗДІЛ 2

ОПИС ВІДІБРАНИХ ДЛЯ АНАЛІЗУ ДАНИХ

Ми прагнемо дізнатись і краще зрозуміти фактори, що впливають на потреби команди, зовнішню координацію команди та її ефективність у складних умовах – масштабна розробка програмного забезпечення. Щоб зрозуміти характеристики, які впливають на потреби команд, та співвідношення між задоволенням цих потреб за допомогою зовнішньої координації команди та результатами роботи команди, було проведено багатоадресне дослідження, в якому одиницею аналізу була команда з розробки програмного забезпечення у масштабному розподіленому проекті з розробки програмного забезпечення [13].

Було відібрано дані про два проекти та команди, залучені до їх розробки. Над кожним проектом працювало декілька команд і всі вони використовували в роботі гнучкі технології [14].

2.1 Опис проекту А

Перший випадок – від компанії, яка розробляє загальні програмні продукти, що пропонуються для відкритого ринку, та складні системи з індивідуальними версіями. На момент досліджень компанія А працювала у швидкій розробці майже 6 років, і зараз гнучкі практики стали стандартною практикою.

Досліджуваний проект у компанії А – це розробка та обслуговування підсистеми, що має кілька компонентів та інтерфейси з іншими підсистемами. Підсистема дуже складна і містить кілька мільйонів рядків коду. Компанія зрозуміла, що їй бракує компетенції в певних сферах та виникають проблеми при впровадженні нових функцій у масштабну систему. Розмір та складність системного домену та знань про продукт такі, що для того, щоб стати досвідченим розробником, потрібно кілька років.

Це призвело до переходу до міжфункціональних команд розробників, які відповідають за функцію від опису високого рівня до моменту її випуску замовнику. Кількість розробників, що працюють над підсистемою, зросла з восьми в 2007 році до п'яти команд у Швеції, восьми власних команд та двох груп на аутсорсингу в Китаї та двох корейських команд. Команди з розробки підтримувались додатковими ролями для управління програмами, управління продуктами та управління релізами. Таким чином на проектом працювало 17 самоврядних міжфункціональних командах.

2.2 Опис проекту В

Другий випадок – від компанії, яка розробляє автоматизовані програмні рішення та вбудоване програмне забезпечення для різних галузей. Досліджуваний проект у компанії В являв собою складну систему з безліччю модулів із взаємозалежностями та вимагає інтеграції знань з різних областей та функціональних можливостей. Система містить кілька мільйонів рядків коду і написана на багатьох мовах програмування.

Компанія В робить великий акцент на виробництві та якості процесів, однією з причин є відповідність вимогам безпеки. Проект слідує методології розробки V-моделі. Завдання структуровані в модулі, а командам компонентів призначено ці модулі. Через велику та складність системи хороший розробник може бачити лише половину системи.

Робота була розподілена між двома групами команд в Швеції та однією групою з Індії. У Швеції було шість команд програмного забезпечення, а група розробників в Індії складалась з двох команд. Обидва шведські сайти займалися розробкою системи з самого початку, тоді як індійський приєднався до неї пізніше. Основна шведська організація, яку в результатах називають Швецією 1, відповідає за продукт і в ній працює більшість розробників програмного забезпечення.

Таким чином, програмні проекти зазвичай ведуться з цього місця, а менеджери підпроектів – у кожному. Експертиза в галузі апаратного забезпечення знаходиться у

другому шведському сайті (в результатах називається Швеція 2), і, отже, з цього сайту ведуться проекти з обладнання. Роботи з розробки організовані в рамках проектів, які можуть включати розробку нових поколінь системи, розвиток нових функціональних можливостей, зведення великих проектів технічного обслуговування та проекти чистого технічного обслуговування.

2.3 Збір даних

Для кожної з компаній дослідниками були організовані співбесіди з окремими членами. Ці співбесіди відбувались за напівструктурованим порядком денним, де окремих членів різних рівнів управління безпосередньо запитували про наступне: (i) характеристики проекту для з'ясування типу та складності завдань у досліджуваних проектах; (ii) їх участь у проекті; та (iii) організаційний процес та зв'язки, необхідні командам при роботі над їх завданнями. Також керівники проектів описали свої щоденні завдання.

Показники ефективності є важливим фактором, що враховує групові дослідження. В [16] авторами було встановлено, що координація експертних знань у командах впливала лише на деякі показники ефективності, але не на всі.

Тому ефективність команди вимірювали шляхом оцінки п'яти пунктів:

- якості роботи,
- командної діяльності,
- здатності досягати цілей проекту,
- ступеня досягнення цілей проекту та
- репутації досконалості роботи.

Дані про результати діяльності дали керівники. Незважаючи на те, що ці оцінки носили суб'єктивний характер, а також загрожували побудові валідності, ми вирішили використовувати ці вимірювання для порівняння результативності між командами та компаніями, оскільки об'єктивні показники, такі як рядки коду на людину на місяць або швидкість команди, можуть бути не порівнянними між різними

компаніями і, отже, часто піддаються неправильному тлумаченню. Усі показники ефективності команди вимірювали за 5-бальною шкалою.

Для кожної команди, включеної в дослідження, ми зібрали показники ефективності роботи команди (п'ять пунктів). Короткий зміст зібраних вимірювань результативності представлено у таблиці 2.1.

Таблиця 2.1 – Показники ефективності роботи команд

Команда	A	B	C	D	E	F
Якість роботи	4	5	4	4	4	3
Командні операції	5	3	5	5	4	2
Можливість досягнення цілей проекту	5	4	5	4	5	2
Ступінь досягнення цілей розробки	4	4	4	3	4	3
Рівень зрілості роботи	5	4	5	4	4	1
Ефективність	4.6	4.0	4.6	4.0	4.2	2.2

Аналіз зібраних даних пройшов двоступеневу процедуру. Перший крок полягав у визначенні та кількісному визначенні характеристик командного завдання за допомогою якісного аналізу, проведеного шляхом кодування. Другий крок передбачав аналіз даних опитувальника, в якому ми закодували якісну частину відповідей (зміст координаційних зв'язків), а потім проаналізували координацію команди та зовнішньої діяльності за допомогою методів аналізу соціальних мереж.

2.3.2 Аналіз потреб в координації команд

Ми визначили характеристики командного завдання, проаналізувавши отримані дані. Було встановлено, що існують фактори, які впливають на координаційні потреби команд у двох категоріях:

- 1) координація досвіду та
- 2) координація роботи.

У таблиці 2.2 ми повідомляємо про ці фактори, що впливають на ефективність команди. В результаті процесу кодування ми визначили значення кожного з факторів для кожної команди, як проілюстровано на кроці 2 на рисунку 2.1.

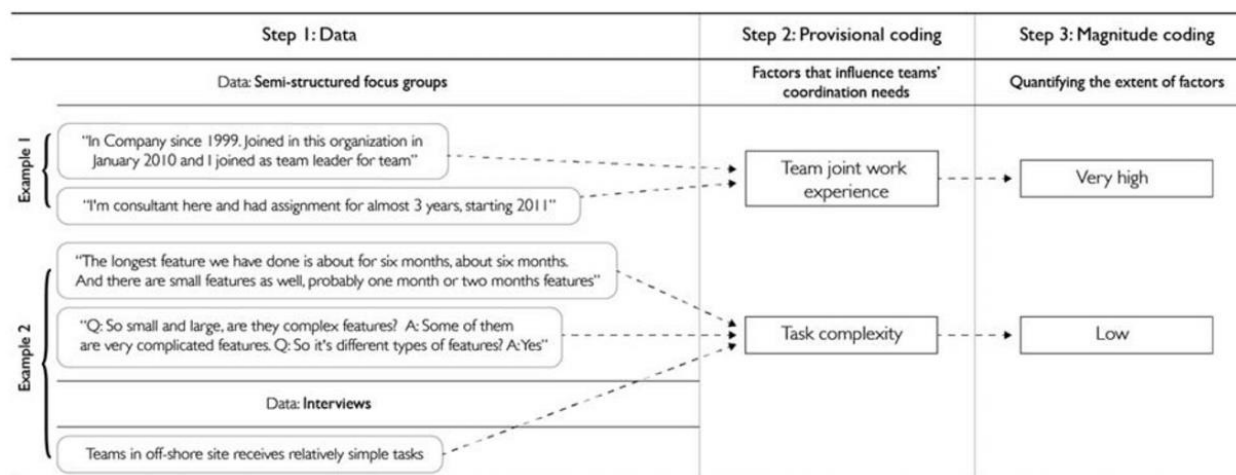


Рисунок 2.1 – Ілюстрація процесу кодування для аналізу координаційних потреб команд

Таблиця 2.2 – Короткий опис характеристик, що впливають на координаційні потреби команд

Координація експертизи		Координація роботи	
Фактор	Вплив	Фактор	Вплив
Знайомство із завданням	Зменшується	Ознайомлення з процесом	Зменшується
За посередництвом:		За посередництвом:	
Досвід членів команди	Зменшується	Стабільність процесу	Зменшується
Спільний досвід роботи команди	Зменшується	Документація процесу	Зменшується
Документація на продукт	Зменшується		
Складність завдання	Збільшується	Складність процесу	Збільшується
За посередництвом:		За посередництвом:	
Взаємозалежність завдання	Збільшується	Взаємозалежність завдання	Збільшується
		Залежність процесу	Збільшується

На другому етапі ми використали кодування за величиною. Цей процес використовується для кількісної оцінки ступеня характерних факторів виконання завдань в окремих командах та опосередковуючих факторів, щоб потім обчислити загальний ступінь характеристичних факторів виконання завдань. Під час складання остаточної величини характеристик завдання ми додали інтерпретацію впливу кожного фактора, як показано на кроці 3 на рисунку 2.1. Результати процесу кодування наведені в наступному розділі в результатах роботи.

2.3.3 Потреба в зовнішній координації команд

Було запропоновано метрику для представлення таких потреб у зовнішній координації і потім ми класифікували кожен взаємозв'язок на дві окремі категорії: координація експертизи та координація роботи, як показано на рисунку 2.2. Результатом цього кроку є класифікація змісту координаційних зв'язків (таблиця 2.3).

На другому кроці ми об'єднали зв'язки, щоб створити набір контактів для координації досвіду та координації роботи для кожної команди в дослідженні.

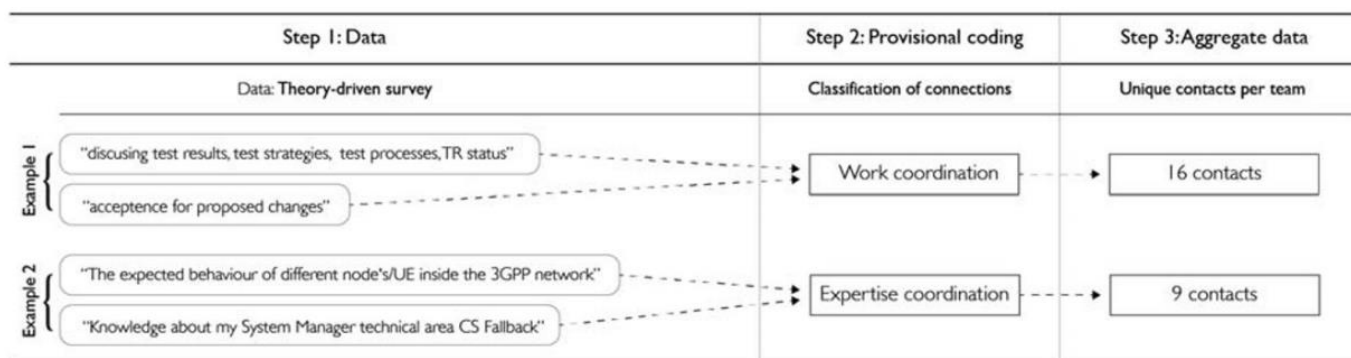


Рисунок 2.2 – Ілюстрація процесу кодування для класифікації контактів команди та зовнішньої координації

На завершальному етапі показано шаблони командної та зовнішньої координації, які візуалізувались за допомогою алгоритму розмітки на основі [16]. Координація експертних знань та координація роботи відображаються різними кольорами, і всі члени команди об'єднуються в єдиному вузлі, щоб підкреслити зовнішні координаційні контакти.

Таблиця 2.3 – Схема класифікації даних, отриманих від координаційних контактів

	Категорія / підкатегорія	Опис	Приклади
Координація експертизи	Технічна інформація	Загальні знання та навички з розробки програмного забезпечення	Навички мов програмування, навички тестування, знання про загальні стандарти безпеки, загальні стандарти та протоколи зв'язку.
	Інформація про домен	Знання про предметну область програми	Досвід роботи в одній області площі, але не обов'язково специфічна для певного продукту. Знання та вміння, які можна передавати між системами або їх можна здобути, не працюючи над специфічною темою.
	Інформація про продукт	Знання про конкретний програмний продукт і конкретні знання про програму	Знання про частини програмних продуктів, стандарти, протоколи та архітектура, властивості коду, тестування, концепція розташування в коді.
Робота по координації	Відгуки про результати роботи команд	Координація результатів роботи команд. Відгуки про проектні пропозиції, інтеграція коду в систему та результати тестування.	Проблеми, відгуки про тестування, результати тестування, пропозиції щодо змін, оглядів, впровадження пропозиції, питання інтеграції.
	Діяльність, пов'язана між собою	Вирівнювання способів роботи при декількох командах	Способи роботи з навколишнім середовищем / інструментами, включаючи тестування, способи роботи з доставкою продукту, правила проектування, поради та підказки, узгодження командних процесів.
Адміністративна інформація	Інструкції	Формальні вимоги до організації роботи.	Що робити, графіки, терміни, пріоритети від лінії.

	Категорія / підкатегорія	Опис	Приклади
	Звітність	Офіційне звітування перед лінійною організацією.	Прогрес тренди та оновлення статусу, інформація про етапи проекту та графіки релізів, офіційні політики та процедури.
	Особистий розвиток та методологія	Знання про результати діяльності людей	Soft skills, знання Scrum, Agile, командоутворення (та інші процеси)

2.3.4 Робота команди

У нашому аналізі ми прагнули досягти єдиного балу для результативності команди на основі агрегованого балу ефективності команди [13]. По-перше, було усереднено п'ять пунктів, щоб розробити показник ефективності команди. Короткий зміст вимірювань продуктивності представлений у таблиці 3.1. Для перевірки внутрішньої узгодженості вимірювань продуктивності було використано методику, запропоновану у [17].

РОЗДІЛ 3

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У цьому розділі ми спочатку описуємо кожну команду та їх мережі за двома категоріями: координація експертизи та координація роботи. По-друге, ми звітуємо про координаційні потреби команд у знаннях та координації роботи. Нарешті, ми демонструємо взаємозв'язок між координацією команди та зовнішніми результатами команд.

3.1 Моделі зовнішньої координації команд

Ми підсумували координаційні контакти (унікальні командно-зовнішні координаційні контакти для команди) та координаційні зв'язки (загальна кількість повідомлених зв'язків від усіх членів команди). У таблиці 3.1 показано дані, що відображають координацію команд на основі аналізу даних, зібраних від працівників, для двох заздалегідь визначених категорій: (1) координація експертизи та (2) координація роботи.

Було встановлено, що контакти з координації експертних оцінок на команду становили від 4 до 20 контактів (середнє = 9,78, SD = 4,87) та 4 – 28 з'єднань (середнє = 14,22, SD = 6,48) на команду. Це свідчить про відносно великі перекриваються зв'язки для координації експертизи, що вказує на те, що члени команди індивідуально покладаються на тих самих експертів. Що стосується координації роботи, було встановлено, що на команду припадало 2 – 30 контактів (середнє = 11,78, SD = 7,74) та 2 – 35 зв'язків (середнє = 13,56, SD = 8,88) на команду. Примітно, що зменшується рівень надлишковості у контактах з питань координації роботи.

Результати показують, що існує велике стандартне відхилення в робочих координаційних контактах та зв'язках, а це вказує на те, що одним командам доводиться частіше керувати робочими залежностями порівняно з іншими.

Таблиця 3.1 – Дані зовнішньої координації між командами

	Ко-манда	К-сть учасників	Мережа контактів				Мережа з'єднань	
			Експертиза	Розмір	Процес	Розмір	Експертиза	Робота
Компанія А	А	6	10	1,67	30	5.00	14	35
	В	5	6	1.20	14	2,80	9	14
	С	5	4	0,80	9	1.80	4	10
	Д	8	6	0,75	12	1,50	17	13
	Е	5	12	2.40	13	2,60	16	13
	Ф	6	7	1.17	2	0,33	14	2
Компанія В	Г	5	13	2,60	7	1,40	13	9
	Н	5	10	2,00	8	1,60	13	12
	І	4	20	5.00	11	2,75	28	14
		Середнє	9,78	1.95	11,78	2.20	14.22	13.56
		Відхил.	4.87	1.32	7,74	1.31	6.48	8,88
		Min	4	0,80	2	0,33	4	2
		Max	20	5.00	30	5.00	28	35

На рисунку 3.1 показані графіки для трьох команд стосовно окремих моделей зовнішньої координації. Кольорами виділено зв'язки з координацією досвіду (експертизи) та зв'язки з координацією роботи.

Положення контакту на зовнішньому графіку координації пов'язане з частотою спілкування – чим ближче до команди, тим частіше спілкування. Команди А – Ф – це спеціальні команди зі справи А, де команди Д – Ф – офшорні команди. Команди Г – І – складові команди у випадку В, де команда Г – команда з іншого шведського місця. Ми спостерігали закономірність для багатьох випадків багаторазового підключення до окремих контактів, що свідчить про велику надмірність контактів (команда Д та команда І). Ми також спостерігали закономірність постійних зв'язків із великою кількістю контактів, що може свідчити про те, що команда шукає конкретні знання або працює над дуже взаємозалежними завданнями (Команда А).

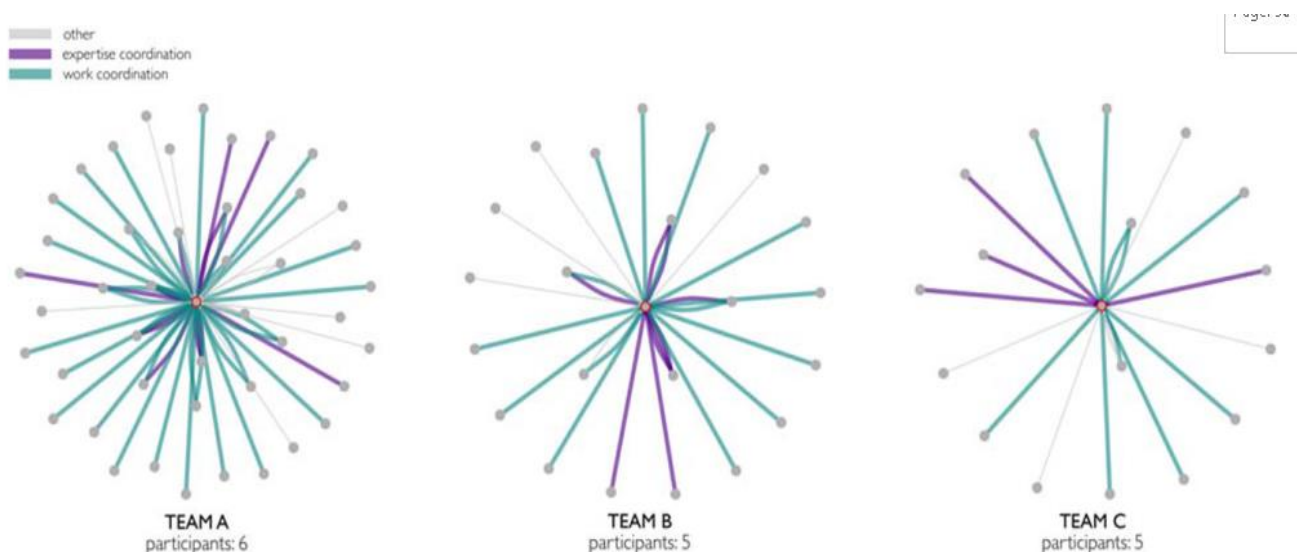


Рисунок 3.1 – Схема координації команди та зовнішньої координації

Порівнюючи інтенсивність координації досвіду та координації роботи, ми виявили відмінності між двома компаніями. У компанії А було визначено, що всі команди (крім команди F) мають більше контактів для координації роботи, ніж для координації експертних знань.

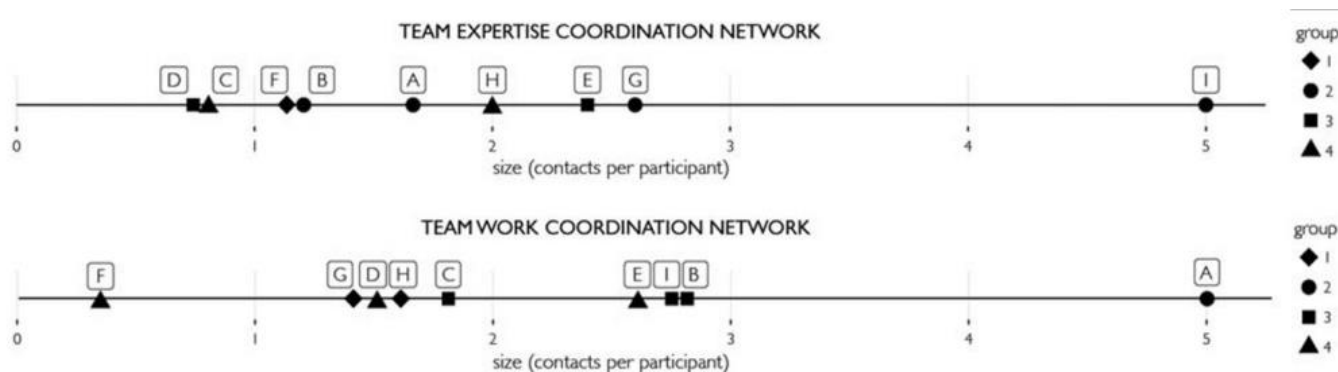


Рисунок 3.2 – Зовнішня координація роботи команди на одного її учасника

У компанії В було встановлено протилежне. Наприклад, команда А та команда D демонстрували велику кількість зв'язків з координації роботи порівняно із зв'язками з координацією експертних знань, тоді як команда Н та команда І демонстрували протилежне. Команда D і команда F демонструють відносно велику надлишковість у зв'язках (це означає, що багато членів команди повідомляли про однакові контакти поза командою), тоді як команда А і команда G демонстрували протилежне. На

рисунку 3.2 ми ілюструємо зовнішню координацію команди на основі кількості контактів на одного респондента. Тут показані відносні відмінності в координації між командою та зовнішніми діями, і видно, кількість контактів на одного респондента для координації експертизи є більш рівномірною (вісім з дев'яти команд мають розмір мережі [контакти на одного учасника] від 0,8 до 2,6), тоді як кількість контактів на одного респондента для координації роботи є більш різноманітною (вісім із дев'яти команд мають розмір мережі [контакти на одного учасника] від 0,33 до 2,8). Середній розмір кількості контактів на одного респондента становить 1,95 (координація експертизи) та 2,20 (координація роботи).

3.2 Профілі команд та їх координаційні потреби

Тут ми наведемо профілі команд з урахуванням потреб у координації по двох категоріях: (1) координація експертизи (у таблиці 3.2) та (2) координація роботи (у таблиці 3.3). Вони засновані на якісному аналізі стенограм фокус-груп, під час якого ми кодували дані із заздалегідь визначеними характеристиками завдання (як описано у розділі 2).

Таблиця 3.2– Командний аналіз потреб у координації експертизи

Команда	Компанія А						Компанія В		
	А	В	С	Д	Е	F	Г	Н	І
Знайомство із завданням	Низький	Помірний	Високий	Помірний	Помірний	Дуже низький	Помірний	Високий	Помірний
Досвід членів команди	Дуже високий	Дуже високий	Дуже високий	Низький	Низький	Дуже низький	Дуже високий	Високий	Дуже високий
Досвід спільної роботи команди з іншими	Високий	Помірний	Високий	Помірний	Помірний	Дуже низький	Дуже високий	Високий	Дуже високий
Документація на продукт	Низький	Високий	Високий	Високий	?	Високий	?	Низький	Низький
Складність завдання	Високий	Дуже високий	Високий	Низький	Помірний	Дуже низький	Високий	Помірний	Високий
Взаємозалежність завдань	Дуже високий	Високий	Високий	Низький	Низький	Дуже низький	Низький	Низький	Високий

У таблиці 3.2 ми повідомляємо фактори, що впливають на координаційні потреби команд у координації експертизи: (i) ознайомлення із завданнями, доповнене трьома опосередковуючими факторами: досвід спільної роботи, досвід спільної команди та документація на продукцію; та (ii) складність завдання, доповнена одним опосередковуючим фактором: взаємозалежність завдання.

Таблиця 3.3– Командний аналіз потреб у координації роботи (процесів)

Команда	Компанія А						Компанія В		
	А	В	С	Д	Е	Ф	Г	Н	І
Розуміння процесу	Низький	Помірний	Помірний	Помірний	Помірний	Низький	Високий	Високий	Помірний
Стабільність процесу	Низький	Низький	Низький	Помірний	Помірний	Помірний	Високий	Високий	Високий
Документація процесу	Низький	Низький	Низький	Помірний	Помірний	Помірний	Високий	Високий	Високий
Складність процесу	Дуже високий	Високий	Високий	Помірний	Помірний	Низький	Помірний	Помірний	Високий
Взаємозалежність завдання	Дуже високий	Високий	Високий	Низький	Низький	Дуже низький	Низький	Низький	Високий
Залежність процесу	Високий	Високий	Високий	Помірний	Помірний	Помірний	Високий	Високий	Високий

У таблиці 3.3 ми повідомляємо про фактори, що впливають на потреби команд у координації роботи: (i) знайомство з процесом, доповнене двома опосередковуючими факторами: стабільність процесу та документація процесу; та (ii) складність процесу, доповнена двома факторами-посередниками: взаємозалежність завдань та залежності процесу.

Щоб зрозуміти подібність та різницю між різними командами та потребами їхніх команд у координації, на рисунку 3.3 ми побудували всі команди на основі двох факторів характеристик завдань для кожної з двох категорій: (1) Потреби у координації експертизи визначаються ознайомленістю із завданнями та складністю завдання та (2) потреби в координації роботи визначаються знайомством із процесом та складністю процесу.

Ми розрізнили чотири різні квадранти характеристик завдання (поєднання низького / високого у двох факторах характеристик завдання). Виходячи з аргументу,

що команди в одному квадранті виявляють потреби подібних команд у зовнішньо-командній координації, ми згрупували ці команди разом, щоб представити якісний підсумок даних про характеристики завдання та їх вплив на координаційні потреби команд.

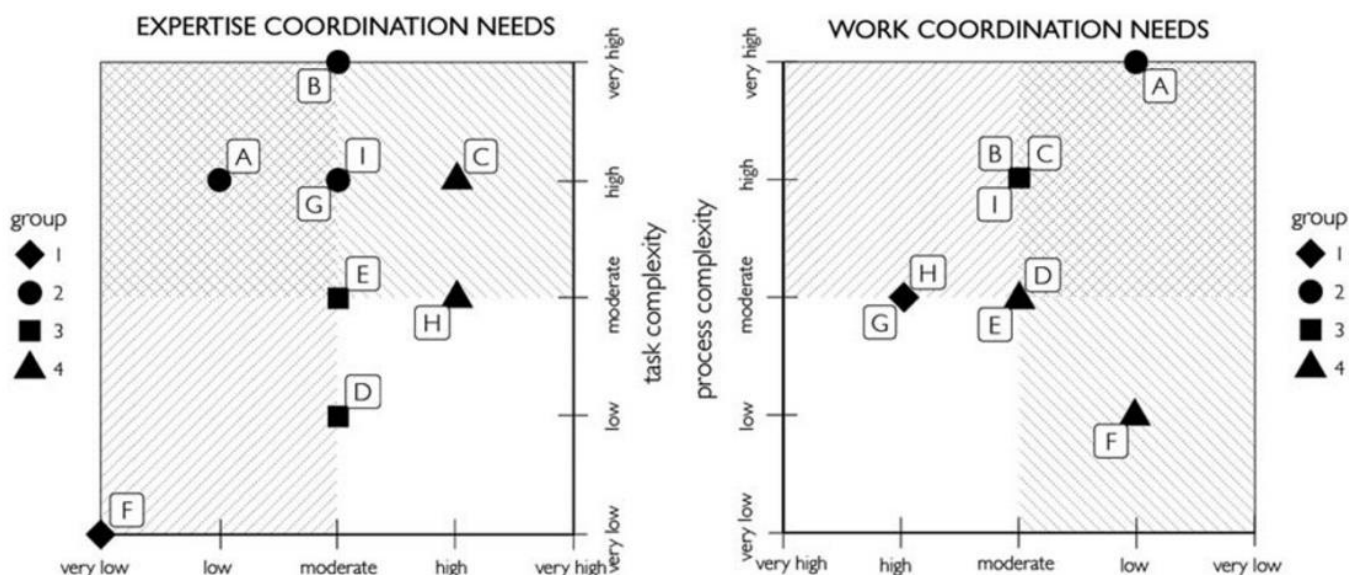


Рисунок 3.3 – Групування команд на основі їх потреб в координації роботи

Різні форми на рисунку 3.3 представляють приналежність до певної групи. Ми визначили чотири різні групи команд у кожній з двох категорій. Далі опишемо групи в кожній категорії, подібність характеристик завдань у кожній групі та відмінності в характеристиках завдань між групами.

У першій з чотирьох груп, як показано на рисунку 3.3 (ліва матриця), це команди С та Н з помірними потребами в координації експертних знань, і команди в цій групі мають високу знайомість із завданнями та помірні фактори складності завдань. У другій групі – (команда D та команда E) низькі потреби в координації знань; ці команди мають завдання з помірною складністю завдань та високою або помірною обізнаністю. Третя група включає команду F з помірними потребами в координації досвіду. Команда отримує завдання з низькою складністю завдань і має низьку обізнаність із завданнями. У четвертій групі – (команда A, команда B, команда G та команда I) з високими потребами в координації знань; ці команди доручаються

завданням із високою або дуже високою складністю завдань, і їм часто доводиться працювати з незнайомими завданнями.

Для обговорення груп ми дотримуємося однакової структури для всіх груп і спочатку повідомляємо про досвід членів команди, приналежність до них, присутність експертів, потім про "втрату" членів або стабільність команди, характеристики завдань і, нарешті, характеристики команди – координація зовнішньої експертизи. Підсумок цих результатів наведено в таблиці 3.2.

3.2.2 Група 1 (команда С і команда Н)

Команди цієї групи виявляють помірні потреби у координації досвіду. Команди отримують відносно звичні завдання із середньою та високою складністю, але потреби у координації знань є помірними, оскільки команди в цій групі мають високий рівень компетентності.

Команди С та Н у цій групі мають високий досвід (у середньому 17,4 та 8 років відповідно) та високий досвід спільної роботи (середня приналежність до команди відповідно 2,4 та 4,4 роки). Обидві команди оцінюють свої існуючі технічні компетенції та знання продукту в команді як добрі, що означає, що команди, як правило, мають досвід, необхідний для вирішення своїх завдань.

Примітно, що обидві команди також мають членів-експертів. Це полегшує ідентифікацію відповідної особи, щоб за потреби звернутись за допомогою за межами команди. Крім того, в обох командах є або члени, що працюють за сумісництвом, або стикалися з нещодавніми змінами в членстві в команді. Двоє з п'яти членів команди в команді Н відносно нові. Час від часу робота в незнайомих областях системи та періодичне збільшення складності завдань є причинами пошуку знань поза командою.

Команда С має невелику кількість контактів з питань координації зовнішньої експертизи з лише чотирма зареєстрованими контактами (0,80 контактів на кожного учасника), а команда Н має найменшу кількість контактів з координації зовнішньої експертизи команди (2,00 контакти на одного учасника).

3.2.3 Група 2 (команда D та команда E)

Команди в цій групі низькі потреби в координації досвіду. Це пов'язано з тим, що ці команди мають достатній досвід роботи та отримують завдання із низькою та середньою складністю.

Команда D та команда E мають у середньому лише 2 та 2,8 роки досвіду роботи у проекті відповідно. Спільний досвід роботи в цих командах є помірним (середня приналежність до команди становить 2 для D та 2,2 роки для команди E). Обидві команди оцінюють свої існуючі технічні компетенції та знання як помірні (в середньому). Однак досвід серед членів команди дещо інший: деякі дуже сильні, деякі дуже слабкі. Як правило, команди погоджуються з тим, що вони мають ресурси, необхідні для вирішення своїх завдань.

В обох командах є члени, що працюють за сумісництвом, або деякі з них приєдналися зовсім недавно, або зазнали частих змін у складі команди. Такі зміни та неповний робочий день не дозволяють створити пул стабільних знань всередині команди. Це збільшує потребу шукати знання за межами команди. Обидві команди отримують відносно прості завдання, невеликі функції та звіти про проблеми. Однак команда G також робить сильний акцент на колективній роботі, що обмежує їх координацію з іншими.

Команда E повідомила про 12 різних контактів з координації зовнішньої експертизи команди (2,4 контакти на учасника), тоді як команда D повідомила про шість контактів координації зовнішньої експертизи групи (0,75 контактів на кожного учасника).

3.2.4 Група 3 (команда F)

У цій групі (тобто в команді F) потреби в координації знань є помірними. Незважаючи на те, що команда F отримує завдання з низькою складністю, ця команда відносно нова і недосвідчена, і їй все одно потрібно набути більше досвіду для вирішення навіть цих завдань.

Досвід членів команди в компанії є низьким (у середньому 2,2 роки, всі члени команди є новими в організації, за винятком одного з 11-річним досвідом роботи). Спільний досвід роботи команди F дуже низький із середньою приналежністю до команди 0,9 років. Оскільки команда F не має досвіду в проекті, навіть прості завдання створюють потребу в отриманні нових знань про завдання для їх вирішення. Для нових та недосвідчених команд документація на продукцію часто є джерелом потрібних знань, і команда F вважає, що вихідний код є хорошим ресурсом і що існує достатня кількість документації щодо архітектури системи.

Команда F має невелику кількість контактів з питань координації зовнішньої експертизи та сім унікальних контактів (1,17 контактів на кожного учасника).

3.2.5 Група 4 (команда А, команда В, команда G та команда I)

Потреби у координації експертних знань є високими для команд четвертої групи. Усі команди мають досвід і отримують відносно незнайомі завдання з великою складністю. Команди мають великий досвід роботи в організаціях (в середньому 10,8, 12,1, 17,2 та 11,3 року відповідно), а спільний досвід роботи для всіх команд також високий (середня приналежність до команди становить 3, 9,1, 6,1). Члени команди мають багато технічних компетенцій та знань про продукцію. Однак команди також зазначили, що їм потрібен ще більший досвід. Усі команди в цій групі були відносно стабільними протягом останніх кількох років, за винятком команди В, оскільки дві особи та керівник команди залишили, а один член команди працює за сумісництвом.

Великий досвід часто є причиною того, що на них покладаються складні завдання. Складність завдань вимагає від команд пошуку нових експертних знань у експертів у різних областях. Далі, з точки зору команд, якість документації на продукцію неоднозначна, так що в основному вона добра, але для потрібно багато зусиль для роботи з нею, а це не завжди корисно. Це знову змушує команди більше покладатися на дискусії з експертами з різних галузей продукції.

Дві команди у цій групі (А та G) повідомили про середню кількість контактів з питань координації зовнішньої експертизи з 10 та 13 контактами (1,67 та 2,60

контакти на одного учасника). Команда В повідомила про шість контактів з координації зовнішньої експертизи команди (1,2 контакти на кожного учасника), що могло бути пов'язано з наявністю офіційного експерта в групі. Команда І повідомила про найбільше число контактів з питань зовнішньої координації експертів у нашій вибірці з 20 унікальних контактів (5,00 контактів на учасника).

3.3 Потреба команд у координації роботи

У першій з чотирьох груп, як показано на рисунку 3.3 (права матриця), це команди (команда G та команда H) з низькими потребами в координації роботи, оскільки ці команди знайомі з процесом та роботою над завданнями, які є відносно ізольованими. У другій групі – (команда D, команда E та команда F) з помірними потребами в координації роботи, оскільки завдання не мають багато взаємозалежностей. У третій групі – (команда B, команда C та команда I) є висока потреба в координації роботи. Ці команди працюють із взаємозалежними завданнями та в середовищі, де слід виконувати ряд процедур (висока складність процесу). У четвертій групі знаходиться команда A, яка має надзвичайно високі потреби в координації роботи, оскільки ця команда працює над завданнями з дуже високою взаємозалежністю завдань.

Для обговорення цих груп ми дотримуємося однакової структури для всіх. Спочатку ми повідомляємо про знайомство команди з процесом та стабільність процесів та якість документації процесу в проекті. Потім ми обговорюємо складність процесу з точки зору двох факторів – взаємозалежності завдання та залежності від процесу, і, нарешті, розглядаються характеристики координації роботи команди та зовнішньої роботи. Підсумок цих результатів наведено в таблиці 3.3.

3.3.1 Група 1 (команда G та команда H)

У цій групі потреба в координації роботи низька. Обидві команди в цій групі є представниками проекту В, де процеси в часі були стабільними, і, отже, команди добре знайомі з процесами.

Обидві команди в групі є представниками проекту В, який дуже довго слідував процесу розробки V-моделі. В обох командах є досвідчені та знайомі з процесами члени. Процес добре задокументований, оскільки всі команди повинні дотримуватися вказівок щодо розробки програмних систем, що мають важливе значення для безпеки.

Для обох команд у цій групі складність процесів є помірною, оскільки команди працюють над призначеними підсистемами і таким чином вирішують відносно незалежні завдання. Однак залежності від процесів високі. Обидві команди мають залежності на рівні завдань, оскільки для виконання завдань потрібні обов'язкові домовленості з інженерами з питань безпеки, керівних принципів, документів з найважливіших причин безпеки та залежностей на рівні системи. Це тому, що групи працюють над складним продуктом, який передбачає взаємодію з апаратним забезпеченням.

В обох команд є відносно невелика кількість контактів з координації роботи з зовнішньою командою, оскільки команда H повідомила про 8 контактів (1,60 контакту на кожного учасника), а команда G повідомила про 7 контактів (1,40 контакти на кожного учасника).

3.3.2 Група 2 (команда D, команда E та команда F)

У цій групі потреби в координації роботи помірні. Команди працюють із відносно ізольованими завданнями, і всі команди приходять з майданчика компанії А, де на момент дослідження стабільність процесу була помірною. Команди в цій групі відносно нові. Отже, знайомство з процесом є від помірного до низького, оскільки їх знання про способи роботи також відносно низькі, особливо у випадку з командою F. Команди встановили, що якість технологічної документації є середньою і що цей фактор зменшує потреби у координації роботи.

Складність процесу для всіх команд цієї групи є низькою. Порівняно недосвідчені команди проекту А отримують завдання з меншою кількістю взаємозалежностей. Залежності процесів відносно помірні, оскільки командам, як правило, потрібно координувати роботу з технічними експертами (яких їм потрібно залучити для затвердження рішень) та тестувальниками. Існують вимоги до рішень щодо взаємодії з такими експертами, як системні менеджери, команди управління проектуванням, групи технічного обслуговування, тестувальники та менеджери з інтеграції. Іноді команді F важко домовитись про реалізацію завдань з технічними експертами, що призводить до повторного виконання неправильно виконаних завдань.

Команда D повідомила про 12 контактів (1,5 контакти на кожного учасника), а команда E – 13 контактів із зовнішньої координації роботи (2,60 контакта на кожного учасника), що в нашій вибірці є середнім. Однак команда F повідомила лише про два контакти з координації роботи з зовнішньою командою (0,33 контакти на учасника), що є низьким значенням.

3.3.3 Група 3 (команда B, команда C і команда I)

Команди цієї групи виявляють високі потреби в координації роботи. Хоча всі команди в цій групі є старшими командами в компанії, вони також працюють над завданнями з великою кількістю взаємозалежностей і повинні дотримуватися декількох і складних процедур.

Знайомство з процесом для всіх команд цієї групи є помірним. Однак причини різні. Як вже згадувалося раніше, проект B дотримується V-моделі, і процеси були стабільними, але команда I була залучена до нового підпроекту розробки з різними способами розробки. Команда B і команда C зазнають труднощів через часті зміни процесів на їх веб-сайті.

Складність процесу для всіх команд цієї групи висока. Команда B та команда C з проекту А працюють в офісі та отримують різноманітні та взаємозалежні завдання в різних частинах системи. Команда I має більше координації зовнішньої роботи з

іншими, оскільки вона відповідає за модуль, який взаємопов'язаний з іншими підсистемами. Робота над підсистемами також призводить до більшої залежності процесів, оскільки зазвичай вона вимагає від команди закріпити свої рішення з експертами поза командою, такими як технічні експерти та архітектори в різних підсистемах. Підхід до координації роботи в цих командах дещо відрізняється.

Команда С та команда В зазначили, що вони більше покладаються на своїх експертів у команді для вирішення залежностей процесу, а команда С повідомила про дев'ять контактів з координації роботи з зовнішньою командою (1,80 контакту на кожного учасника). Однак команди проекту В повідомили про середнє значення 14 для зовнішніх контактів з координації роботи (2,80 контакту на кожного учасника). Команда І зазначила, що вона активно координує роботу з іншими, але повідомила про 11 контактів з зовнішньої координації (2,75 контактів на кожного учасника).

3.3.4 Група 4 (команда А)

Команда А має дуже високі потреби в координації роботи. Це старша команда в компанії. Вона відрізняється від команд групи 3 тим, що отримує завдання, які мають ще більше залежностей від інших підсистем та рівнів.

Знання процесів для команди А є низьким, оскільки вони зазнають труднощів із нещодавніми змінами в процесах на їхній ділянці, оскільки головний офіс наполягає на впровадженні нових процесів у проект. Наприклад, Команда А також бере участь у впровадженні багатьох змін для пілотних досліджень та випробувань.

Складність процесів для команди А дуже велика, оскільки її члени працюють над завданнями, які мають дуже високу взаємозалежність. Команда А повідомила про дуже велику кількість контактів з координації роботи з зовнішньою роботою команди з 30 унікальними контактами для координації роботи (5,00 контактів на учасника), до якої також входить багато неформальних експертів та колишніх членів проекту.

3.4 Вплив рівня зовнішньої координації на результатами роботи команд

Щоб зрозуміти взаємозв'язок між задоволенням потреб команд у зовнішній координації та яким чином це пов'язано з результатами роботи команд, було співставлено обговорені раніше висновки з даними про ефективність команд (див. рис. 3.4). На рисунку по осі Y показано потреби відповідних команд в координації експертних знань та необхідність координації роботи. На осі X показано відповідні зовнішні координаційні контакти для двох категорій: координація експертизи та координація роботи. Метрика – це кількість контактів на одного члена команди, щоб нормалізувати відмінності між різними командами, і, таким чином, порівняти їх. Коли команди розташовані по діагоналі, це означає, що фактична кількість координаційних контактів може потенційно задовольнити потреби в знаннях та координації роботи (чим вищі потреби команд, тим більша кількість командно-зовнішніх координаційних контактів).

Ефективність візуалізується за принципом теплової мапи. Кожна кольорова точка на малюнку – це одна команда, а колір вказує на ефективність команди від 1 (червоний) до 5 (зелений). Задоволення потреб команд в координації експертизи та процесів призведе до кращої роботи команди, тоді як незадоволені потреби призведуть до недостатньої ефективності команди.

Обидві потреби – у знаннях та координації роботи – повинні бути задоволені для досягнення кращих результатів у команді. Результати аналізу потреб команд у координації досвіду показують, що дві з чотирьох вискоефективних команд знаходяться на діагоналі або вище (Команда I та Команда G). Однак дві команди (Команда A і Команда C) знаходяться нижче діагоналі, і хоча вони повідомили про невелику кількість контактів з питань координації, вони працюють дуже добре.

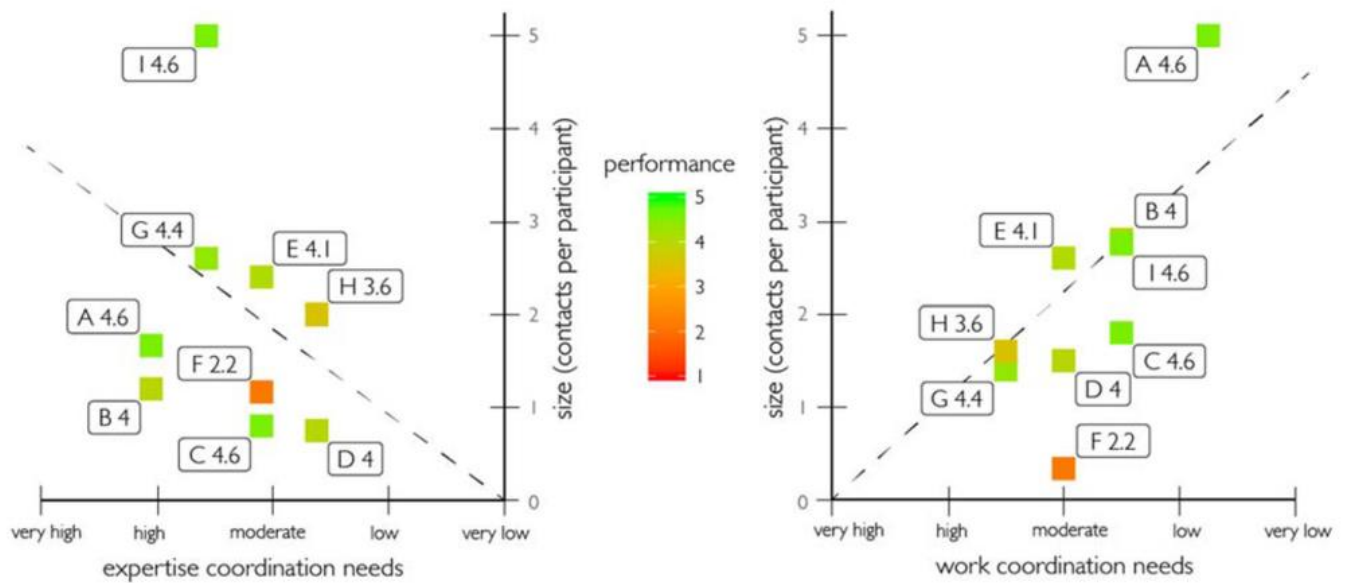


Рисунок 3.4 – Ефективність роботи команди проти координаційних потреб досвіду та процесів

Усі чотири команди, які мають середні показники ефективності, знаходяться на або трохи нижче діагональної лінії. Команда F має дуже низькі показники, і вона також розміщено нижче лінії. Результати щодо потреби команд у координації роботи не демонструють чітких індивідуальних закономірностей. Команда F має дуже малу кількість контактів з питань координації, а також вона має недостатні результати. Команда A повідомила про надзвичайно велику кількість контактів з координації роботи з зовнішніми командами, і її продуктивність вище діагоналі.

Загалом команда E, G та I перебувають над лінією в категорії координації експертних знань та вище або близько до лінії в категорії координації роботи, що вказує на те, що ці команди можуть реалізувати свої потреби. Команда A знаходиться нижче рівня в категорії координації експертних знань, але високо над лінією в категорії координації роботи. Команда B знаходиться нижче лінії в категорії координації експертних знань, але наближається до лінії в категорії координації роботи та має добрі результати. Команда H дуже близько до лінії в обох категоріях і показала середні показники. В обох категоріях під лінією три команди. Команда D трохи нижче в обох категоріях потреб координації та демонструє середню

ефективність, тоді як команда F значно нижче в категоріях потреб координації обох команд і демонструє слабку ефективність. Це свідчить про те, що потреби їхніх команд у координації не виконуються. Хоча команда C також знаходиться за межею в обох категоріях потреб координації, ця команда оцінюється як дуже успішна.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Впровадження в Україні світового досвіду щодо покращення умов і безпеки праці в ІТ-компаніях

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. З метою належного правового забезпечення необхідно розширити та доповнити перелік основних професій комп'ютерної галузі у національному класифікаторі ДК-003-2010, а також підготувати відповідний випуск у кваліфікаційному довіднику посад фахівців ІТ-індустрії, що сприятиме вирішенню питань їх соціального захисту, пенсійного забезпечення, атестації робочих місць основних професій за умовами праці на предмет подальших певних видів пільг та компенсацій за важкі шкідливі і небезпечні умови праці. Важливим напрямом стосовно визначення професійної придатності фахівців з інформаційних технологій є проведення психофізіологічної експертизи відповідно до 5 статті Закону України «Про охорону праці».

Робота з комп'ютерами нового покоління характеризується певним психофізіологічними перенавантаженнями, втомую зорового аналізатора, гіпокінезією, відсутність диференційованих норм праці при роботі з новою комп'ютерною технікою в залежності від віку, статі, категорії зорової роботи, режимів праці і відпочинку (протягом робочого дня, тижня, щорічного режиму відпусток). Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів.

Особлива роль з точки зору збереження та відновлення здоров'я працюючих в комп'ютерні галузі належить попереднім та періодичним наглядом з подальшої

психофізіологічної експертизи і встановленням професійної придатності при роботі з комп'ютерами нового покоління, який супроводжується виникненням певних факторів професійного ризику електро-травматизму при їх ремонті та обслуговуванні. В цьому зв'язку необхідне запровадження експертизи на предмет безпечної експлуатації ПЕОМ, тобто офіційне підтвердження фактичних параметрів електробезпеки, їх відповідності вимогам нормативної документації фахівців, які проводять таку експертизу повинні пройти навчання і перевірку знань відповідно до вимог ДНАОП 0.00-8.20-99.

За результатами експертизи повинні прийматися рішення про відповідність ПЕОМ нормам безпеки, терміни чергової експертизи, оформлюються протоколи вимірювань і випробувань, проведені у разі потреби розрахунки та експертний висновок. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії [49].

Заслуговує на увагу зарубіжний досвід створення у приміщеннях та в зоні їх розміщення на територіях підприємств спеціальних візуальних комфортних умов та забезпечення вимог виробничої естетики, дотримання норм рівнів виробничого шуму та акустичної тиші за межами офісу. Також дуже важливим є використання в офісних приміщеннях та кабінетах психофізіологічного розвантаження функціональної музики, яка сприяє попередженню перевтоми і підтриманню необхідного рівня розумової працездатності фахівців комп'ютерної галузі [50], [51].

В цьому напрямі заслуговує на увагу створення при великих центрах інформаційних технологій кімнат (кабінетів) психофізіологічного розвантаження працівників галузі (на 5 місць). Зарубіжний досвід охорони праці при використанні новітніх інформаційних технологій та сучасного комп'ютерного обладнання передбачає з метою попередження наслідків монотонної праці, підвищення рівня рухової активності і покращення розумової працездатності

фахівців ІТ-індустрії під час технологічних перерв участь у спеціальних облаштованих приміщеннях необхідним спортивним інвентарем та різними тренажерами відповідних фізичних вправ, індивідуальних тренінгових завдань відповідно до віку, статі та категорії зорової роботи.

Такий підхід дозволяє зняти надлишкове психофізіологічне перевантаження, підвищити працездатність центральної нервової системи, попередити перевтому зорового аналізатора. Показана ефективність проведення різноманітних за своєю спрямованістю вправ робітників цієї галузі (приблизно на 5-30%).

Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

4.2 Вплив електромагнітного імпульсу (ЕМІ) ядерного вибуху на елементи виробництва та заходи захисту

Ядерні вибухи в атмосфері й більш високих шарах призводять до виникнення потужних електромагнітних полів з довжиною хвиль від 1 до 1000 м і більше. Ці поля через короткочасне існування називають електромагнітним імпульсом (ЕМІ). ЕМІ виникає при ядерному вибусі у воєнний час, у мирний час – при випробуванні ядерної зброї або ядерних аваріях і катастрофах в атмосфері й космосі.

Основною причиною виникнення ЕМІ тривалістю менше 1 с вважають взаємодію гамма-променів і нейтронів ядерного вибуху з атомами газів повітря, внаслідок чого з них вибиваються електрони (ефект Комптона) і хаотично розлітаються в середовищі позитивно заряджених атомів газів. Важливе значення має також виникнення асиметрії в розподілі просторових електричних зарядів, пов'язаних з особливостями поширення гамма-променів і утворення електронів.

Гамма-промені, які випускаються із зони вибуху в напрямі поверхні землі, поглинаються в більш щільних шарах атмосфери, вибиваючи з атомів повітря швидкі електрони, які летять у напрямку гамма-променів зі швидкістю світла, а позитивні іони (залишки атомів) залишаються на місці. У результаті поділу і переміщення позитивних і негативних зарядів у цій області й у зоні вибуху, а також при взаємодії зарядів з геомагнітним полем Землі утворюються елементарні й результуючі електричні та магнітні поля ЕМІ, які досягають поверхні землі в зоні радіусом кількох сотень кілометрів. Виникають сильні поперечні токи і утворюється подібність великої "плоскої антени", яка випромінює потужний ЕМІ з часом наростання порядна 10 нс і тривалістю більше 230 нс; зі смугою частот від 10 кГц до 100 МГц. Залежно від висоти ядерного вибуху за інших однакових умов змінюються характер, інтенсивність ЕМІ і дальність його поширення.

При наземному і низькому повітряному вибуху уражаюча дія НМІ спостерігається на відстані кількох кілометрів від центру вибуху. Під час ядерного вибуху на висотах від 3 до 25 км утворюється симетричне джерело генерації, але радіус поширення ЕМІ залишається обмеженим внаслідок сильного поглинання гамма-випромінювання в щільних шарах атмосфери.

Найбільшу уражаючу дію має ЕМІ, що виникає при екзоатмо-сферному вибуху (більше 40 км). Зі збільшенням висоти вибуху збільшується і район джерела генерації ЕМІ, досягаючи в діаметрі тисячі кілометрів і товщини 20–40 км. Так, під час вибуху на висоті 80 км.

ЕМІ буде поширюватися на площі радіусом 960 км, а під час вибуху на висоті 160 км – на площі радіусом 1400 км. Екзоатмосферний ЕМІ характеризується дуже малим часом наростання (декілька сотень наносекунд), високою інтенсивністю електричного поля (більше 50 кВ/хв) і магнітного поля (близько 130 А/хв). Розряд блискавки порівняно з ЕМІ має значно більшу тривалість зростання і спаду (5–300 мкс), створює дуже потужні поля (близько 100 кВ/хв), несе значно більшу енергію, але спектр частот становить близько 10 МГц, тоді як для ЕМІ він більше – 100 МГц. Пікове значення ЕМІ може досягти 50 000 В/хв, що дорівнює всій енергії

яка випромінюється в радіочастотній частині спектра. Уражаюча дія ЕМІ обумовлена виникненням напруги і струмів у провідниках різної довжини, розміщених у повітрі, землі.

ЕМІ захвачують спектр частот від десятків до кількох сотень мегагерц, тобто діапазон, в якому працюють установки електропостачання, зв'язку і радіолокації. Напруженість електромагнітного поля, створюваного ЕМІ, досягає 50 000 В/м, тоді як у радіолокації вона не перевищує 200 В/м, а у зв'язку – 10 В/м.

У серпні 1958 р. у момент заатмосферного термоядерного вибуху, проведеного США над островом Джонсон, на Гавайях, які знаходяться за 1000 км від епіцентру вибуху, погасло освітлення на вулицях. Це сталося в результаті дії ЕМІ на повітряні лінії електропередач, які відіграли роль протяжних антен.

Величина ЕМІ залежно від ступеня асиметрії вибуху може бути різною – від десятків до сотень кіловольт на метр антени, тоді як чутливість звичайних УДК-приймачів становить кілька десятків або сотень мікрівольт. Так, у разі наземного вибуху потужністю 1 Мт напруженість поля на відстані 3 км становить близько 50 кВ/м, а на відстані 16 км – 1 кВ/м. А у разі заатмосферного вибуху такої ж потужності напруженість поля становитиме тисячі кіловольт на метр площі в кілька тисяч квадратних кілометрів земної поверхні.

Уражаюча дія ЕМІ в приземній області й на землі пов'язана з акумулюванням його енергії довгими металевими предметами, рамними і каркасними конструкціями, антенами, лініями електропередачі та зв'язку, в них виникають сильні наведені струми, які руйнують підключене електронне та інше чутливе устаткування. У районі дії ЕМІ безпосередній контакт людини зі струмопровідними предметами небезпечний.

ЕМІ уражає радіоелектронну і радіотехнічну апаратуру. В провідниках індукуються високі напруги і струми, які можуть призвести до постійних або тимчасових пошкоджень ізоляції кабелів, відключення реле і переривників, пошкодження елементів зв'язку, магнітних запам'ятовуючих пристроїв у ЕОМ і системах передачі даних тощо. Найбільш уразливими елементами обладнання є

напівпровідникові прилади – транзистори, діоди, випрямлячі, інтегруючі кола, цифрові процесори, управляючі й контрольні прилади. Чутливі до пошкодження ЕМІ транзистори звукової частоти, перемикаючі транзистори, інтегруючі кола та ін.

Особливо чутливими до впливу ЕМІ є 6 основних груп об'єктів і систем:

1) системи передачі електроенергії: повітряні ЛЕП, кабельні лінії, різні види з'єднувальних ліній і повітряна електропроводка;

2) системи виробництва, перетворення і накопичення енергії: електростанції, генератори постійного і змінного струму, трансформатори, перетворювачі струмів і напруг, комутатори і розподільні пристрої, електричні батареї і акумулятори, паливні, сонячні й термоелементи;

3) системи регулювання і управління: електромеханічні й електронні датчики та інші елементи автоматики, комп'ютерні установки, мікропроцесори;

4) системи споживання електроенергії: електродвигуни і електромагнітні, нагрівальні, холодильні, вентиляційні, освітлювальні установки та кондиціонери;

5) системи електротяги: електроприводи, напівпровідникові та інші типи перетворювачів;

6) системи радіозв'язку, передачі, зберігання і накопичення інформації: антени, хвилеводи, коаксильні кабелі, електронні прилади, радіопередавачі, радіоприймачі, установки автономного електропостачання, змішувачі, телефонні апарати, телеграфні установки, заземлені кабелі й проводи, АТС.

Найбільш стійкі до ЕМІ вакуумні електронні прилади, які виходять із ладу при енергії 1 Дж. Величина енергії ЕМІ залежить від ширини періоду частот антенних систем.

Більшість систем зв'язку працюють у діапазоні частот від середніх до ультрависоких і будуть пошкодженими залежно від робочого періоду частот. Радіолокаційні системи менше пошкоджуються від ЕМІ, тому що вони працюють у періоді частот, де щільність енергії ЕМІ невелика. Іскріння, яке виникає під впливом високого електричного поля ЕМІ, може спричинити спалахування парів бензину та інших налив у сховищах.

Якщо ядерний вибух стався поблизу лінії електропостачання, зв'язку великої довжини, то наведені в них напруги можуть поширюватися по проводах на багато кілометрів, пошкоджувати апаратуру й уражати людей, які знаходяться на безпечній відстані відносно інших уражаючих факторів ядерного вибуху.

ЕМІ небезпечний і за наявності міцних споруд, розрахованих на стійкість проти ударної хвилі наземного ядерного вибуху, проведеного на відстані кількох сотень метрів.

Сучасний рівень знань про природу і властивості ЕМІ дає можливість розробити захист від нього і впровадити заходи захисту до яких входять схеми, стійкі до електромагнітної інтерференції, радіоелектронні елементи стійкі до ЕМІ, екранування окремих пристроїв або цілих електронних систем.

ВИСНОВКИ

Використання кількох джерел даних (інтерв'ю, фокус-групи та опитування, на яких базується дослідження, дозволило підтвердити потребу у координації роботи команд у великих розподілених проектах. На результати впливає інтерпретація спостережуваних та досліджуваних явищ. Існує ризик того, що висновки також можуть бути пояснені факторами, які уникнули нашої уваги. Одна з причин полягає в тому, що дані на збирались на основі фактичної координації роботи команди з іншими, а покладались на дані, які власноруч задокументовані на основі свідчень учасників проекту. Іншими словами, було досліджено сприйняття координації командами. Отже, досвід та обмін інформацією могли бути перебільшені або недооцінені респондентами.

Моделі координації можуть також змінюватися з часом на основі змін у потребах координації експертних знань та координації роботи, а також змін, що стосуються процесів у проектах та через зміну персоналу.

Інший наслідок цього обмеження пов'язаний з вимірюванням продуктивності. Використовували командні оцінки результативності, які за своєю суттю є суб'єктивними. Ми використовуємо цей тип вимірювання ефективності тому, що немає доступних об'єктивних вимірювань, які можна для вимірювань у кількох компаніях одночасно.

Друге обмеження - використання кодування величини для аналізу потреб команд на основі напівструктурованих даних які є якісними характеристиками. При перетворенні кількісних шкал у якісні, особлива увага приділялася обговоренню зведення даних до єдиного масштабу та необхідності врахування контексту, що враховує конкретний випадок.

Потрібні подальші дослідження з додатковими зразками даних, щоб зрозуміти повну сукупність того, які фактори впливають на координацію команди широкомасштабному розподіленому середовищі. Подальші дослідження, які відбирають більше команд з потенційно різними потребами та моделями зовнішньої

координації між командами, необхідні для подальшого розуміння взаємодії між зовнішньою координацією та ефективністю роботи команди. Нарешті, на узагальнення висновків впливає стратегія відбору проб та випадки, включені в наше дослідження. Були знайдені дані лише стосовно двох масштабних проектів розподіленого програмного забезпечення у двох компаніях.

Команди, що працюють в одному проекті, часто матимуть різні потреби залежно від різних факторів, і тому до них слід ставитися по-різному. Це означає, наприклад, що обов'язкові координаційні або комунікаційні форуми та ритуали можуть бути не однаково корисними для всіх. Компанії повинні, натомість, зосередитись на наданні командам можливості отримувати досвід та вирішувати залежності від роботи за допомогою відповідних контактів, коли це необхідно. Крім того, для поліпшення підтримки команд компанії повинні покращити обізнаність та доступність експертів та підтримку ролей команд. Організації також повинні підтримувати необхідну інфраструктуру для місцевого та міжвідомчого досвіду та координації роботи. Нарешті, наше дослідження показує, що задоволення потреб у координації команд є важливим для досягнення бажаних результатів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Šmite, Darja, et al. "Software teams and their knowledge networks in large-scale software development." *Information and Software Technology* 86 (2017): 71-86.
2. Ihor, Bodnarchuk, et al. "Multicriteria Choice of Software Architecture Using Dynamic Correction of Quality Attributes." *International Conference on Computer Science, Engineering and Education Applications*. Springer, Cham, 2019.
3. Kharchenko, Alexander, Ihor Bodnarchuk, and Vasyl Yatcyshyn. "The method for comparative evaluation of software architecture with accounting of trade-offs." *American Journal of Information Systems* 2.1 (2014): 20-25.
4. Харченко, Олександр Григорович, Василь Володимирович Яцишин, and Ігор Едуардович Райчев. "Інструментальний засіб розробки та комунікації вимог якості до програмних систем." (2010).
5. A Kharchenko, I Bodnarchuk, I Halay, V Yatcyshyn. *An Optimal Trade-off Solution of the Software Architecture Choice Problem // Journal of Information and Computing Science*, 2016. PP. 281 – 290.
6. Ihor, Bodnarchuk, et al. "Multicriteria Choice of Software Architecture Using Dynamic Correction of Quality Attributes." *International Conference on Computer Science, Engineering and Education Applications*. Springer, Cham, 2019.
7. Moe, N. B., Dingsøy, T., and Dybå, T., "Overcoming Barriers to Self-Management in Software Teams," *IEEE Software*, vol. 26, no. 6, pp. 20-26, 2009.
8. Boehm, B. and Turner, R., *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley 2003.
9. Boehm, B. W. and Ross, R., "Theory-W software project management principles and examples," *IEEE Transactions on Software Engineering*, vol. 15, no. 7, pp. 902-916, 1989.

10. Nyrud, H. and Stray, V., "Inter-Team Coordination Mechanisms in Large-Scale Agile," in Proceedings of the Scientific Workshop Proceedings of XP2017, 2017: ACM.
11. Larman, C. and Vodde, B., Large-scale Scrum: More with LeSS. Addison-Wesley Professional, 2016.
12. Leffingwell, D., SAFe 4.0 Reference Guide: Scaled Agile Framework for Lean Software and Systems Engineering. Addison-Wesley Professional, 2016.
13. Stewart DW, Shamdasani PN. Focus groups: Theory and Practice. 20 Sage publications; 2014.
14. Sablis, Aivars, Darja Smite, and Nils Moe. "Team-external coordination in large-scale software development projects." Journal of Software: Evolution and Process (2020): e2297.
15. Faraj, Samer, and Lee Sproull. "Coordinating expertise in software development teams." Management science 46.12 (2000): 1554-1568.
16. Kline RB. Principles and practice of structural equation modeling Guilford publications; 2015.
17. DeVellis RF. Scale development: Theory and applications. 26 Sage publications; 2016.
18. Інформаційна культура підприємств, види інформаційної культури, інформаційне поведіння [Електронний ресурс]. – Електронні дані (Лекції). –Режим доступу: <https://lektsii.com/1-78900.html>
19. Пивоваров М.Г., Медко Д.А. Перспективи створення і розвитку інформаційно-комунікаційної системи України // Економіка: проблеми теорії та практики: Зб. наук. праць. – Вип. 49. –Дніпропетровськ: Дніпропетр. Нац. Ун-т, 2000. – С. 56-61.
20. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин МОЗ України від 10.12.1998 № 7. // Офіційний сайт Верховної Ради України. – [Електронний ресурс]. – Режим доступу <https://zakon.rada.gov.ua/rada/show/v0007282-98>

21. Бідяк О. Профілактика отруєння хлором. // Офіційний сайт управління держпраці в Тернопільській області. – Електронний ресурс]. – Режим доступу <https://te.dsp.gov.ua/profilaktyka-otruyennya-hlorom/>
22. Zhou M, Mockus A, Weiss D. Learning in offshored and legacy software projects: How product structure shapes organization; 2009.
23. Rus I, Lindvall M. Knowledge management in software engineering. IEEE softw. 2002;19(3):26.
24. Sengupta B, Chandra S, Sinha V. A research agenda for distributed software development. In: Proceedings of the 28th international conference on Software engineering; 2006:731-740.
25. Earl M. Knowledge management strategies: Toward a taxonomy. J manag infor syst. 2001;18(1):215-233.
26. Dingsøyr T, Røyrvik E. An empirical study of an informal knowledge repository in a medium-sized software consulting company IEEE Computer Society; 2003:84-92.
27. Dingsøyr T, Djarraya HK, Røyrvik E. Practical knowledge management tool use in a software consulting company. Commun ACM. 2005;48(12):96-100.
28. Nahapiet J, Ghoshal S. Social capital, intellectual capital, and the organizational advantage. Acad manag rev. 1998;23(2):242-266.
29. Akgün AE, Byrne J, Keskin H, Lynn GS, Imamoglu SZ. Knowledge networks in new product development projects: a transactive memory perspective.
30. Infor manag. 2005;42(8):1105-1120.
31. Mathiassen L, Vogelsang L. The role of networks and networking in bringing software methods to practice. IEEE; 2005:256b-256b.
32. Hannay JE, Benestad HC. Perceived productivity threats in large agile development projects ACM; 2010:15.
33. Tiwana A, Mclean ER. Expertise integration and creativity in information systems development. J Manag Inf Syst. 2005;22(1):13-43.

34. Bauer TN, Erdogan B. Organizational socialization: The effective onboarding of new employees.; 2011.
35. Huckman RS, Staats BR. Team familiarity, role experience, and performance: evidence from Indian software services. *Manag Sci.* 2009;55(1):85-100.
36. Lewis K. Measuring transactive memory systems in the field: scale development and validation. *J Appl Psychol.* 2003;88(4):587-604.
37. Ramesh B, Cao L, Mohan K, Xu P. Can distributed software development be agile. *Commun ACM.* 2006;49(10):41-46.
38. Ovaska P, Rossi M, Marttiin P. Architecture as a coordination tool in multi-site software development. *Softw Process: Improve Prac.* 2003;8(4):233-247.
39. Ko AJ, Myers BA, Coblenz MJ, Aung HH. An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *IEEE Trans softw eng.* 2006;32(12):971-987.
40. Andres HP, Zmud RW. A contingency approach to software project coordination. *J Manag Infor Syst.* 2002;18(3):41-70.
41. Dingsøyr T, Moe NB, Seim EA. Coordinating knowledge work in multiteam programs: findings from a large-scale agile development program. *Proj Manag J.* 2018;49(6):64-77.
42. Herbsleb JD. Global software engineering: the future of socio-technical coordination. In: *Future of Software Engineering (FOSE'07) IEEE; 2007:*
 43. 188-198.
44. Martini A, Pareto L, Bosch J. A multiple case study on the inter-group interaction speed in large, embedded software companies employing agile. *Softw Evol Process.* 2016;28(1):4-26.
46. Van De Ven AH, Delbecq AL, Koenig R Jr. Determinants of coordination modes within organizations. *Am sociolo rev.* 1976:322-338.
47. collab=Lunenburg FC. Organizational structure: Mintzberg's framework. *Int j scholar acad intell diver.* 2012;14(1):1-8.

48. Dingsøy T, Moe NB, Fægri TE, Seim EA. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empir Softw Eng.* 2018;23(1):490-520.
49. Espinosa JA, Slaughter SA, Kraut RE, Herbsleb JD. Familiarity, complexity, and team performance in geographically distributed software development. *Org sci.* 2007;18(4):613-630.
50. Moe NB, Smite D, Sblis A, Börjesson AL, Andréasson P. Networking in a large-scale distributed agile project, Vol. 12 ACM; 2014.
51. Runeson P, Host M, Rainer A, Regnell B. *Case Study Research in Software Engineering: Guidelines and examples*: John Wiley & Sons; 2012.
52. Stewart DW, Shamdasani PN. *Focus groups: Theory and Practice*. 20 Sage publications; 2014.
53. Sjoberg DI, Dyba T, Jorgensen M. The future of empirical methods in software engineering research *IEEE*; 2007:358-378.
54. Marsden PV. *Survey Methods for Network Data*, The SAGE handbook of social network analysis, vol. 25; 2011;370-388.
55. Manteli C, Van Den Hooff B, Van Vliet H. The effect of governance on global software development: An empirical research in transactive memory systems. *Inf Softw Technol.* 2014;56(10):1309-1321.
56. Heath S, Fuller A, Johnston B. Chasing shadows: defining network boundaries in qualitative social network analysis. *Qual Res.* 2009;9(5):645-661.

ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



9–10 грудня 2020 року

**ТЕРНОПІЛЬ
2020**

А. Вовк, П. Оберванюк ПРОБЛЕМИ ПРОЕКТУВАННЯ ПРОГРАМНОЇ АРХІТЕКТУРИ ДЛЯ АЖІЛЕ-ПРОЄКТІВ	
A. Vovk, N. Obervaniuk PROBLEMS OF SOFTWARE ARCHITECTURE DESIGN FOR AJILE-PROJECTS	156
О. Оліферчук, Г. Цуприк РОЗРОБКА СИСТЕМИ ВЗАЄМОДІЇ МОДУЛІВ: «КАФЕДРА» ТА «ДОКУМЕНТООБІГ» З ВИКОРИСТАННЯМ «.NET FRAMEWORK»	
O. Oliferchuk, H. Tsupryk IMPLEMENTATION OF THE .NET FRAMEWORK AND INTERACTION BETWEEN MODULES «ACADEMIC DEPARTMENT» «DOCUMENT MANAGEMENT»	157
Д. Ониськів, І. Коноваленко РОЗРОБКА ТА ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ КЛІМАТИЧНИМИ ПАРАМЕТРАМИ ТЕПЛИЦІ	
D. Onyskiv, I. Konovalenko DEVELOPMENT AND RESEARCH AUTOMATED CONTROL SYSTEMS OF CLIMATIC PARAMETERS GREENHOUSES	158
А. Ребуха ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМ ВИЯВЛЕННЯ ТА ЗАПОБІГАННЯ ВТОРГНЕНЬ У РОБОТУ ІНФОРМАЦІЙНИХ СИСТЕМ	
A. Rebukha COMPARATIVE RESEARCH OF INTRUSION DETECTION SYSTEMS AND INTRUSION PREVENTION SYSTEMS OPERATION IN INFORMATION SYSTEMS	159
Б. Бережний ЯК ПЕРЕКЛАСТИ ІНТЕРНЕТ-МАГАЗИН НА ПЛАТФОРМІ MAGENTO 2 НА УКРАЇНСЬКУ МОВУ?	
B. Bereznyi HOW TO TRANSLATE AN ONLINE STORE ON THE MAGENTO 2 PLATFORM INTO UKRAINIAN?	160
В. Залізняк, Г. Цуприк РОЗРОБКА СУЧАСНОГО ON-LINE СЕРВІСУ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ВЕБ-ПРОГРАМУВАННЯ	
V. Zalisnyak, H. Tsupryk DEVELOPMENT OF MODERN ON-LINE SERVICE WITH USE OF THE WEB-TECHNOLOGIES	161
С. Дячук, В. Малярський, Я. Кінах ПРОЄКТУВАННЯ ПРОГРАМНИХ WEB-СИСТЕМ НА ОСНОВІ ВИКОРИСТАННЯ ЗАСОБІВ КЕРУВАННЯ КОНТЕНТОМ	
S. Dyachuk, V. Malyarsky, I. Kinakh DESIGN OF SOFTWARE WEB-SYSTEMS USING CONTENT MANAGEMENT TOOLS	162
Н. Куцик, М. Петрик РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ АНАЛІЗУ СТАТИСТИКИ	
N. Kutsyk, M. Petryk DEVELOPMENT OF AN AUTOMATED SYSTEM OF ANALYSIS OF STATISTICS	163
А. Дубчак, Я. Литвиненко ПАТТЕРНИ ТА ПРИНЦИПИ. ПАНАЦЕЯ ЧИ ПРОБЛЕМА	
A. Dubchak, I. Lytvynenko PATTERNS AND PRINCIPLES. PANACEA OR PROBLEM	164

УДК 004.415

А.І. Вовк, Н.-П. Б.Оберванюк

(Тернопільський національний технічний університет імені Івана Пулюя)

ПРОБЛЕМИ ПРОЕКТУВАННЯ ПРОГРАМНОЇ АРХІТЕКТУРИ ДЛЯ AJILE-ПРОЄКТІВ

UDC 004.415

A.I. Vovk, N.-P.B. Obervaniuk

PROBLEMS OF SOFTWARE ARCHITECTURE DESIGN FOR AJILE- PROJECTS

Сучасні технології програмування (гнучке та екстремальне програмування, технологія SCRUM та інші) по суті є ітераційними. При виконанні поточної ітерації можуть вноситися зміни у вимоги або обмеження, що потребуватиме внесення відповідних змін у розділи проекту, в тому числі і в розділ архітектури. Вибір варіанта архітектури здійснюється з множини альтернатив, які конструюються на основі функціональних вимог із стандартних компонентів (патернів). Для підвищення обґрунтованості прийнятих рішень та автоматизації процесу використовуються методи оптимізації та багатокритеріального вибору[1]. Для оцінювання альтернатив по окремих критеріях якості найбільш ефективним є метод аналізу ієрархій (MAI) або його модифікований варіант. В цих методах відносна оцінка альтернатив визначається з використанням експертної інформації, і при включенні в розгляд нових альтернатив потрібно повторно проводити експертне оцінювання та розрахунки ваг альтернатив. Для оцінювання та вибору архітектури по множині критеріїв, як правило, використовують лінійну згортку [2].

Для вирішення цієї задачі можна також застосувати нелінійну скалярну згортку, в якій реалізований принцип «далі від обмежень». Однак, тут теж виникає проблема збіжності до оптимуму процедури симплекс-планування при визначенні ваг критеріїв якості. Для уникнення перерахованих проблем бажано побудувати цільову функцію для вибору архітектури в аналітичному вигляді, структура і параметри якої визначалися б об'єктивно, на основі експерименту, а не постулювались.

Пропонується для побудови цільової функції вибору архітектури використати МГУА в поєднанні з MAI. В цьому методі, для вибору моделі цільової функції генеруються різні структури моделей в обраному класі. Селекція моделей і прийняття рішення про завершення процесу відбувається за значенням критерію, обчисленого на послідовності експертних значень цільової функції, які визначаються методом аналізу ієрархій. Оскільки експертам необхідно визначати оцінки альтернатив по сукупності критеріїв якості, то критерій неузгодженості матриці парних порівнянь може перевищувати допустиме значення і отримані оцінки виявляться некоректними. Тому пропонується використовувати модифікований метод аналізу ієрархій, в якому ваги альтернатив визначаються з умови мінімізації неузгодженості матриці парних порівнянь.

Література.

1. Kharchenko, A.; Halay, I.; Bodnarchuk, I. Multicriteria architecture choice of software system under design and reengineering. 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT). Anais. In: CSIT. Lviv, Ukraine: IEEE, 4–8, set. 2016.
2. Bodnarchuk, Ihor, et al. Adaptive Method for Assessment and Selection of Software Architecture in Flexible Techniques of Design. In: 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). IEEE, 2018. p. 292–297.