

DOI: <https://doi.org/10.52256/2710-3986.2-97.2022.09>

УДК 378.004.4

Олександр Нестеренко,
доктор технічних наук, доцент,
завідувач кафедри інформаційних технологій,
Міжнародний Європейський Університет,
Київ, Україна
ORCID ID 0000-0001-5329-889X

ІНФОЛОГІЧНЕ МОДЕЛЮВАННЯ ПРОСТОРУ ОСВІТИ З ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті встановлено, що на якість освіти з інженерії програмного забезпечення впливає урахування в її методичному забезпеченні сучасних промислових трендів та ІТ-тенденцій, а також динаміки команд розробників програмного забезпечення. Визначено, що напрямками навчання мають бути не лише характеристики технологій і програмних проєктів, а й людські чинники відповідно до методології групової динаміки і комунікацій, у тому числі й на необхідності урахування спілкування та роботи в команді.

Досліджено деталізацію промислових трендів та оцінено стан урахування сучасних тенденцій в освітніх програмах. З'ясовано, що компетентності майбутнього фахівця з програмної інженерії, які визначаються як поєднання знань, технічних навичок та людських диспозицій, перетинаються в областях формалізованого і неформалізованого, що приводить до того, що навчальні задачі зазвичай стають багатофакторними та породжуються освітні проблеми, пов'язані з розширенням фундаментальних (традиційних) наборів компетентностей і навичок.

Удосконалено підхід до навчально-методичного забезпечення освіти з інженерії програмного забезпечення на основі моделювання інтеграції вимог сучасних трендів розвитку індустрії та галузі ІТ, міжнародних рекомендацій та можливостей потенціалу набуття «м'яких» навичок. Запропоновано онтологічну інфологічну модель предметної області як основу бази знань інтелектуальної інформаційної системи підтримки освітньої діяльності з інженерії програмного забезпечення.

Ключові слова: *якість освіти, інформаційні технології, групова динаміка, онтології, інфологічне моделювання простору освіти*

Постановка проблеми. Останніми роками у процесах формування інтелектуального потенціалу людства шляхом надання особистості можливостей для вдосконалення та самореалізації усе більш особливого значення набуває підвищення якості освіти. У наш час цифрових трансформацій, коли основою технологічного розвитку стає програмне забезпечення, в області комп'ютерної освіти ключовою дисципліною виступає інженерія програмного забезпечення (далі – ПЗ). Ця дисципліна значною мірою відрізняється від інших предметів своєю багатогранністю за рахунок інтегрування технічних і технологічних доміант з інженерними підходами, з економічними категоріями і положеннями менеджменту. Тому її методологічне забезпечення пов'язане із певними труднощами, а організація процесу навчання потребує постійного оновлення підходів, методів і засобів щодо ознайомлення студентів з сучасними тенденціями у сфері технологій і програмування.

Аналіз дослідженості проблеми у світовій науковій літературі [1-3] свідчить, що на розвиток програмної інженерії істотний вплив здійснюють сучасні промислові тенденції. Перехід до нового щабля – «Індустрії 4.0», яка характеризується роботизацією і кіберсоціалізацією, несе хвилю нових, ще не бачених викликів до сфери програмування. Водночас завдяки складності програмних процесів розвиток передових програмних технологій, практика застосування інновацій в цій сфері потребують зміщення акценту з індивідуальної роботи окремих розробників на діяльність багатофункціональних команд.

Відповідно до цих умовиводів основою набуття знань з ПЗ має стати нова парадигма глибокого розуміння сучасних технологічних процесів та здобуття професійних навичок на основі практики. Вочевидь, у таких умовах досягти потрібного синергетичного ефекту неможливо без формування у закладах вищої освіти відповідного інформаційно-освітнього середовища, при

проектуванні та формуванні якого пріоритетним має бути впровадження засобів інтелектуалізації, принципів відкритої освіти і відкритої науки.

Аналіз останніх досліджень та публікацій. За більш ніж піввікову історію розвитку програмної інженерії в багатьох публікаціях, починаючи від фундаментальних видань до наявних публікацій у наукових журналах і працях конференцій наведена широка панорама тем, що охоплюють всі етапи і технології розробки програмних систем. За ці десятиліття інженерія програмного забезпечення істотно змінилася, водночас вона як досить нова інженерна дисципліна все ще розвивається на основі сукупності основоположних принципів. Це підтверджує систематичний аналіз низки видань за певний період, кожне з яких пропонувало свої власні набори принципів програмної інженерії [1].

У роботі [2] також проведений систематичний огляд наукових праць з досліджень тенденцій розробки програмного забезпечення (далі – ПЗ), процесів і практик розробки, підходів до навчання та еволюції ПЗ з часом. Проведений аналіз свідчить про наявні розриви між індустрією ПЗ та освітою. Вказується, що в освітній сфері багато сучасних технологічних тенденцій не знаходять належного відображення. .

Водночас у роботі [3] подано систематичне дослідження публікацій за сорок років, в яких представлені дослідження особистості в програмній інженерії. Автори показали, що за цей час найбільш повторюваними дослідницькими темами, серед інших, були освіта, характеристики особистості інженера-програміста та індивідуальна продуктивність. Головний висновок свідчить й про те, що ця сфера ще не є зрілою і потребує додаткових досліджень.

Дійсно, профіль роботи інженера-програміста включає не тільки так звані «тверді навички» (технології програмування, побудова архітектур), але також й «м'які навички», такі як обізнаність про ефекти команди та подібні людські фактори. В статті [4] відзначається, що цим навичкам, як правило,

важко навчити в аудиторіях, а отже, нинішня освіта, в основному, зосереджена на твердих, а не на м'яких навичках. Проте, оскільки розроблення ПЗ дедалі усе більше поширюється у глобальному масштабі, значення м'яких навичок у програмуванні швидко зростає. Тому пропонується поєднувати останні досягнення та передовий досвід для покращення навчальних програм, долаючи розрив між очікуваннями галузі та тим, що може надати освіта.

Ці питання у контексті командної розробки ПЗ розглядає науковий та освітній напрямок групової динаміки і комунікацій (далі – ГДК). У роботі [5, с. 38] зазначено, що «Складність розробки програмних систем не є основним чинником успіху. Визначальну роль в успішності програмного бізнес-проєкту відіграє команда розробників, її розмір, розташування і розподіл повноважень, а також нормативні обмеження та вимоги управління».

Дослідження [6] показує що ефективне використання методології групової динаміки забезпечує вищий моральний дух команди розробників програмних додатків і приводить до підвищення продуктивності праці. Також зрілість робочих груп, з психологічної точки зору, тісно пов'язана з позитивним ефектом від навчання щодо групової психології розвитку.

Водночас у [7] підтверджується, що однією з проблем у процесах розробки ПЗ є проблема атрофії знань з вищезазначених питань. В роботі вивчено кілька факторів динаміки малих команд, що впливають на процеси управління знаннями.

Особливе місце в проблематиці програмної інженерії займає керування роботами команди розробників програм у процесі виконання плану проєктів. Дослідження [8] підтверджує, що визначення критеріїв ефективності роботи команди й оцінка процесів і продуктів проєктів з використанням загальних методів планування і контролю робіт безпосередньо базується на методології ГДК, а ефективність малих робочих груп тісно пов'язана з психологічними рисами особистості, сумлінністю та екстраверсією членів команди.

У статті [9] розглянуто теоретичні та практичні питання формування та керування малою групою розробників ПЗ у розрізі методичних та практичних аспектів використання курсу "Групова динаміка і комунікації" для забезпечення успішної адаптації майбутніх фахівців у складі команди з урахуванням низки особливостей. Один із шляхів реалізації викладання цієї дисципліни в контексті викликів сучасних інформаційних технологій, який визначений на основі аналізу підходів вітчизняних та закордонних учених до вивчення соціально-психологічних особливостей групи та групової динаміки, розглянуто у статті [10].

Комунікації є одним з основних засобів забезпечення програмних проєктів, адже чимало з проєктів натрапляють на серйозні проблеми, або взагалі не вдаються, оскільки зацікавлені сторони по-різному розуміються та/або використовують різну термінологію. Так само часто відбувається й в освітньому середовищі. Вочевидь, досягти потрібних результатів навчання з урахуванням зазначених тенденцій неможливо без формування у закладах освіти відповідного інформаційного середовища. В роботі [11] показано доцільність впровадження в цей процес трансдисциплінарного підходу як найбільш узагальненого та універсального, що долає межі і недоліки дисциплінарної роздробленості, надає більш повне уявлення про світ і фундаментальні взаємозв'язки. Для підтримки освітньої та наукової діяльності в педагогічних системах закладів вищої освіти необхідним є створення та розвиток хмарного навчально-дослідницького середовища в контексті пріоритетів відкритої науки, відкритої освіти та формування Європейського дослідницького простору [12].

Забезпечити проблеми спілкування та формування освітнього контенту в цих умовах пов'язане передусім зі створенням відповідних інформаційних моделей.

Питанням моделювання освітнього середовища присвячено чимало публікацій. Так, наприклад, в роботі [13] представлено дослідження,

спрямоване на покращення дизайну сучасного навчального середовища виробництва програмного забезпечення за допомогою набору моделей для опису не тільки процесів, а й інформації, що генерується процесами, для підтримки виконання процесів і технологічної інфраструктури, необхідної в контексті практичного курсу програмної інженерії. Останнім часом набула значної популярності в різних сферах побудова інформаційних моделей на базі комп'ютерних онтологій [14]. Метою роботи [15] є представлення інтелектуальної системи для підтримки оцінювальної діяльності у віртуальних навчальних середовищах, архітектура якої складається з трьох рівнів: інтелектуальний рівень, на якому розроблено онтологічну модель, інфраструктурний рівень, розроблений із рішенням хмарних обчислень, і рівень, який містить моделі регулювання.

У виданні [16] одним із перших були порушені питання використання онтологій у дослідженнях програмного забезпечення та програмних проєктів. В основному ця робота висвітлює два застосування онтологій у програмній інженерії та програмних технологіях: обмін знаннями про проблемну область та використання загальної термінології серед усіх зацікавлених сторін. У той же час метою статті [17] є спроба запропонувати онтологічний підхід до забезпечення інформаційної підтримки прийняття управлінських рішень у задачах Software Engineering Management, визначення людських ресурсів на основі інформаційних моделей.

Отже, науково-методичні передумови освіти в області ПЗ потребують подальших досліджень в контексті пріоритетів сучасних технологічних трендів та групової психології, а також створення інформаційного навчально-дослідницького середовища. Концептуальний базис дослідження цих питань безперервно прогресує, а особливості освітнього середовища постійно розвиваються. Тому актуальність завдання полягає у врахуванні цих процесів у методологічних принципах навчання з проєктування та розробки якісного програмного продукту.

Мета статті – розробити інфологічну модель освітньої предметної області на основі дослідження інформаційно-технологічних особливостей інженерії програмного забезпечення в сучасних умовах.

Об'єктом дослідження є освітні процеси з інженерії програмного забезпечення в сучасних умовах динаміки індустріальних трендів розробки програмного забезпечення та підвищення важливості людських чинників. Предмет дослідження – інфологічна модель освітнього простору з інженерії програмного забезпечення.

Виклад основного матеріалу дослідження. Відправним посилком досліджень, що витікає з вищенаведеного аналізу джерел, є визнання відчутного впливу на зміст, методи та організаційні форми навчання з ПЗ характерних рис сфери ІТ, які на сучасному етапі полягають у зростанні динаміки технологічних змін, у різкому збільшенні обсягів інформації, потрібної для опрацювання та навчання. Крім того, розвиток передових програмних технологій, світова практика застосування інновацій потребують зміщення освітніх акцентів з індивідуальної роботи окремих розробників на діяльність багатофункціональних команд, на врахування значної чисельності аспектів, пов'язаних із комунікаціями.

Виготовляти якісне і надійне програмне забезпечення можуть тільки колективи, що володіють певною культурою і зрілістю. Це пов'язане із тим, що загальнолюдські властивості під впливом професійної діяльності починають виступати як специфічні професійні навички. Такий перехід потребує не лише нового підходу до організації процесу навчання з програмної інженерії, але й нових методів і засобів щодо ознайомлення студентів з діяльністю команд. Зазначені притаманні предметній області ПЗ властивості продукують освітні проблеми, пов'язані з переглядом фундаментальних (традиційних) наборів компетентностей і навичок, необхідністю створення таких умов навчання, щоб постійно відстежувати швидкі технологічні зміни. Загалом це призводить до того, що освітні задачі в

області ІІЗ набувають ознак багатофакторності і слабкої структурованості, що врешті-решт впливає на якість навчання. У цих умовах на передній план інформаційного процесу освітньої діяльності виходить задача пошуку нового знання для подолання зазначених проблем. Можна зробити висновок, що ця діяльність в царині ІІЗ є безпосередньо пов'язаною з поданням інформаційної картини предметної області.

Дослідники та фахівці, що пропонують інноваційні підходи до підтримки освітнього процесу, у своїй більшості спираються на відповідні моделі та методики, серед яких переважають ідеї щодо гармонійного поєднання пов'язаних дисциплін та різноманітних галузевих технологій, важливості у створенні належних навчальних програм та виборі придатної методики навчання відповідних тем технологічних трендів та співпраці суб'єктів галузі щодо спільних цілей освіти.

Ці підходи відповідають парадигмам Global Computing Education, наведеним в останньому релізі CC2020 серії звітів Computing Curricula від Association for Computing Machinery (ACM) та IEEE Computer Society (IEEE-CS). Після тривалої перерви з виходу SE 2014 (Software Engineering Curricular Volume) міжнародні організації повернулися до розгляду керівних принципів програм бакалаврству у комп'ютерних професіях, зокрема ІІЗ. Один з чотирьох основних принципів, яких дотримувалася цільова група при розробці CC2020, пов'язаний з необхідністю враховувати в освітньому процесі майбутні тенденції та бачення промисловості, відстежувати зміни в технологіях та дослідженнях з соціологічного спектра. Автори документа CC2020 вважають, що професійні та наукові спільноти повинні відігравати провідну роль у наданні підтримки вищій освіті різними способами, зокрема формулюванням навчальних методичних рекомендацій.

З аналізу цього документу можна зробити висновок, що усі комп'ютерні дисципліни, особливо ІІЗ, повинні будуватися на професійних ноу-хау індустріальних трендів, трендів інформаційних технологій, у тому

числі й на необхідності урахування спілкування та роботи в команді на основі «м'яких навичок» (рис. 1).

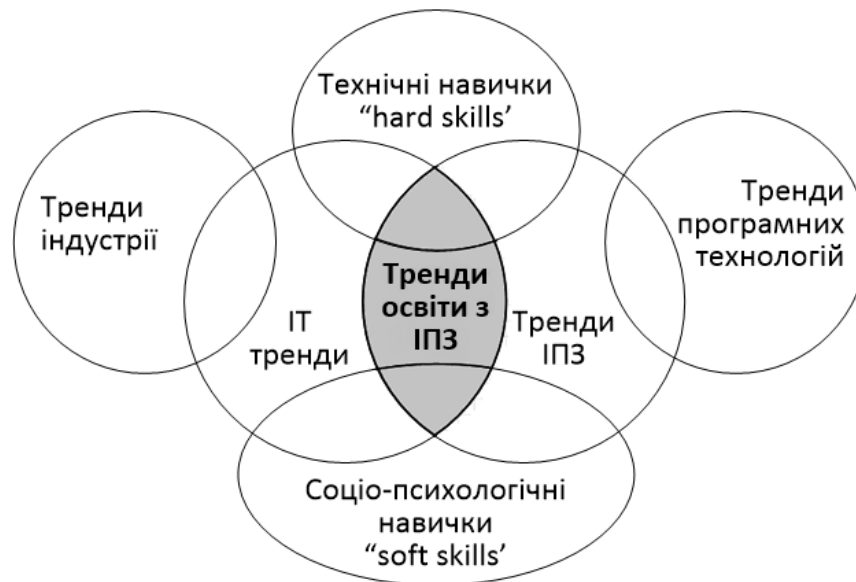


Рис. 1. Вплив на сучасні тренди освіти з інженерії програмного забезпечення / Influence on modern trends in software engineering education

Щоб проілюструвати багатофакторність предметної області програмної інженерії, яка спричиняє постійне збільшення інформаційного навантаження на сферу освіти за спеціальністю ІПЗ, проведено деталізацією сучасних галузевих трендів (табл. 1). Ці напрямки, які або вже існують, або очікуються у найближчому майбутньому, визначені на основі аналізу різних джерел бізнес-аналітики, а також враховують дані, наведені у СС2020 та у [18].

Таблиця 1.

**Деталізація трендів індустрії, ІТ та програмної інженерії /
Detailing trends in industry, IT and software engineering**

Індустріальні тренди	Тренди ІТ	Тренди програмних технологій та програмної інженерії
<ul style="list-style-type: none"> - Індустрії 4.0 - Цифрова трансформація (Digital transformation) - Глобальне підключення (Global connection) - Багато обчислювальних засобів - Збільшення програмного забезпечення - Конфіденційність даних і цифрова етика (Data privacy and digital ethics) - Роботизація (Robotics) - Злиття технологій (Merger of technologies) - Гнучке підприємство (Enterprise Agile) - Кіберсоціалізація (Cyber-socialization) 	<ul style="list-style-type: none"> - Інтернет речей (Internet of Things – IoT) - Хмарні обчислення (Cloud computing) - Високопродуктивні обчислення (High-performance computing) - Блокчейн (Blockchain) - Кібербезпека (Cybersecurity) - Аналіз даних (Data science) - Система систем (System of system) - Автономія, вбудовані системи (Embedded systems) - Інтерактивні технології (Interactive technologies) - Вузький штучний інтелект (Narrow artificial intelligence) - Когнітивні технології (Cognitive technologies) - Повсюдний комп'ютинг (Ambient computing) - Поєднання біології та обчислювальної техніки (біокомп'ютер) - Квантовий комп'ютинг (Quantum computing) 	<ul style="list-style-type: none"> - Відкрите програмне забезпечення (Open Source Software – OSS) - Швидкі зміни, потреба в гнучкості та масштабуванні (Agile Software Development – ASD, Large-scale Agile) - DevOps - Підвищений акцент щодо зручності використання та цінності (Usability, User Experience – UUE) - Критичність програмного забезпечення та необхідність надійності (Safeti, Security, Trust – SST) - Global Software Engineering (GSE) - Зростання потреб у комерційно готових доступних продуктах (COTS), повторному використанні застарілих версій (Reusability) - Збільшення інтеграції програмного забезпечення та системна інженерія - Програмна інженерія для автономних систем (Software engineering for autonomous systems – SAS) - Бережливий стартап (Lean Startup -LS) - Безперервна інженерія програмного забезпечення (Continuous Software Engineering – CSE)

Відповідність освітніх програм сучасним парадигмам. З аналізу спектра технологій, наведених у табл. 1, з якими стикаються розробники ПЗ, зрозуміло, що освітні програми мають бути гнучкішими у підходах щодо технічних компетенцій. Однак цей аналіз також показує, що набуття

студентами соціопсихологічних навичок роботи у командах потребує ще більш гнучкої структури освітніх програм, урахування архітектурних стратегій підприємства і керування командами в рамках розвитку можливостей персоналу. Це підтверджує проведене експертами з закладів освіти та за участі авторів оцінювання вибірки з десяти трендів, яке визначає співвідношення (у відсотках) вагомості технічних і соціо-психологічних навичок у кожному тренді (рис. 2).

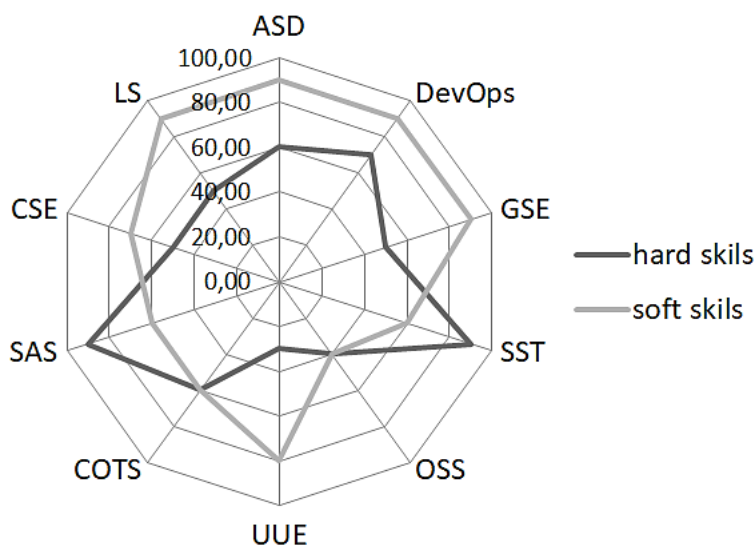


Рис. 2. Співвідношення технічних і соціо-психологічних навичок в освіті з інженерії програмного забезпечення відповідно до сучасних технологічних трендів / The ratio of technical and socio-psychological skills in education in software engineering in accordance with modern technological trends

Як вказувалось вище в огляді літератури, у своїй більшості сучасні галузеві тренди ще не досягли в університетах різних країн статусу навчальних дисциплін із загально визнаною програмою навчання. Врахування освітніми програмами підготовки в напрямках сучасних галузевих трендів та набуття соціо-психологічних навичок у вітчизняних закладах освіти відповідає цим судженням, що можна проілюструвати наступними прикладами. Так у вибірці з двадцяти освітніх програм з інженерії програмного забезпечення, що реалізуються у вітчизняних технічних університетах, які входять до академічного рейтингу «Топ-200 Україна 2021», проведеного Центром

міжнародних проєктів «Євроосвіта», лише у п'яти програмах передбачено обов'язкову освітню компоненту, що повинна, крім іншого, знайомити студентів з сучасним розвитком технологій (має різні варіації назв – методи та засоби комп'ютерних інформаційних технологій, інформаційні технології, вступ до ІТ, вступ до фаху, методи та засоби перетворення інформації тощо). Певною мірою розгляд сучасних трендів передбачається лише у вибіркових дисциплінах. Також у вищевказаній вибірці лише у шести програмах серед обов'язкових освітніх компонентів зазначається «Групова динаміка і комунікації». Попри те, що організаційною основою для розробки ПЗ є проєкт, дисципліна «Менеджмент проєктів програмного забезпечення», або «Управління ІТ-проєктами», зустрічається тільки у восьми програмах. Таким чином, незважаючи на те, що виконання програмних проєктів і ефективність ІТ-бізнесу загалом безпосередньо пов'язані з можливостями персоналу, переважно з людськими чинниками, ці питання не знайшли належного відображення в освітніх програмах.

Ще одним напрямком, на який в освітніх програмах з ПЗ на сьогодні уваги майже не приділяється, це питання прийняття рішень та їх інформаційно-аналітичного забезпечення. Особливо це стосується й застосування в цій діяльності інтелектуальних технологій підтримки прийняття рішень, аналізу даних та методів і засобів штучного інтелекту. Важливий чинник діяльності членів і менеджерів групи та проєктів полягає у виборі найкращого рішення, необхідності подолання впливу можливої неузгодженості рішень, що пов'язується, зокрема, з поняттям концептуальної, або ментальної моделі навколишнього світу, з якістю і повнотою зібраної і поданої для опрацювання інформації. Основні проблеми, що при цьому виникають, обумовлені тим, що формулювання критеріїв вибору є суто творчим процесом, а якість підготовлених рекомендацій безпосередньо залежить не лише від рівня професіоналізму та відповідальності членів групи, а й від володіння ними сучасних технологій прийняття рішень, уявлення про

те, як людина приймає рішення та знання щодо особливостей сприйняття людиною самого факту підготовки рішення.

Слід зазначити, що розроблення комп'ютерних програм у сучасних реаліях є масовою професією, але одночасно й однією з найскладніших професій, пов'язаною з володінням математичним й логічним мисленням. Крім того, програміст повинен мати здатності до колективної роботи, розуміти інтереси співтовариства та користувачів програмних продуктів. Тому компетенції фахівця з програмної інженерії визначаються як поєднання знань («знаю що»), технічних навичок («знаю як») та людських диспозицій («знаю чому»). Ці компетенції перетинаються у формалізованому і неформалізованому доменах (рис. 3). З одного боку – це знання як науковий факт та практикована поведінка з категоричністю оцінок технічної раціональності, з іншого – диспозиції особистості, що пов'язані з інтелектуальними, соціальними та моральними якостями, оцінними судженнями, що не так легко піддаються беззаперчній оцінці. Технічні навички також частково підпадають під дію культурних норм, які під впливом досвіду праці позначаються на практикованій поведінці.



Рис. 3. Формальне і неформальне у компетентностях майбутнього фахівця з інженерії програмного забезпечення / Formal and informal in the competence of the future specialist in software engineering

Отже, для підтримки будь-яких методичних підходів до освітнього процесу з ІІЗ необхідно враховувати необхідність подолання проблем узгодженості формального і неформального, багатофакторності та невизначеностей, а також трансдисциплінарності. Очевидно, що суттєвої допомоги у забезпеченні цієї підтримки може надати подання та аналіз на різних рівнях освітнього процесу значної сукупності гетерогенних структурованих і неструктурованих даних. З одного боку – це дані, що характеризують науково-технічний базис основних ІТ-трендів та проектний менеджмент, наприклад, за міжнародними рекомендаціями РМВОК (Project Management Body of Knowledge), з іншого – це дані, що описують соціально-психологічні особливості персоналу за методологією понять ГДК, зокрема відповідно до рекомендацій People Capability Maturity Model (P-CMM).

Таким чином, для забезпечення якісного опрацювання та підвищення об'єктивності формування характеристик навчально-методичних матеріалів предметну область (ПдО) простору освіти з ІІЗ доцільно представити у вигляді певної інфологічної моделі. Така модель повинна найбільш точно відображати структуру та деталізацію об'єктів, понять, критеріїв ПдО у вигляді ієрархічних та мережевих структур елементів і враховувати їх взаємний вплив в умовах багатофакторності. Погрішності на етапі такої структуризації зазвичай призводять до утворення хибних освітніх програм, які зумовлюють отримання незадовільних компетентностних результатів.

Одним із методів моделювання, що дає змогу вирішувати зазначену проблему, є онтологічне представлення ПдО як детальний опис за допомогою концептуальної схеми, що містить інформацію про властивості, а також про відношення між поняттями та об'єктами. У загальному випадку ця онтологія містить інформаційні описи на основі об'єктно-орієнтованої процедури формалізації, а кожна модель може бути представленою певною таксономією. Така структура відображає розуміння проблеми людиною. Важливо, що онтологія дає змогу представляти опис компонент певною формальною

мовою, яка може інтерпретуватися деякою процедурою (алгоритмом) і бути реалізованою інформаційною технологією. З застосуванням онтологій інформаційний супровід розв'язання освітніх задач здійснюється на основі інтерпретаційних функцій вибору, які керують процесом постачання даних для опрацювання.

Таким чином, за онтологічною термінологією світ (*world*), у якому відбуваються процеси освіти з ІПЗ, можна представити інфологічною моделлю, що є значною за обсягом системою таксономії ієрархії взаємодії класів (понять) і концептів, які задаються за допомогою бінарних відношень (рис. 4).

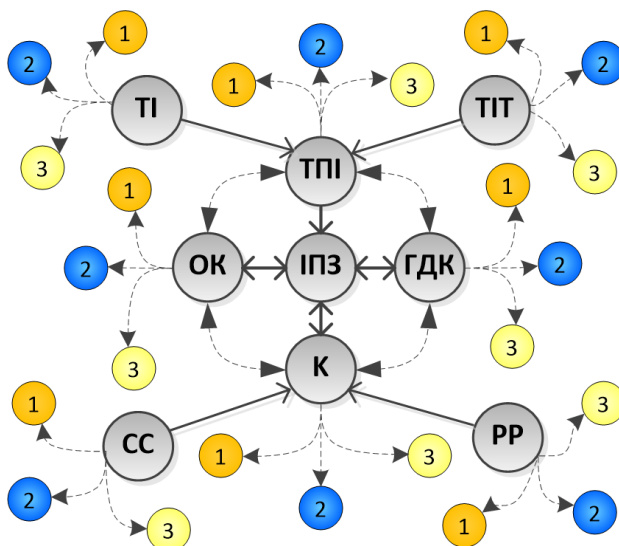


Рис. 4. Інфологічна модель освітньої предметної області (ТІ – індустріальні тренди; ТІТ – тренди ІТ; ТПІ – тренди програмної інженерії; ІПЗ – освітня програма «Інженерія програмного забезпечення»; ОК – освітні компоненти програми; К – освітні компетентності і програмні результати навчання; ГДК – групова динаміка і комунікації; СС – серія міжнародних рекомендацій Computing Curricula; РР – Р-СММ та РМВОК; 1, 2, 3 і так далі – концепти і поняття класів моделі) / Infological model of educational subject area

В моделі прийнято позначення суцільною лінією зі стрілкою стандартного відношення типу is-a, тобто відношення між деякими двома класами, коли один є підкласом (нащадком) іншого класу (який, у свою чергу, є суперкласом для нащадку). Відношення is-a застосовується у контексті наслідування, тобто можна, наприклад, сказати, що ТПІ наслідує певні

властивості ТІТ. Пунктирні стрілки означають відношення типу «has-a», тобто взаємозв'язок, в якому об'єкти (концепти і поняття) належать до іншого об'єкту (класу). Множинні відношення has-a утворюють дерево приналежності. Також в моделі введено додаткове специфічне відношення is part, тобто є частиною, що позначається суцільною лінією з двосторонніми стрілками.

Наведена модель фактично є мета-моделлю, або онтологією подання, мета якої описати область представлення знань, створити мову для специфікації інших онтологій нижчих рівнів.

Ця мета-модель відповідає стандарту вищої освіти за спеціальністю 121 «Інженерія програмного забезпечення» ступеня вищої освіти «бакалавр» (далі—Стандарт), де об'єктами предметної області визначено програмне забезпечення, процеси, інструментальні засоби та ресурси розробки, супроводження та забезпечення якості програмного забезпечення. Ці об'єкти враховуються елементами моделі ТІ, ТІТ, ТІІ, на основі яких формується перелік ОК освітньої програми, як обов'язкових, так і вибірових. Зважаючи на важливість освітньої компоненти «Групова динаміка і комунікації» для розвитку загальних і фахових компетенцій, у цій моделі вона виділена в окремий клас. При формуванні профілю освітньої програми перелік компетентностей і програмних результатів навчання, наведених в Стандарті, уточнюється та доповнюється додатковими компетентностями і результатами навчання на основі актуальних міжнародних рекомендацій. Це також є важливим підґрунтям формування переліку ОК.

Серед вимог до внутрішнього забезпечення якості освіти, передбачених Стандартом, необхідним є забезпечення наявності інформаційних систем для ефективного управління освітнім процесом. В межах загальної проблеми облаштування середовища навчальної діяльності на основі інформаційних систем актуальною є задача розроблення обґрунтованих методів опрацювання інформаційних освітніх ресурсів та побудови на їх основі технологічних програмних засобів інтелектуалізації цих систем.

Підґрунтям процесів інтелектуалізації є бази знань, які зручно формувати на основі онтологій. Наведена вище інфологічна модель, побудована на засадах онтологічних описів класів, концептів та відношень освітнього простору, може бути концептуальною основою формування правил цих баз [19]. Такі інтелектуальні системи можуть використовуватись не лише відповідальними особами університетів для формування та розвитку навчально-методичних матеріалів, а й викладачами і студентами в освітньому процесі.

Висновки з дослідження і перспективи подальших розвідок.

Результати дослідження підтверджують, що в освіті з інженерії програмного забезпечення усе більш важливу роль відіграє гармонійне поєднання набуття студентами компетентностей як в царині сучасних трендів програмних технологій, процесів і практик, які застосовні та популярні в промислових галузях, так і в сфері соціо-психологічного базису командної роботи та культурних норм професійної діяльності.

Разом із тим освітній діяльності притаманний постійно зростаючий неструктурований текстовий та мультимедійний контент, опрацювання якого пов'язане з відчутними проблемами. Це протиріччя породжує стримування інноваційного розвитку сфери освіти з ІІЗ, що може негативно впливати на темпи зростання цієї галузі. Проведене дослідження певною мірою заповнює вищевказані прогалини, пропонуючи підхід до моделювання освітнього середовища та запровадження відповідних прогресивних інформаційних технологій для підтримки освітнього процесу.

Таким чином до елементів наукової новизни отриманих результатів дослідження можна віднести запропоновані удосконалення навчально-методичного забезпечення освіти з ІІЗ на основі інфологічного моделювання простору освіти, яке враховує сучасні тренди розвитку індустрії та галузі ІТ та положення міжнародних рекомендацій, а також пропозиції з інформаційно-аналітичної підтримки освітнього процесу.

Практична значущість результатів дослідження полягає у тому, що побудована модель та пропозиції зі створення відповідної інтелектуальної інформаційної системи можуть стати основою процесу осучаснення методології освіти в сфері ПЗ.

Заплановані наступні дослідження, орієнтовані на подальшу розбудову онтологічної моделі, наповнення її контентом та створення відповідної бази знань, що має стати основою інформаційної системи освітнього середовища університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Al-Sarayreh K. T., Meridji K., Abran A. Software engineering principles: A systematic mapping study and a quantitative literature review. *Engineering Science and Technology*. 2021. 24(3). P. 768-781. <https://doi.org/10.1016/j.jestch.2020.11.005>
2. Cico O., Jaccheri L., Nguyen-Duc A., Zhang, H. Exploring the intersection between software industry and Software Engineering education – A systematic mapping of Software Engineering Trends. *Journal of Systems and Software*. 2021. 172. 110736. URL: <https://doi.org/10.1016/j.jss.2020.110736>
3. Cruz S., da Silva F. Q. B. Capretz L. F. Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior*. 2015. 46. P. 94–113. URL: <https://doi.org/10.1016/j.chb.2014.12.008>
4. Kuhrmann M., Femmer H., Eckhardt J. Controlled Experiments as Means to Teach Soft Skills in Software Engineering. In book: *Overcoming Challenges In Software Engineering Education: Delivering Non-Technical Knowledge And Skills*. IGI Global, 2014. 10. P. 180-197. URL: <https://doi.org/10.4018/978-1-4666-5800-4>.
5. Групова динаміка ефективних команд розробників : монографія / С. М. Пасєка, Н. М. Пасєка та ін. Івано-Франківськ : ІФНТУНГ, 2022. 302 с.
6. Grena L., Goldmanb A., Jacobsson C. The perceived effects of group developmental psychology training on agile software development teams. *IEEE Software*. 2020. 37(3). P. 63–69. URL: <https://doi.org/10.1109/MS.2019.2955675>
7. Basri S., O'Connor R. The Impact of Software Development Team Dynamics on the Knowledge Management Process. *23rd International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*. 2011. P. 339–342.
8. Endriulaitienė A., Cirtautienė L. Team effectiveness in software development: the role of personality and work factors. *Business: Theory and Practice*. 2021. 22. P. 55-68. URL: <https://doi.org/10.3846/btp.2021.12824>
9. Мнушка О. В., Савченко В. М. Формування та керування командою розробників програмного забезпечення. *Вісник НТУ «ХП»*. Серія: Інформатика

та моделювання. Харків: НТУ «ХПІ». 2020. №1 (3). С. 99–112. URL: <https://doi.org/10.20998/2411-0558.2020.01.09>

10. Бич Ю. Навчальний курс "Групова динаміка і комунікації" в контексті підготовки сучасних фахівців у галузі інформаційних технологій. Вісник Львівського університету. Серія педагогічна. 2017. Вип. 32. С. 511–519.

11. Rostoka M., Guraliuk A., Cherevychnyi G., Vyhovska O., Poprotskyi I., Terentieva N. Philosophy of a Transdisciplinary Approach in Designing an Open Information and Educational Environment of Institutions of Higher Education. Revista Romaneasca pentru Educatie Multidimensionala, 2021. 13(3). P. 548–567. <https://doi.org/10.18662/rrem/13.3/466>

12. Bykov V. Yu., Shyshkina M. P. The conceptual basis of the university cloud-based learning and research environment formation and development in view of the open science priorities. Information Technologies and Learning Tools. 2018. 68(6). P. 1–19.

13. Naranjo J.F.R, Rossi A.C., Souza S.N.A., Becerra J.L.R. Designing a reference architecture for a collaborative software production and learning environment. 13th Latin American Conference on Learning Technologies (LACLO 2018), 2019. P. 408–415. DOI: 10.1109/LACLO.2018.00075

14. Nesterenko O., Trofymchuk O. Patterns in forming the ontology-based environment of information-analytical activity in administrative management. Eastern-European Journal of Enterprise Technologies. 2019. 5/2 (101). P. 33–42. DOI: 10.15587/1729-4061.2019.180107

15. Lopez R.B.S., Gurrola I.I.M. Intelligent System for Customizing Evaluation Activities Implemented in Virtual Learning Environments: Experiments and Results. Computacion y sistemas, 2022. 26 (1). P. 473–484. URL: <https://doi.org/10.13053/CyS-26-1-4182>

16. Piattini M., Calero C., Ruiz, F., (Ed.). Ontologies for Software Engineering and Software Technology. Springer. 2006. URL: <https://doi.org/10.1007/3-540-34518-3>

17. Nesterenko O., Selin Yu. The Teams Information Model for Software Engineering Management. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), 2021. 1. P. 341–344. URL: <https://doi.org/10.1109/CSIT52700.2021.9648737>

18. Boehm B. A view of 20th and 21st Century Software Engineering. 28th International Conference on Software Engineering (ICSE'06), 2006. P. 12–29. URL: <https://doi.org/10.1145/1134285.1134288>

19. Комп'ютерні онтології та їх використання у навчальному процесі. Теорія і практика : Монографія / С. О. Довгий, В. Ю. Велічко, Л. С. Глоба, О. Є. Стрижак та ін. Київ : Інститут обдарованої дитини. 2013. 310 с.

Стаття надійшла до редакції 03.06.2022

Oleksandr Nesnerenko. Infological modeling of the space for software engineering education.

It is determined that the quality of software engineering education is influenced by modern industrial trends and IT trends, software technology tendencies, as well as the dynamics of software development teams. Given the wide range of technologies faced by groups of developers, it is determined that teaching should be based not only on the characteristics of technologies and projects, but also on the personal characteristics of programmers in accordance with the methodology of group dynamics and communications. In general, this leads to the fact that learning tasks are usually multifactorial and information overloaded. This produces educational problems related to expansion the fundamental (traditional) sets of competencies and skills and requires methodological delineation for the prospect of their wider application in pedagogical systems of software engineering. To support this area, the target audience should include industry employers and other actors. It was revealed that current issues also include problems of quality and completeness of information collected and submitted to support educational problems, questions to acquaint students with information support of decision-making processes. The approach to software engineering education is improved by modelling the integration of the requirements of current industry and IT industry trends, international recommendations, and the potential of soft skills. An ontological model of the subject area of software engineering as a basis of knowledge base of intelligent information system for application within pedagogical systems of higher education is offered. The following researches are planned, focused on the further development of the ontological model, filling it with content, creating an appropriate knowledge base and improvement of methodical materials.

Key words: *quality education, information technologies, group dynamics, ontologies.*

REFERENCES

1. Al-Sarayreh, K. T., Meridji, K., & Abran, A. (2021). Software engineering principles: A systematic mapping study and a quantitative literature review. *Engineering Science and Technology*, 24(3), 768-781. <https://doi.org/10.1016/j.jestch.2020.11.005> [in English].
2. Cico, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2021). Exploring the intersection between software industry and Software Engineering education – A systematic mapping of Software Engineering Trends. *Journal of Systems and Software*, 172, 110736. URL: <https://doi.org/10.1016/j.jss.2020.110736> [in English].
3. Cruz, S., da Silva, F. Q. B., & Capretz, L. F. (2015). Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior*, 46, 94-113. URL: <https://doi.org/10.1016/j.chb.2014.12.008> [in English].
4. Kuhrmann, M., Femmer, H., & Eckhardt, J. (2014). Controlled Experiments as Means to Teach Soft Skills in Software Engineering. In book: *Overcoming Challenges In Software Engineering Education: Delivering Non-Technical Knowledge*

And Skills. IGI Global, 10, 180-197. URL:<https://doi.org/10.4018/978-1-4666-5800-4>

5. Pasięka, M. S., Pasięka, N. M., Romanyshyn, Yu. L., & Sheketa, V. I. (2022). Group dynamics of effective development teams. IFNTUNH, Ivano-Frankivsk. [in Ukrainian].

6. Grena, L., Goldmanb, A., & Jacobsson, C. (2020). The perceived effects of group developmental psychology training on agile software development teams. *IEEE Software*, 37(3), 63-69. URL:<https://doi.org/10.1109/MS.2019.2955675> [in English].

7. Basri, S., & O'Connor, R. (2011). The Impact of Software Development Team Dynamics on the Knowledge Management Process. *Proceeding of the 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, 339-342. [in English].

8. Endriulaitienė, A., & Cirtautienė, L. (2021). Team effectiveness in software development: the role of personality and work factors. *Business: Theory and Practice*, 22, 55-68. URL: <https://doi.org/10.3846/btp.2021.12824>

9. Mnushka, O. V., & Savchenko, V. M. (2020). Forming and managing a team of software developers. *Visnyk Natsionalnoho tekhnichnoho universytetu "KhPI"*, 1(3), 99-112. [in Ukrainian]. URL: <https://doi.org/10.20998/2411-0558.2020.01.09> [in English].

10. Bych, Yu. (2017). Educational course "Group Dynamics and Communications" in the context of training modern specialists in the field of information technology. *Visnyk Lvivskoho universytetu. Seriya pedahohichna*, 32, 511-519. [in Ukrainian].

11. Rostoka, M., Guraliuk, A., Cherevychnyi, G., Vyhovska, O., Poprotskyi, I., & Terentieva, N. (2021). Philosophy of a Transdisciplinary Approach in Designing an Open Information and Educational Environment of Institutions of Higher Education. *Revista Romaneasca pentru Educatie Multidimensionala*, 13(3), 548-567. URL: <https://doi.org/10.18662/rrem/13.3/466> [in English].

12. Bykov, V. Yu., & Shyshkina, M. P. (2018) The conceptual basis of the university cloud-based learning and research environment formation and development in view of the open science priorities. *Information Technologies and Learning Tools*, 68(6), 1-19. [in English].

13. Naranjo, J.F.R, Rossi, A.C., Souza, S.N.A., & Becerra, J.L.R. (2019). Designing a reference architecture for a collaborative software production and learning environment. *Proceeding of the 13th Latin American Conference on Learning Technologies (LACLO 2018)*, 408-415. DOI: 10.1109/LACLO.2018.00075 [in English].

14. Nesterenko, O., & Trofymchuk, O. (2019). Patterns in forming the ontology-based environment of information-analytical activity in administrative management. *Eastern-European Journal of Enterprise Technologies*, 5/2 (101), 33-42. DOI: 10.15587/1729-4061.2019.180107 [in English].

15. Lopez, R.B.S., & Gurrola, I.I.M. (2022). Intelligent System for Customizing Evaluation Activities Implemented in Virtual Learning Environments:

Experiments and Results. *Computacion y sistemas*, 26 (1). 473-484. URL: <https://doi.org/10.13053/CyS-26-1-4182> [in English].

16. Piattini, M., Calero, C., & Ruiz, F., (Ed.). (2006). *Ontologies for Software Engineering and Software Technology*. Springer. URL: <https://doi.org/10.1007/3-540-34518-3> [in English].

17. Nesterenko, O., & Selin, Yu. (2021). The Teams Information Model for Software Engineering Management. *Proceeding of the 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, 1, 341-344. URL: <https://doi.org/10.1109/CSIT52700.2021.9648737> [in English].

18. Boehm, B. (2006). A view of 20th and 21st Century Software Engineering. *Proceeding of the 28th International Conference on Software Engineering (ICSE'06)*, 12-29. URL: <https://doi.org/10.1145/1134285.1134288> [in English].

19. Computer ontologies and their use in the educational process. Theory and practice: Monograph / S. O. Dovhyi, V. Yu. Velichko, L. S. Hloba, O. Ye. Stryzhak [and other]. Kyiv: Instytut obdarovanoi dytyny. 2013. 310 p. [in Ukrainian].