ISSN 1998-7927(print) ISSN 2664-6498 (online)

DOI: https://doi.org/10.33216/1998-7927-2026-304-6-102-114

УДК 004.89:004.41

ГРУПОВА ДИНАМІКА ТА КОМУНІКАЦІЇ В УМОВАХ ВЗАЄМОДІЇ ІТ-КОМАНДИ Й ШТУЧНОГО ІНТЕЛЕКТУ

Юскович-Жуковська В.І.

GROUP DYNAMICS AND COMMUNICATION IN THE INTERACTION OF IT TEAM AND ARTIFICIAL INTELLIGENCE

Yuskovych-Zhukovska V.I.

У статті досліджено трансформацію групової динаміки та комунікацій у командах розробників програмного забезпечення в умовах інтеграції технологій штучного інтелекту у процеси Software Engineering. Актуальність дослідження зумовлена переходом від використання AI як допоміжного інструменту окремого фахівця до його включення у командні процеси, життєвий цикл розроблення програмного забезпечення та механізми колективного прийняття технічних рішень. Методологічною основою є соціотехнічний підхід, відповідно до якого IT-команда розглядається як динамічна гібридна система взаємодії людей, AI-компонентів, програмних засобів та інфраструктури. Встановлено, що інтеграція AI одночасно змінює технічне і соціотехнічне середовища команди, трансформує інформаційні потоки, розподіл ролей і відповідальності, механізми координації, лідерства, довіри, згуртованості, психологічної безпеки та колективного прийняття рішень. Показано, що традиційна комунікація «людина–людина» доповнюється каналами «людина–AI» та «AI–AI», унаслідок чого формується гетерогенна комунікаційна мережа. Обґрунтовано необхідність каліброваної довіри до AI-рекомендацій, людської верифікації результатів і збереження відповідальності за остаточні рішення за учасниками команди. На основі узагальнення результатів запропоновано Human–AI Group Dynamics Transformation Model (HAGDTM), яка описує взаємозв'язок між AI-технологіями, технічним середовищем, ролями, інформаційними потоками, груповою динамікою та командними результатами. Для подальшої емпіричної операціоналізації моделі запропоновано інтегральний Group Transformation Index (GTI), що враховує ефективність комунікації, розподіл ролей, рівень довіри, якість колективного знання, прийняття рішень, продуктивність і рівень залежності від AI. Практичне значення результатів полягає у можливості використання моделі для аналізу та проєктування командних процесів, визначення меж делегування завдань AI, організації людського контролю та оцінювання ефективності гібридної взаємодії в програмно-інженерних командах.

Ключові слова: *розробник ПЗ, IT-команда, людино-ШІ взаємодія, групові комунікації*

Вступ .Сучасний розвиток технологій штучного інтелекту зумовлює суттєву трансформацію підходів до розроблення програмного забезпечення (ПЗ), змінюючи традиційні моделі виконання програмно-інженерних завдань, розподіл функцій між учасниками процесу та способи взаємодії з програмними засобами. Генеративні моделі, великі мовні моделі, AI coding assistants та автономні AI-агенти поступово інтегруються у різні етапи Software Development Lifecycle — від аналізу вимог і проєктування до генерації коду, тестування, налагодження, документування та супроводу програмних систем. У результаті штучний інтелект перестає розглядатися виключно як інструмент автоматизації окремих операцій і стає компонентом програмно-інженерного процесу, здатним впливати на структуру та організацію діяльності команд розробників.

Традиційно процес розроблення ПЗ розглядається переважно з позицій технологій, методологій, архітектури, програмних засобів та організації життєвого циклу програмного продукту. Водночас результативність Software Engineering значною мірою визначається характером взаємодії членів IT-команди, розподілом ролей і відповідальності, інформаційними потоками, координацією діяльності та механізмами прийняття колективних рішень. Це дозволяє розглядати команду розробників ПЗ не лише як сукупність IT-фахівців, які використовують певний набір технологій, а як соціотехнічну систему, у якій технічні та соціальні компоненти функціонують у взаємозв'язку.

Як видно з рис.1, команда розробників ПЗ функціонує одночасно у двох взаємопов'язаних середовищах:

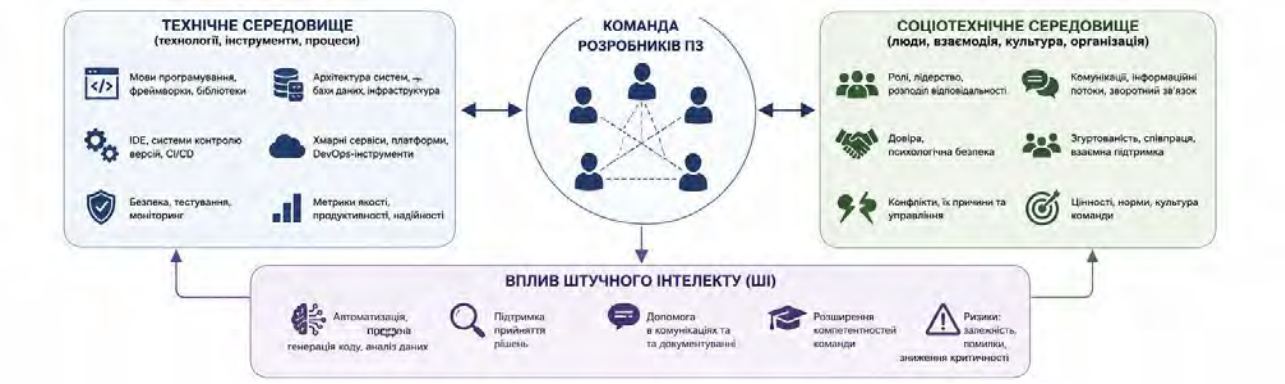


Рис. 1. Два взаємопов'язані середовища функціонування команди розробників програмного забезпечення

1. Технічне середовище
- технології та інструменти;
- IDE, Git, CI/CD, DevOps;
- архітектура ПЗ;
- бази даних та інфраструктура;
- тестування, моніторинг;
- технологічні процеси та обмеження.

2. Соціотехнічне середовище
- групова динаміка;
- ролі та розподіл відповідальності;
- лідерство;
- комунікації;
- довіра;
- психологічна безпека;
- конфлікти;
- згуртованість;
- командні норми та культура;
- прийняття колективних рішень.

Інтеграція технологій штучного інтелекту змінює одночасно обидва середовища функціонування команди. У технічному середовищі AI забезпечує генерацію та аналіз програмного коду, автоматизацію тестування, пошук дефектів, створення документації, підтримку рефакторингу та інші функції Software Engineering. У соціотехнічному середовищі його вплив проявляється через зміну інформаційних потоків, способів координації, розподілу завдань, групових ролей, механізмів обміну знаннями та прийняття колективних рішень. Таким чином, впровадження AI створює новий тип командної взаємодії, у якому поряд із традиційною комунікацією «людина–людина» виникають і розвиваються взаємодії «людина–AI» та, у разі використання AI-агентів, «AI–AI».

Аналіз останніх досліджень і публікацій. Одним із актуальних напрямів сучасних досліджень є аналіз впливу штучного інтелекту та великих мовних моделей на життєвий цикл розроблення програмного забезпечення. AI розглядається як фактор трансформації Software Development Lifecycle та відзначається його застосування у генерації коду, налагодженні, тестуванні й супроводі програмного забезпечення [1]. Тобто, AI вже впливає не на окрему операцію програміста, а на різні етапи SDLC. Важливим емпіричним напрямом є дослідження практичного використання AI coding assistants. У роботі [2] досліджено використання AI-помічників у таких видах діяльності, як реалізація нових функціональних можливостей, тестування, triaging помилок, рефакторинг та створення текстових артефактів. Дослідження спрямоване не лише на визначення переваг AI, а й на встановлення того, на яких етапах процесу розроблення програмісти використовують або не використовують AI поступово інтегрується у реальні процеси розроблення ПЗ. Отже, змінюється не лише інструментарій окремого розробника, а й організація спільної роботи команди.

У роботі [3] досліджено використання AI-помічників у Daily Scrum та feature refinement. Автори аналізують не лише технологічні можливості AI, а й його зв'язок із team collaboration dynamics, тобто динамікою командної співпраці. Дослідження сформувало практичні рекомендації щодо інтеграції AI у Agile-процеси. Отже, ця робота демонструє перехід від питання: «Що AI може зробити для програміста?» до питання: «Як AI змінює спосіб спільної роботи команди?». Але дослідження розглядає переважно окремі Agile-зустрічі, а не комплексну трансформацію групової динаміки та комунікацій команди.

Автори [4] розглядають сучасну Software Engineering як людиноцентричну соціотехнічну екосистему та ставлять питання про те, як AI-powered agents змінюватимуть human-to-human, human-to-agent та agent-to-agent communication. Окремо визначено питання майбутнього Software Engineering teams та впливу AI на командну співпрацю, комунікацію і довіру. Таким чином, можна стверджувати, що майбутня команда розробників ПЗ — це вже не просто група людей із набором програмних інструментів, а соціотехнічна система, у якій взаємодіють люди, AI-агенти, програмні засоби та процеси розроблення.

Наша стаття може зробити наступний крок — концептуально систематизувати саме групову динаміку та комунікації в цій системі.

Науковці [1] підкреслюють, що AI поступово проникає у різні фази SDLC — від генерації та

аналізу коду до тестування, налагодження та супроводу. При цьому автори визначають цілий комплекс відкритих питань, пов'язаних із адаптацією традиційних Software Engineering методологій до AI-supported development. Це створює важливу передумову для нашого дослідження: якщо AI змінює процеси SDLC, він неминуче впливає і на організацію спільної роботи людей, які беруть участь у цих процесах.

Ще один важливий напрям пов'язаний із переходом від одного AI-помічника до LLM-based multi-agent systems for Software Engineering розглянуто у [5], де автори показують потенціал спеціалізованих AI-агентів для розв'язання складних програмно-інженерних завдань. Тобто, якщо сьогодні ми маємо: *розробник ↔ AI-помічник,* то розвиток *multi-agent systems* може привести до: *розробник ↔ AI-agent₁ ↔ AI-agent₂ ↔ AI-agent₃ ↔ розробник.*

Тобто структура команди та комунікаційна мережа потенційно стають гетерогенними та посилюється роль групової динаміки.

Дослідження [6] показує трансформацію ролі програміста в умовах інтеграції LLM. Автори характеризують сучасного розробника дедалі більше як стратегічного співробітника AI-систем, а не виключно виконавця програмного коду. Серед нових видів діяльності виділяються prompt engineering, постійне підвищення кваліфікації, взаємодія з AI та оцінювання його результатів.

У дослідженні [7] проаналізовано використання GenAI студентами під час командної розробки програмного забезпечення. Важливо, що AI використовувався не тільки для генерації коду, а й для ретроспективної перевірки відповідності реалізації вимогам і дизайну. Майбутні програмні інженери навчаються працювати вже не лише в середовищі «людина–людина», а в середовищі «людина–людина–AI». Адже якщо змінюється роль окремого розробника:

programmer → AI-assisted developer → AI supervisor / evaluator / coordinator

то змінюється і рольова структура команди в цілому. Тобто, виникає проблема трансформації групових ролей.

У працях [8, 9] показано, що технології штучного інтелекту вже активно інтегруються в інструменти управління IT-проєктами та зумовлює поступову трансформацію управлінських практик, групової динаміки та комунікацій у командах розробників ПЗ. Потреба адаптації розробників ПЗ усіх рівнів до використання AI-агентів становиться невід'ємною складовою професійної діяльності IT-фахівців.

Таким чином, аналіз сучасних досліджень показує, що вплив штучного інтелекту на Software Engineering уже активно досліджується на рівні SDLC, окремих програмно-інженерних завдань, AI-assisted development, Agile-взаємодії, multi-agent systems та трансформації професійних ролей. Водночас наявні дослідження переважно розглядають окремі компоненти цієї трансформації. Недостатньо системних досліджень, у яких інтеграція AI розглядається як фактор комплексної трансформації групової динаміки команди розробників програмного забезпечення, що одночасно охоплює групові ролі, комунікації, координацію, лідерство, довіру, згуртованість, конфліктну взаємодію, психологічну безпеку та колективне прийняття рішень.

Невирішеною залишається також проблема концептуального опису взаємодії людських та AI-компонентів як елементів єдиної соціотехнічної системи. Існуючі підходи здебільшого концентруються або на технологічній ефективності AI, або на окремих аспектах Human–AI Collaboration, не формуючи цілісної моделі трансформації групової динаміки команди розробників ПЗ. Саме ця проблема визначає напрям подальшого дослідження.

Невирішені частини загальної проблеми. Попри активний розвиток досліджень у галузі AI-assisted Software Engineering та Human–AI Collaboration, питання трансформації групової динаміки команд розробників програмного забезпечення під впливом штучного інтелекту залишаються недостатньо систематизованими. Більшість сучасних досліджень зосереджується на технологічних можливостях AI, продуктивності окремих розробників, якості генерованого програмного коду, автоматизації програмно-інженерних завдань або використанні AI у конкретних етапах SDLC. Водночас вплив AI на команду як цілісну соціотехнічну систему потребує подальшого дослідження.

Недостатньо визначеним залишається питання трансформації групових ролей у командах розробників. Поява AI-помічників та AI-агентів змінює не лише набір інструментів, доступних програмісту, а й характер його професійної діяльності. Розробник дедалі частіше виконує функції постановника завдань для AI, оцінювача результатів, контролера їхньої якості та координатора взаємодії між різними AI-компонентами. Це може зумовлювати перерозподіл функцій і відповідальності між членами команди, однак механізми такої трансформації ще не отримали достатнього концептуального обґрунтування.

Окремою невирішеною проблемою є трансформація комунікаційних процесів. Традиційна модель командної взаємодії переважно базується на комунікації «людина–людина». Інтеграція AI створює додаткові канали «людина–AI», а використання автономних і мультиагентних систем — потенційно також «AI–AI». Унаслідок цього змінюються структура інформаційних потоків, швидкість поширення інформації, способи формування та перевірки знань і механізми зворотного зв'язку. Потребує дослідження питання, як ці зміни впливають на ефективність командної комунікації.

Недостатньо дослідженим залишається також вплив AI на координацію та колективне прийняття рішень. Якщо частина аналітичних, прогнозних або рекомендаційних функцій передається AI-системам, виникає необхідність визначення меж відповідальності між людиною та штучним інтелектом. Це особливо важливо для командних рішень, оскільки результат взаємодії залежить не лише від якості AI-рекомендації, а й від способу її інтерпретації та прийняття командою.

Отже, невирішеною залишається проблема комплексного концептуального представлення трансформації групової динаміки команди розробників програмного забезпечення в умовах інтеграції штучного інтелекту. Для її розв'язання необхідно перейти від аналізу окремих проявів використання AI до системного розгляду взаємопов'язаних компонентів командної взаємодії, у якій люди та AI-компоненти функціонують у межах єдиного соціотехнічного середовища.

Метою дослідження є концептуалізація трансформації групової динаміки та комунікацій у командах розробників програмного забезпечення в умовах інтеграції технологій штучного інтелекту та визначення основних компонентів соціотехнічної взаємодії «людина–людина–AI».

Для досягнення поставленої мети визначено такі **завдання:**

1. проаналізувати сучасні підходи до використання AI у процесах Software Engineering та Human–AI Collaboration;
2. визначити основні напрями впливу AI на групову динаміку команд розробників ПЗ;
3. дослідити трансформацію групових ролей і розподілу відповідальності в умовах взаємодії людей та AI-компонентів;
4. визначити зміни у структурі командних комунікацій, інформаційних потоків та механізмах координації;
5. проаналізувати потенційний вплив AI на лідерство, довіру, згуртованість, конфліктну взаємодію та психологічну безпеку;
6. дослідити трансформацію механізмів колективного прийняття рішень;
7. сформувати концептуальну модель групової динаміки команди розробників ПЗ як соціотехнічної системи, що включає людські та AI-компоненти.

Об'єктом дослідження є процеси групової взаємодії та комунікацій у командах розробників програмного забезпечення.

Предметом дослідження є закономірності та механізми трансформації групової динаміки, комунікацій, ролей, координації та колективного прийняття рішень у командах розробників ПЗ під впливом інтеграції технологій штучного інтелекту.

Методологічною основою дослідження є соціотехнічний підхід, відповідно до якого команда розробників програмного забезпечення розглядається як динамічна система, що формується взаємодією людських, технологічних та організаційних компонентів. У межах такого підходу інтеграція штучного інтелекту розглядається не лише як технологічне вдосконалення програмних засобів, а як фактор зміни структури та механізмів командної взаємодії.

Виклад основного матеріалу дослідження. Інтеграція технологій штучного інтелекту у процеси розроблення програмного забезпечення змінює характер діяльності команди не лише через автоматизацію окремих програмно-інженерних операцій, а й через трансформацію способів розподілу завдань, відповідальності та інформації між учасниками процесу. Якщо на ранніх етапах застосування AI переважно розглядалося як використання окремого програмного інструменту розробником, то сучасні генеративні моделі, AI coding assistants та AI-агенти поступово інтегруються безпосередньо у робочі процеси команди.

У традиційній моделі програмної розробки основні комунікаційні та функціональні зв'язки формуються між людьми. Розробник отримує завдання від іншого учасника команди, аналізує вимоги, створює програмний код, передає результати на перевірку та отримує зворотний зв'язок. Програмні засоби при цьому виконують переважно інструментальну функцію.

Умовно таку структуру можна представити як:

$Human_1 \leftrightarrow Human_2 \leftrightarrow Human_3$

за підтримки: *Software Tools.*

Інтеграція генеративного AI створює додатковий рівень взаємодії:

$Human \leftrightarrow AI \leftrightarrow Software\ Environment.$

У командному середовищі це призводить до формування складнішої структури:

$Human_1 \leftrightarrow Human_2 \leftrightarrow AI \leftrightarrow Human_3 \leftrightarrow Software\ Environment.$

За використання декількох AI-агентів структура стає ще складнішою:

$Human_i \leftrightarrow AI_j \leftrightarrow AI_k \leftrightarrow Human_m.$

Отже, AI перестає бути лише зовнішнім інструментом і стає активним компонентом соціотехнічного середовища команди.

А далі в самій статті я пропоную вже формалізувати цю ідею.

Наприклад: $G = (V, E)$, де:

- $V=H \cup A$ — множина учасників системи, яка включає людей H та AI-компоненти A;
- E — множина комунікаційних і функціональних взаємодій.

Тоді команда розробників перестає бути просто:

$Developer_1 \leftrightarrow Developer_2 \leftrightarrow Developer_3$

і перетворюється на:

$Developer_1 \leftrightarrow AI \leftrightarrow Developer_2 \leftrightarrow Developer_3 \leftrightarrow AI\text{-}agent$

Як було показано у вступі, команда розробників функціонує у двох взаємопов'язаних середовищах — технічному та соціотехнічному. Інтеграція AI впливає на обидва ці середовища одночасно.

У **технічному середовищі** відбуваються зміни, пов'язані з:

- автоматизованою генерацією програмного коду;
- інтелектуальним аналізом коду;
- автоматизацією тестування;
- пошуком і класифікацією дефектів;
- автоматизованим рефакторингом;
- генерацією документації;
- аналізом вимог;
- підтримкою проєктування програмних систем;
- використанням AI-агентів для виконання комплексних програмно-інженерних завдань.

Проте технологічна трансформація породжує зміни і в **соціотехнічному середовищі**. Зокрема, змінюються:

Roles → Communication → Coordination

а через них:

Trust, Leadership, Cohesion, Conflict, Decision Making.

Таким чином, технологічні зміни не є ізольованими від групової динаміки. Вони створюють ланцюг взаємопов'язаних трансформацій, у якому зміна способу виконання програмно-інженерних завдань може призводити до зміни способу взаємодії членів команди.

На основі аналізу сучасних досліджень можна запропонувати такий концептуальний ланцюг:

AI Technologies→Technical Environment→Roles and Information Flows→Group Dynamics→Team Outcomes.

Тобто, AI-технології змінюють технічне середовище. Це, в свою чергу, зумовлює зміни ролей та інформаційних потоків, що впливає на групову динаміку та може відображатися на командних результатах.

Групова динаміка також може впливати на характер використання AI. Наприклад, рівень довіри між членами команди може визначати готовність команди використовувати AI-рекомендації, а командні норми можуть визначати допустимі межі автоматизації та контролю результатів AI.

Тому ми розглядаємо систему як динамічний контур зворотного зв'язку, як взаємний вплив технологічного та соціального компонентів:

AI→Team Transformation→Team Behavior→AI Adoption→AI.

І. Трансформація групових ролей у командах розробників програмного забезпечення.

Традиційна команда розробників програмного забезпечення характеризується функціональним розподілом ролей, у межах якого окремі учасники відповідають за аналіз вимог, проєктування, програмування, тестування, управління конфігураціями, DevOps-процеси, управління проєктом та інші види діяльності. Такий розподіл формується відповідно до професійних компетентностей учасників команди та особливостей життєвого циклу програмного продукту. Програмні засоби при цьому переважно виконують інструментальну функцію, тоді як відповідальність за постановку завдання, вибір рішення, оцінювання його результату та прийняття остаточного рішення залишається за людиною.

Інтеграція генеративного штучного інтелекту змінює цю модель. AI coding assistants, великі мовні моделі та AI-агенти можуть виконувати частину операцій, які раніше безпосередньо здійснювалися розробниками. До них належать генерація програмного коду, створення тестів, аналіз помилок, рефакторинг, формування документації, пошук альтернативних реалізацій та підтримка аналізу технічних рішень. Унаслідок цього трансформується не лише набір інструментів програміста, а й функціональний зміст його ролі в команді.

У традиційній моделі роль розробника умовно може бути представлена як:

Developer = Task Executor + Code Producer.

В умовах інтеграції AI вона набуває більш складної структури:

Developer = Task Designer + AI User + Result Evaluator + Decision Maker + Coordinator.

Таким чином, програміст поступово переміщується від переважно виконавчої ролі до ролі організатора та контролера взаємодії з AI-компонентами. Це не означає зникнення традиційних програмно-інженерних компетентностей. Навпаки, здатність оцінювати коректність, безпеку, продуктивність і відповідність отриманого результату вимогам стає ще більш важливою.

Одним із найбільш помітних наслідків інтеграції генеративного AI є зміна співвідношення між створенням та оцінюванням програмного результату. Якщо традиційний процес передбачав безпосереднє написання більшої частини програмного коду розробником, то використання LLM дозволяє передавати AI частину операцій генерації.

У цьому випадку програміст виконує послідовність:

Task → Prompt → AI Output → Evaluation → Modification → Integration.

Отже, програмний результат формується вже не виключно як результат безпосереднього написання коду людиною. Виникає ітеративний цикл взаємодії людини та AI, у якому людина формує завдання і контекст, AI генерує або пропонує рішення, а розробник перевіряє, модифікує та інтегрує отриманий результат.

Це зумовлює появу нових функціональних ролей одного й того самого фахівця:

- ініціатор — формує завдання для AI;
- контекстний координатор — забезпечує AI необхідною інформацією;
- оцінювач — перевіряє отриманий результат;
- редактор — модифікує результат відповідно до вимог;
- контролер якості — оцінює функціональні та нефункціональні характеристики;

- інтегратор — включає результат у загальну програмну систему;
- відповідальний за рішення — приймає остаточне рішення щодо використання AI-результату.

Таким чином, одна з ключових трансформацій полягає у переході від моделі:

«розробник ПЗ створює результат»

до моделі:

«розробник ПЗ організовує, оцінює та контролює процес отримання результату за участю AI».

Зміна ролей неминуче породжує питання відповідальності. AI може генерувати програмний код, пропонувати архітектурні рішення або створювати тестові сценарії, однак юридична, професійна та організаційна відповідальність за використаний у програмному продукті результат не може автоматично передаватися AI.

Тому виникає принципова відмінність між: *AI contribution* та *Human responsibility*.

AI може брати участь у створенні результату, але остаточне рішення щодо його прийняття залишається за відповідальним учасником команди.

У командному середовищі це створює нову проблему розподілу відповідальності:

Responsibility = Human Decision + AI Contribution + Human Verification.

Чим більшу кількість функцій передано AI, тим важливішими стають механізми людського контролю, верифікації та аудиту результатів.

Особливої актуальності це набуває у випадку використання автономних AI-агентів. Якщо агент може самостійно аналізувати завдання, генерувати код, запускати тести, виявляти помилки та повторювати цикл удосконалення, межа між «інструментом» і «активним учасником процесу» стає менш очевидною. Відповідно, команді необхідно визначати, які рішення можуть делегуватися AI, а які повинні залишатися виключною компетенцією людини.

Застосування AI не обмежується зміною ролі одного програміста, змінюється рольова структура IT-команди в цілому.

У традиційній команді основними ролями можуть бути:

Developer, Tester, Architect, DevOps, Project Man ager.

У AI-augmented команді до них додаються функції, пов'язані з організацією та контролем AI-взаємодії:

AI Coordinator, AI Evaluator, AI Supervisor, AI Or chestrator.

У реальній IT- команді одна людина може поєднувати декілька таких посад.

Наприклад, senior developer може одночасно виконувати функції: *Developer + A I Supervisor + Evaluator.*

Tech lead може поєднувати: *Technical Leader + AI Coordinator + Decision Maker.*

А спеціалізований AI-агент може виконувати окрему функціональну роль:

AI Agent = Code Generator + Tester + Analyzer.

Отже, традиційна статична структура ролей поступово може перетворюватися на динамічну функціональну структуру, у якій конкретний учасник або AI-компонент отримує певну роль залежно від завдання.

Особливо важливою є поява гібридних ролей, у яких функції людини та AI взаємодоповнюють одна одну.

Наприклад:

$Human_{Developer} + AI_{Code\ Generator} \rightarrow Software\ Implementation$

або

$Human_{Tester} + AI_{TestGenerator} \rightarrow Test\ Suite.$

У такій моделі результат є продуктом не окремого учасника, а гібридної взаємодії.

Це дозволяє запропонувати поняття гібридної командної ролі — функціональної ролі, виконання якої забезпечується спільною діяльністю людини та AI-компонента.

Відповідно, одиницею аналізу групової динаміки стає не тільки окремий член команди, а й зв'язка «людина–AI», яка виконує певну функцію:

$Traditional\ Team = \{Human_1, Human_2, \ldots, Human_n,$

тоді як:

$AI\text{-}augmented\ Team = \{Human_1, \ldots, Human_n, AI_1, \ldots, AI_m\}.$

Оскільки AI-компоненти відрізняються від людей за природою, рівнем автономності, механізмами прийняття рішень та відповідальністю, тому така команда є гетерогенною соціотехнічною системою.

Зміна рольової структури впливає і на лідерство. Традиційний технічний лідер значною мірою забезпечує розподіл завдань, контроль виконання, технічну координацію та прийняття ключових рішень.

В умовах широкого використання AI частина аналітичних та виконавчих функцій може бути автоматизована. У результаті роль лідера зміщується від безпосереднього контролю виконання до організації взаємодії між людьми, AI та програмними процесами.

Умовно цю трансформацію представимо:

Technical Control→Coordination→AI Governanc e→Collective Decision Making.

Технічний лідер повинен не лише оцінювати роботу членів команди, але й визначати:

- де доцільно використовувати AI;
- які функції можуть бути делеговані AI;
- які результати потребують обов'язкової людської перевірки;
- як контролювати якість AI-generated artifacts;
- хто несе відповідальність за остаточне рішення;
- як інтегрувати результати AI у командні процеси.

Таким чином, AI не обов'язково зменшує роль лідера; він змінює її зміст.

Зміна ролей є важливою не сама по собі, а через її вплив на інші характеристики команди. Перерозподіл функцій між людиною та AI може змінювати:

Roles → Communication → Coordination → Trust → Cohesion → Decision Making.

Наприклад, якщо AI бере на себе значну частину генерації програмного коду, змінюється характер комунікації між розробником і code reviewer. Якщо AI використовується для аналізу вимог, змінюється взаємодія між бізнес-аналітиком, розробником і тестувальником. Якщо AI-агенти самостійно виконують частину завдань, виникає потреба координувати не лише людей, а й AI-компоненти.

Таким чином, трансформація ролей є первинним механізмом, через який технологічна інтеграція AI поширюється на інші компоненти групової динаміки.

Це дозволяє сформулювати перше концептуальне положення нашого дослідження:

Положення 1. Інтеграція AI у Software Engineering трансформує рольову структуру команди шляхом переходу від переважно людиноцентричного розподілу функцій до гібридного розподілу, у якому функціональні ролі можуть виконуватися людиною, AI-компонентом або їхньою комбінацією.

Трансформація комунікацій у командах розробників програмного забезпечення. Комунікація є одним із базових механізмів функціонування команди розробників програмного забезпечення, оскільки забезпечує передавання вимог, технічних рішень, результатів роботи, зворотного зв'язку та інформації про стан виконання завдань. У традиційній команді основна частина комунікаційних зв'язків формується між людьми: розробниками, тестувальниками, DevOps-фахівцями, проєктними та технічними керівниками, бізнес-аналітиками та іншими учасниками процесу.

Інтеграція технологій штучного інтелекту створює додатковий комунікаційний рівень. AI перестає бути лише пасивним програмним інструментом і стає джерелом інформації, рекомендацій, програмних артефактів та альтернативних рішень. Унаслідок цього традиційна структура комунікацій:

Human ↔ Human

доповнюється взаємодіями: *Human ↔ AI ,* та, у разі використання мультиагентних систем:

AI ↔ AI.

У результаті формується гетерогенна комунікаційна мережа:

Communication Network = {H − H, H − AI, AI − AI},

де H−H — комунікації між людьми, H−AI — взаємодія людини з AI-компонентами, а AI−AI — інформаційний обмін між AI-агентами.

У традиційній команді значна частина інформації передається безпосередньо між учасниками. Наприклад, бізнес-аналітик формує вимоги, розробник уточнює технічні деталі, тестувальник повідомляє про виявлені дефекти, а технічний лідер координує прийняття рішення.

AI додає до цієї структури новий тип комунікаційного вузла.

У найпростішому випадку:

Developer → AI → Developer.

Розробник формує запит, AI генерує результат, а людина оцінює його.

Проте AI-результат може бути переданий іншим членам команди:

Developer → AI → Artifac t→ Reviewer.

Таким чином, AI стає проміжною ланкою в інформаційному потоці.

У складніших системах можливе формування мережі:

$Human_1 \leftrightarrow AI_1 \leftrightarrow AI_2 \leftrightarrow Human_2.$

У цьому випадку інформація може проходити через декілька AI-компонентів до того, як потрапити до IT-фахівця (людини) , який приймає остаточне рішення.

Це змінює не тільки кількість комунікаційних зв'язків, а й їхню структуру.

Одним із найбільш істотних наслідків інтеграції AI є зміна способу формування, поширення та перевірки інформації.

У традиційній команді інформаційний потік представимо:

Source → Human → Human → Decision.

У AI-augmented команді він може набувати форми:

Source → Human → AI → Human → Decision,

або:

$Source \rightarrow AI \rightarrow Human_1 \rightarrow Human_2 \rightarrow Decision.$

Отже, інформація може проходити через додаткові етапи генерації, трансформації та перевірки.

Це створює принципово нову проблему — походження інформації.

Член команди повинен розуміти:

- хто сформував інформацію;
- чи є вона первинною або AI-generated;
- на яких даних ґрунтується рекомендація;
- чи була вона перевірена людиною;
- хто відповідає за її використання;
- наскільки результат відповідає контексту конкретного програмного проєкту.

Таким чином, у команді виникає потреба не тільки в інформаційному обміні, але й у верифікації та трасуванні інформаційних потоків.

Особливу роль AI може виконувати як комунікаційний посередник між учасниками команди.

Наприклад, AI може:

- узагальнювати результати обговорень;
- формувати протоколи зустрічей;
- структурувати вимоги;
- аналізувати issue та pull requests;
- пояснювати технічні рішення;
- генерувати документацію;
- формувати рекомендації;

- знаходити суперечності у вимогах;
- допомагати підготувати альтернативні варіанти рішення.

У такому випадку AI не просто передає інформацію. Він здійснює її структурування, узагальнення та перетворення.

Тому комунікаційний процес набуває форми:

Information → AI Processing → Structured Knowledge → Team.

Це може зменшувати інформаційне навантаження на членів IT-команди, але одночасно створює нові ризики.

Якщо AI неправильно інтерпретує інформацію, помилка може бути поширена на декількох учасників команди. Отже, AI може виступати не лише підсилювачем комунікації, але й потенційним множником інформаційних помилок.

Важливою характеристикою групової комунікації є feedback. У традиційній команді зворотний зв'язок переважно надходить від інших людей:

Human → Artifact → Human Feedback.

Наприклад, розробник створює програмний код, після чого code reviewer або тестувальник надає оцінку.

За використання AI з'являється додатковий рівень:

Human → Artifact → AI Feedback → Human Evaluation.

AI може швидко надати рекомендації щодо стилю коду, потенційних дефектів, тестового покриття або альтернативної реалізації.

Однак AI-feedback не є еквівалентом людського експертного оцінювання. Тому формується багаторівнева система зворотного зв'язку, у якій AI-рекомендація повинна співвідноситися з професійною оцінкою людини.

У цьому контексті роль розробника змінюється:

Feedback Receiver → Feedback Evaluator.

Тобто розробник повинен не лише отримати рекомендацію, а й визначити її релевантність, коректність та доцільність.

Зміна комунікаційних потоків безпосередньо впливає на координацію діяльності команди.

Традиційно координація базується на узгодженні дій між людьми:

$Human_1 \rightarrow Human_2 \rightarrow Human_3$.

AI може частково автоматизувати координаційні операції, наприклад аналізувати статуси завдань, визначати потенційні залежності, формувати нагадування або пропонувати порядок виконання робіт.

У результаті виникає модель:

Human Team ↔ AI Coordinator.

За використання multi-agent systems можлива ще складніша структура:

Human Team ↔ AI Coordinator ↔ AI Agents.

Це означає, що координація поступово може переходити від виключно людського процесу до гібридної координації, у якій AI підтримує розподіл інформації та завдань.

Однак остаточне рішення щодо пріоритетів, ресурсів та відповідальності повинно залишатися контрольованим людьми.

Поряд із позитивними ефектами інтеграція AI створює нові ризики для командної комунікації.

До основних можна віднести:

1. Інформаційне перевантаження. Зростання кількості AI-generated артефактів може збільшувати обсяг інформації, яку повинні опрацювати члени команди.

2. Втрата контексту. AI може генерувати формально коректний результат, який не враховує специфічний контекст проєкту.

3. Поширення помилкової інформації. Помилкова AI-рекомендація може бути повторно використана кількома учасниками команди.

4. Зниження прозорості походження рішення. У складній системі важко встановити, чи було рішення сформоване людиною, AI або їхньою комбінацією.

5. Надмірна залежність від AI. Якщо члени команди починають автоматично приймати AI-generated рекомендації, зменшується роль критичного професійного оцінювання.

6. Фрагментація комунікацій. Використання різними учасниками різних AI-систем може створювати паралельні інформаційні потоки, які не завжди синхронізуються.

Таким чином, ефективність AI-комунікації визначається не тільки технічною якістю AI-системи, але й правилами її включення у командний інформаційний контур.Зміна комунікацій є принципово важливою тому, що саме через комунікаційні процеси реалізуються більшість інших компонентів групової динаміки.

Можемо представити цей зв'язок так:

AI → Communication → Information Flow → Coordination → Group Dynamics.

Наприклад, зміна інформаційних потоків може впливати на розподіл ролей; зміна координації — на лідерство; доступність AI-рекомендацій — на довіру; а зміна механізмів прийняття рішень — на згуртованість і конфліктну взаємодію.

Тому комунікацію доцільно розглядати не як окремий елемент команди, а як інтегруючий механізм, через який технологічні зміни поширюються на соціальну структуру команди.

Це дозволяє сформулювати друге концептуальне положення:

Положення 2. Інтеграція AI трансформує комунікаційну структуру команди шляхом доповнення традиційної взаємодії «людина–людина» каналами «людина–AI» та, у мультиагентних системах, «AI–AI», що зумовлює зміни інформаційних потоків, зворотного зв'язку та механізмів командної координації.

Таким чином, трансформацію комунікацій можна розглядати як посередницький механізм між

технологічними змінами та змінами групової динаміки.

Узагальнимо:

AI Technologies → Communication Transformation → Role Transformation → Coordination → Group Dynamics.

При цьому зв'язок не є лінійним. Зміни групової динаміки, своєю чергою, впливають на характер використання AI. Наприклад, рівень довіри в команді може визначати готовність її членів використовувати AI-рекомендації, а командні норми — встановлювати допустимий рівень AI-автономності.

Тому остаточна модель повинна враховувати двосторонній зв'язок між технологічним і соціальним середовищами.

Трансформація групової динаміки та командних результатів в умовах взаємодії людини й AI. Інтеграція технологій штучного інтелекту у процеси розроблення програмного забезпечення не обмежується зміною окремих професійних функцій або каналів комунікації. Зміни, що виникають у технічному середовищі, поступово поширюються на соціальну структуру команди та впливають на механізми її функціонування. У результаті трансформуються групові ролі, способи координації діяльності, лідерство, довіра, згуртованість, конфліктна взаємодія та процеси прийняття колективних рішень.

Отже, вплив AI на команду доцільно розглядати як багаторівневий процес, у якому технологічні зміни опосередковано впливають на групову динаміку через трансформацію ролей та інформаційних потоків.

Узагальнено цей процес можна представити:

AI → Technical Environment → Roles + Communication → Group Dynamics → Team Outcomes.

При цьому зазначена залежність не є суто лінійною. Зміни групової динаміки можуть, своєю чергою, визначати характер використання AI, рівень його прийняття командою та ступінь делегування йому окремих функцій.

Одним із безпосередніх наслідків зміни ролей та інформаційних потоків є трансформація механізмів координації. У традиційній команді координація забезпечується переважно через розподіл завдань між людьми, синхронізацію їхніх дій, зустрічі, обмін повідомленнями, code review та інші форми організаційної взаємодії.

В умовах AI-augmented Software Engineering частина координаційних функцій може підтримуватися або автоматизуватися AI-компонентами. AI може аналізувати стан завдань, визначати залежності між ними, формувати рекомендації щодо пріоритетів, узагальнювати результати роботи та забезпечувати інформаційну підтримку прийняття рішень.

У результаті формується модель:

Human Coordination → Hybrid Coordination → Human − AI Coordination.

При цьому AI не обов'язково замінює координатора. Значно важливішою є зміна характеру координаційної функції: людина переходить від безпосереднього контролю кожної операції до організації правил взаємодії між людьми, програмними засобами та AI-компонентами.

Аналогічні зміни відбуваються у сфері лідерства. Технічний лідер команди дедалі більше виконує функції не лише контролю виконання завдань, а й AI governance — визначення меж використання AI, контролю якості його результатів, розподілу відповідальності та забезпечення людського контролю над критичними рішеннями.

Таким чином:

Leadership = Technical Leadership + AI Governance.

Це означає, що компетентність лідера сучасної команди повинна включати не тільки технічні та організаційні знання, а й здатність керувати гібридною людино-AI взаємодією.

Окремого значення набуває проблема довіри. У традиційній команді довіра формується переважно між людьми та пов'язана з професійною компетентністю, відповідальністю, передбачуваністю поведінки та якістю комунікації.

Поява AI створює новий об'єкт довіри: *Human → Trust → AI.*

Водночас виникає і зворотне питання: *Human → Verify → AI Output.*

Тобто команда повинна визначати, якою мірою AI-рекомендація може бути прийнята без додаткової перевірки.

Надмірна довіра до AI може призводити до некритичного прийняття помилкових результатів, тоді як надмірна недовіра може знижувати потенційні переваги використання технології.

Тому оптимальним є не максимальний рівень довіри, а калібрована довіра, яка відповідає рівню ризику конкретного AI-рішення. Психологічна безпека також набуває нового виміру. Член команди може, наприклад, не погоджуватися з AI-рекомендацією або ставити під сумнів рішення, сформоване AI. Тому командна культура повинна забезпечувати можливість відкрито ставити під сумнів як людські, так і AI-generated рішення.

Зміна розподілу функцій між людьми та AI може створювати нові джерела конфліктів.

До них належать:

- різне ставлення членів команди до використання AI;
- незгода щодо рівня допустимої автоматизації;
- конфлікти щодо розподілу відповідальності;
- різна довіра до AI-generated результатів;
- конкуренція між альтернативними AI-рішеннями;
- невизначеність щодо професійних ролей.

Тому конфлікт у сучасній команді може виникати вже не тільки між людьми: *Human ↔ Human,* але й навколо взаємодії: Human ↔ AI.

Водночас AI може мати і позитивний вплив на згуртованість, якщо використовується як спільний інформаційний ресурс, що забезпечує команду доступом до єдиного інформаційного контексту.

Отже, вплив AI на згуртованість не є однозначним і залежить від способу організації командної взаємодії.

Особливо важливою є трансформація процесів колективного прийняття технічних рішень.

Традиційна модель може бути представлена: $Human_1 + Human_2 + ... + Human_n \rightarrow Decision.$

У AI-augmented команді: $Human_1 + Human_2 + AI \rightarrow Decision.$

А у мультиагентному середовищі: $Human\ Team + AI_1 + AI_2 + ... + AI_n \rightarrow Decision.$

AI може генерувати альтернативні рішення, оцінювати варіанти, виявляти потенційні ризики та надавати аргументацію. Проте остаточне колективне рішення повинно враховувати не лише технічну оцінку AI, а й контекст проєкту, ризики, бізнес-вимоги та професійний досвід команди.

Таким чином, виникає гібридне прийняття рішень, у якому AI розширює інформаційну базу рішення, але не обов'язково замінює колективну відповідальність людей.

Концептуальна модель трансформації командної взаємодії. На основі проведеного аналізу пропонується концептуальна модель трансформації взаємодії в команді розробників програмного забезпечення під впливом AI.

Її центральна ідея полягає у тому, що AI-технології змінюють технічне середовище, через яке трансформуються ролі та інформаційні потоки, що своєю чергою впливають на групову динаміку та командні результати.

Модель має такий загальний вигляд:

AI Technologies → Technical Environment → Roles + Information Flows → Group Dynamics → Team Outcomes.

При цьому блок Group Dynamics включає:

GD = {Coordination, Leadership, Trust, Cohesion, Conflict, Psychological Safety, Decision Making}.

А Team Outcomes можуть характеризуватися такими показниками, як:

TO = {Productivity, Quality, Coordination Efficiency, Decision Quality, Adaptability}.

Отже, модель пов'язує технологічний рівень, комунікаційно-рольовий рівень, соціальний рівень та результативний рівень функціонування команди.

Запропонований підхід дозволяє перейти від розгляду AI як окремого інструменту програміста до його розуміння як компонента соціотехнічної системи команди.

На відміну від підходів, які зосереджуються переважно на продуктивності AI coding assistants, автоматизації окремих етапів SDLC або ефективності AI-агентів, запропонована концепція поєднує технологічні та соціальні аспекти командної взаємодії.

Основна наукова ідея полягає у тому, що: ефект від інтеграції AI у Software Engineering визначається не лише функціональними можливостями AI-систем, а й тим, як їх інтеграція змінює ролі, інформаційні потоки, координацію та групову динаміку команди.

Таким чином, ефективність AI-augmented команди доцільно оцінювати не лише за кількістю згенерованого коду або швидкістю виконання окремих операцій, а й за якістю командної взаємодії та результативністю соціотехнічної системи в цілому.

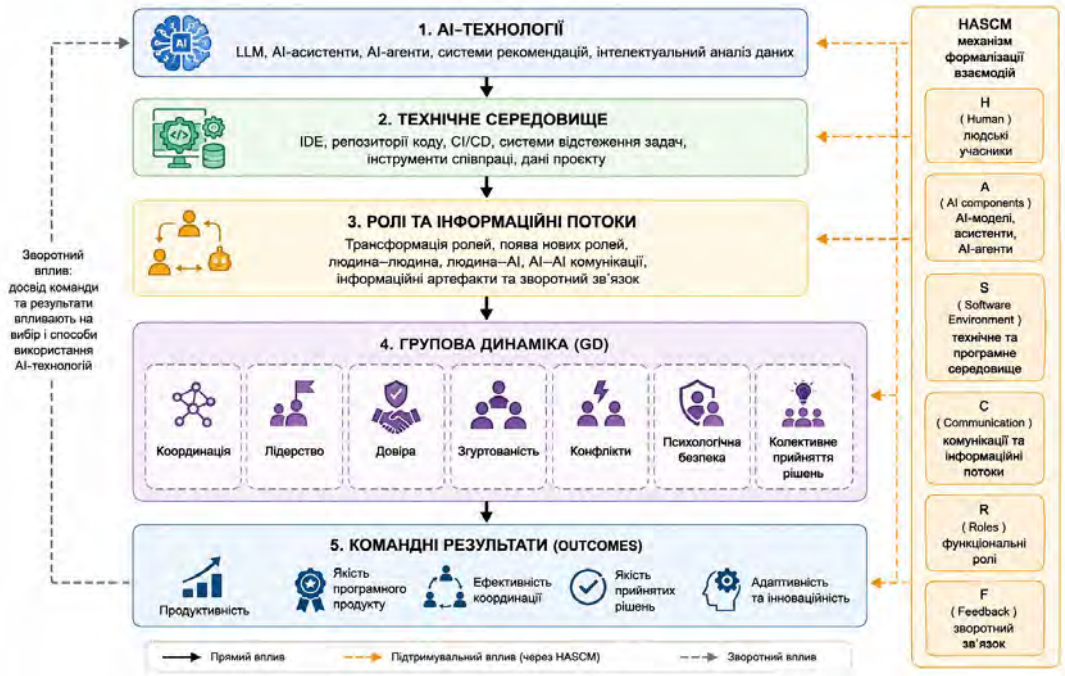


Рис. 2. Концептуальна модель трансформації командної взаємодії в умовах інтеграції AI-технологій

На рис. 2 представлено концептуальну модель трансформації командної взаємодії в умовах інтеграції AI-технологій. Модель відображає послідовний взаємозв'язок між AI-технологіями, технічним середовищем, трансформацією функціональних ролей та інформаційних потоків, груповою динамікою і командними результатами. Центральним соціальним рівнем моделі є групова динаміка, що охоплює координацію, лідерство, довіру, згуртованість, конфлікти, психологічну безпеку та колективне прийняття рішень. Формалізацію взаємодії між людськими учасниками, AI-компонентами, програмним середовищем, ролями, комунікаціями та зворотним зв'язком забезпечує запропонована модель HASCM. Передбачений також зворотний вплив командного досвіду та отриманих результатів на подальший вибір і способи використання AI-технологій.

Висновки. Проведене дослідження показало, що інтеграція технологій штучного інтелекту у процеси розроблення програмного забезпечення зумовлює не лише технологічну модернізацію окремих етапів Software Engineering, а й комплексну трансформацію командної взаємодії. AI перестає розглядатися виключно як програмний інструмент окремого розробника та поступово стає активним компонентом соціотехнічного середовища, у якому функціонує команда розробників ПЗ. Дослідження має концептуально-аналітичний характер. Запропонована модель визначає теоретичну структуру взаємопов'язаних компонентів, яка може бути використана як основа для подальшої емпіричної перевірки.

Проведений аналіз показує, що вплив AI на групову динаміку має системний характер.

На відміну від підходу, за якого AI розглядається виключно як інструмент автоматизації, запропонована модель трактує його як компонент соціотехнічної архітектури групи.

Вплив штучного інтелекту на команду розробників програмного забезпечення не обмежується трансформацією технологічних процесів, а поширюється на соціотехнічне середовище команди, змінюючи групову динаміку, комунікації, ролі, механізми координації та прийняття колективних рішень.

Обгрунтовано доцільність розгляду сучасної команди розробників як гібридної соціотехнічної системи, у якій одночасно взаємодіють люди, AI-компоненти, програмні засоби та інфраструктура розроблення. Встановлено, що інтеграція AI впливає одночасно на технічне та соціотехнічне середовища IT-команди, створюючи нові форми взаємодії та змінюючи традиційні механізми організації колективної діяльності.

Показано, що трансформація ролей та комунікацій є механізмом, через який технологічна інтеграція AI поширюється на групову динаміку. Зміни стосуються координації, лідерства, довіри, згуртованості, конфліктної взаємодії, психологічної безпеки та колективного прийняття рішень. Особливої актуальності набуває формування каліброваної довіри до AI-рекомендацій, за якої рівень довіри визначається характером завдання, потенційним ризиком та необхідністю людської верифікації.

На основі узагальнення результатів дослідження запропоновано концептуальну модель трансформації командної взаємодії, яка описує послідовний взаємозв'язок:

AI Technologies → Technical Environment → Roles + Information Flows → Group Dynamics → Team Outcomes.

Запропонована модель дозволяє розглядати вплив AI не як безпосередню зміну одного параметра діяльності команди, а як багаторівневий соціотехнічний процес, у якому технологічні зміни опосередковано впливають на командні результати через трансформацію ролей, комунікацій та групової динаміки.

Практичне значення результатів полягає у можливості використання запропонованої концепції для аналізу та проєктування командних процесів у сучасних Software Engineering організаціях, визначення меж делегування завдань AI, формування правил людського контролю, організації комунікацій між людьми та AI-компонентами, а також оцінювання ефективності гібридної командної взаємодії.

Концептуальна модель являє інтеграцію групової динаміки, комунікаційних процесів та людино-машинної взаємодії в єдину концептуальну модель функціонування IT-команди в умовах використання AI. На відміну від підходів, орієнтованих переважно на індивідуальну взаємодію розробника з AI-інструментом, запропонована модель переносить фокус на командний рівень і враховує взаємозалежність ролей, відповідальності, інформаційних потоків, координації та зворотного зв'язку (Таблиця 1).

Наукова новизна дослідження полягає у розробленні концептуальної моделі трансформації групової динаміки та комунікацій IT-команди в умовах взаємодії з технологіями штучного інтелекту. На відміну від підходів, у яких Human–AI interaction переважно розглядається як взаємодія окремого фахівця з AI-інструментом або як засіб підвищення продуктивності виконання окремих завдань Software Engineering, запропонована модель розглядає AI як компонент соціотехнічного середовища IT-команди, що впливає на структуру ролей, інформаційні потоки, координацію та процеси прийняття рішень.

Відмінною особливістю моделі є інтеграція трьох взаємопов'язаних рівнів: групової динаміки IT-команди, комунікаційної взаємодії та людино-машинної взаємодії. На рівні групової динаміки враховуються ролі учасників, характер їхньої координації, розподіл відповідальності, взаємна довіра та зміна структури командної взаємодії. На комунікаційному рівні моделюються формування та

Таблиця 1

Відмінності запропонованої моделі від існуючих підходів до Human–AI interaction в IT-командах

Ознака	Існуючі підходи	Запропонована модель
Об’єкт взаємодії	Людина – AI	IT-команда – AI
Роль AI	Інструмент/асистент	Компонент соціотехнічної системи
Групова динаміка	Не є центральним об’єктом	Є складовою моделі
Комунікація	Передавання запитів/результатів	Контекст, намір, інформаційні потоки, feedback, координація
Ролі та відповідальність	Переважно не формалізуються	Враховуються у структурі взаємодії
Командна координація	Розглядається побічно	Безпосередньо включена в модель
Результат	Виконання окремого завдання	Трансформація взаємодії та командного результату
Рівень моделювання	Окрема взаємодія людина–AI	**Єдина модель «людина – IT-команда – AI»**

передавання інформації, контекст взаємодії, намір, програмний артефакт і зворотний зв’язок. На рівні Human–AI interaction AI розглядається не лише як інструмент виконання команди, а як активний програмний компонент, який може виконувати різні функціональні ролі у процесі Software Development Lifecycle.

Таким чином, запропонована модель відрізняється від існуючих підходів не окремим розглядом AI, комунікації або групової динаміки, а їх формалізованим поєднанням у межах єдиного соціотехнічного представлення IT-команди. Це створює основу для дослідження того, як зміна способу взаємодії з AI трансформує розподіл ролей, структуру комунікацій, механізми координації та командні результати.

Перспективи подальших досліджень полягають у розробленні формальних метрик оцінювання HASCM, моделюванні комунікаційної мережі «людина–людина–AI», визначенні кількісних показників ефективності гібридної команди та проведенні емпіричної валідації запропонованої моделі на реальних проєктах розроблення програмного забезпечення.

На основі проведеного аналізу запропоновано Human–AI Group Dynamics Transformation Model (HAGDTM).

Модель передбачає шість взаємопов'язаних компонентів:

Компонент	**Зміст трансформації**
AI-інтеграція	включення AI у групову діяльність
Комунікації	зміна інформаційних потоків
Ролі та лідерство	перерозподіл функцій
Соціально-психологічна взаємодія	довіра, безпека, конфлікти, згуртованість
Колективне знання	інтеграція людських та AI-generated ресурсів
Прийняття рішень	розширення та оцінювання альтернатив

Компоненти даної моделі формують систему зворотних зв'язків:

AI↔Communication↔Roles↔Trust↔Knowledge↔ Decision↔Group Dynamics.

Таким чином, група розглядається як динамічна соціотехнічна система.

Для подальшого емпіричного дослідження запропоновано інтегральний індекс трансформації групової взаємодії:

$$GT_I = w_C C + w_r R + w_t T + w_k K + w_d D + w_p P - w_a A,$$

де: GT_I — Group Transformation Index;

C — ефективність комунікації;

R — ефективність розподілу ролей;

T — рівень довіри;

K — якість колективного знання;

D — якість прийняття рішень;

P — продуктивність;

A — рівень залежності від AI;

w_i — вагові коефіцієнти.

Показник є концептуальним і призначений для подальшої емпіричної операціоналізації.

Зростання ефективності можливе лише за умови якісної інтеграції AI у комунікаційні та соціальні процеси групи.

Саме через комунікацію AI впливає на довіру, координацію, розподіл ролей та формування спільного знання.

Тому: *AI → Communication → Group Dynamics* можна розглядати як один із базових причинно-наслідкових ланцюгів запропонованої моделі.

Практичне значення отриманих результатів полягає у можливості використання моделі при організації діяльності студентських, наукових, проєктних, управлінських та програмно-інженерних команд.

Перспективи подальших досліджень полягають у емпіричній перевірці запропонованої моделі, аналізі комунікаційних графів людино-ШІ груп, визначенні показників AI centrality, когнітивної додатковості та групової ефективності, а також у

досліджені специфіки AI-assisted комунікацій у програмно-інженерних командах.

Література

1. Guimaraes E., Nascimento N. AI in the Software Development Lifecycle: Insights and Open Research Questions // Companion Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25). New York : Association for Computing Machinery, 2025. P. 1353–1357. DOI: https://doi.org/10.1145/3696630.3730538.
2. Sergeyuk A., Golubev Y., Bryksin T., Ahmed I. Using AI-based coding assistants in practice: State of affairs, perceptions, and ways forward // Information and Software Technology. 2025. Vol. 178. Art. 107610. DOI: https://doi.org/10.1016/j.infsof.2024.107610
3. Cabrero-Daniel B., Herda T., Pichler V., Eder M. Exploring Human-AI Collaboration in Agile: Customised LLM Meeting Assistants // Agile Processes in Software Engineering and Extreme Programming. XP 2024 / eds. D. Šmite, E. Guerra, X. Wang, M. Marchesi, P. Gregory. Cham : Springer, 2024. Vol. 512. P. 163–178. (Lecture Notes in Business Information Processing). DOI: https://doi.org/10.1007/978-3-031-61154-4_11
4. Abrahão S., Grundy J., Pezzè M., Storey M.-A., Tamburri D. A. Software Engineering by and for Humans in an AI Era // ACM Transactions on Software Engineering and Methodology. 2025. Vol. 34, No. 5. Art. 129. P. 129:1–129:46. DOI: https://doi.org/10.1145/3715111
5. He J., Treude C., Lo D. LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead // ACM Transactions on Software Engineering and Methodology. 2025. Vol. 34, No. 5. Art. 124. P. 124:1–124:30. DOI: https://doi.org/10.1145/3712003
6. De La Cruz E., Le H., Meduri K., Nadella G. S., Gonaygunta H. Redefining the Programmer: Human-AI Collaboration, LLMs, and Security in Modern Software Engineering // Computers, Materials & Continua. 2025. Vol. 85, No. 2. P. 3569–3582. DOI: https://doi.org/10.32604/cmc.2025.068137
7. Salomon M., Chin K. D., Holmes R., Fritz T., Murphy G. C. An Exploration of How Generative AI Affects Workflow and Collaboration in a Software Engineering Course // Proceedings of the 2025 ACM SIGPLAN International Symposium on SPLASH-E (SPLASH-E '25). New York : Association for Computing Machinery, 2025. P. 42–54. DOI: https://doi.org/10.1145/3758317.3759680
8. Kasatkin D., Yuskovych-Zhukovska V., Bogut O. The Features of Orchestration for Multi-Agent Artificial Intelligence Systems Applied to the Tasks of HR Process Automation in IT Companies // Herald of Khmelnytskyi National University. Technical Sciences. 2026. Vol. 361, No. 1. P. 444–451. DOI: https://doi.org/10.31891/2307-5732-2026-361-61
9. Yuskovych-Zhukovska V., Bogut O. The impact of artificial intelligence on the efficiency of IT project management // Innovation-driven development in education, digital economy, and applied technologies : monograph. Katowice : The University of Technology in Katowice Press, 2025. P. 228–232. URL: http://www.wydawnictwo.wst.pl/uploads/files/e28c93cb04ed5369e1c94cb14399b880.pdf

Yuskovych-Zhukovska V.I. Group dynamics and communication in the interaction of IT team and artificial intelligence.

This article examines the transformation of group dynamics and communication within software development teams as artificial intelligence technologies are integrated into software engineering processes. The relevance of this research stems from the shift from using AI as a support tool for individual specialists to its integration into team processes, the software development life cycle, and mechanisms for collective technical decision-making. The methodological foundation is the socio-technical approach, according to which an IT team is viewed as a dynamic hybrid system of interaction between people, AI components, software tools, and infrastructure. It has been established that the integration of AI simultaneously changes both the technical and socio-technical environments of the team, transforming information flows, the distribution of roles and responsibilities, and the mechanisms of coordination, leadership, trust, cohesion, psychological safety, and collective decision-making. It is shown that traditional "human – human" communication is supplemented by "human – AI" and "AI – AI" channels, resulting in the formation of a heterogeneous communication network. The paper justifies the need for calibrated trust in AI recommendations, human verification of results, and the retention of responsibility for final decisions by team members. Based on a synthesis of the results, the Human – AI Group Dynamics Transformation Model (HAGDTM) is proposed, which describes the interrelationship between AI technologies, the technical environment, roles, information flows, group dynamics, and team performance. For further empirical operationalization of the model, an integrated Group Transformation Index (GTI) is proposed, which takes into account communication effectiveness, role distribution, trust levels, the quality of collective knowledge, decision-making, productivity, and the level of dependence on AI. The practical significance of the results lies in the possibility of using the model to analyze and design team processes, determine the limits of task delegation to AI, organize human oversight, and evaluate the effectiveness of hybrid interaction in software engineering teams.

Keywords: *software developer, IT team, human-AI interaction, group communication*

Юскович-Жуковська Валентина Іванівна - кандидат технічних наук, доцент, заступник декана факультету кібернетики ПВНЗ «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука»,
https://orcid.org/0000-0002-4236-1467
valivanivna1@gmail.com
valivanivna@i.ua

Дата першого надходження статті 04.08.2026.
Дата прийняття статті до друку після рецензування 26.08.2026.
Дата публікації 30.09.2026.

Стаття з відкритим доступом, відповідно до умов ліцензії **Creative Commons (CC BY 4.0)**