ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Сучасний розвиток індустрії IT-технологій та програмної інженерії пов'язаний з розробкою програмного забезпечення (ПЗ), що базується на використанні принципів прикладного системного аналізу та знання-орієнтованих, інтелектуальних підходів до створення складних інформаційно-управляючих систем різного призначення: в технічних, організаційних, соціально-економічних застосуваннях, тощо. Саме ці проблеми визначаються та вирішуються в роботах таких провідних вчених як Ф.І. Андон, О.В. Палагін, К.Л Лаврищева, К. Амблер (C. Ambler), М. Фаулер (M. Fowler), Дж. Хайсмит (D. Heihsmith) та інших фахівців.

Значна частина цих досліджень присвячена вирішенню проблем розробки та практичного використання новітніх, так званих гнучких або адаптивних методів та засобів створення ПЗ, які на відміну від традиційних мають забезпечити можливість адекватно враховувати постійні зміни, що відбуваються в процесі проектування та реалізації ПЗ. Ці зміни, передусім, стосуються вимог майбутніх користувачів програмного забезпечення щодо його властивостей, які мають бути врахованими в кінцевому програмному продукті, і що, в свою чергу, визначають його працездатність та якість. Таким чином, коректно визначені та специфіковані вимоги щодо ПЗ є важливим інформаційним ресурсом в процесах його розробки, і тому саме питання інженерії вимог, і зокрема, трасування вимог є актуальною темою багатьох досліджень в цій області. В той же час аналіз останніх робіт в цьому напрямку показує, що практично всі існуючи гнучкі методології розробки програмного забезпечення, такі як, наприклад, XP (eXtreme Programming), ASD (Adaptive Software Development) або Scrum, ще не достатньо ефективно вирішують цю проблему. Однією з головних причин, що зумовлюють цю ситуацію, є суттєве протиріччя, що виникає між наявними у гнучких підходах принципами та практиками відмови розробників ПЗ від великого обсягу проектної документації та необхідністю мати дієві механізми для накопичення, аналізу та управління інформацією щодо вимог користувачів ПЗ. Це протиріччя може бути подоланим шляхом розробки нових інтелектуальних моделей та інформаційних технологій саме для підвищення ефективності вирішення комплексу задач інженерії вимог в гнучких методологіях розробки програмного забезпечення, і таким чином постановка задачі даного дослідження є досить актуальною та практично значущою.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана на кафедрі автоматизованих систем управління НТУ «ХПІ» відповідно до завдань прикладних держбюджетних НДР МОН України: «Розробка систем підтримки прийняття рішень в складних інформаційно-управляючих комплексах» (ДР № 0109U002424) та «Розробка інтелектуальних моделей та технологій для підвищення ефективності проектування та супроводу складних програмних систем» (ДР № 0111U002288) де здобувач, відповідно, брав участь як співвиконавець окремих розділів та відповідальний виконавець.

Мета і задачі дослідження. Метою дисертаційної роботи є підвищення повноти та точності трасування вимог в гнучких процесах створення програмного забезпечення шляхом розробки знання-орієнтованих моделей та інформаційної технології накопичення, аналізу та управління даними щодо вимог користувачів.

Для досягнення цієї мети поставлено такі задачі:

– аналіз особливостей гнучких процесів розробки ПЗ та визначення в них характерних проблем накопичення, обробки та управління вимогами на прикладі методології Scrum;

– застосування принципів прикладного системного аналізу та кібернетичних методів для аналізу та синтезу ефективних механізмів адаптивного трасування вимог в технологічному процесі розробки програмного забезпечення;

– розробка математичних моделей та інформаційних технологій для підвищення ефективності трасування вимог до ПЗ за критеріями повноти та точності їх опрацювання;

– розробка шаблонів проектування (патернів), які б забезпечили можливість побудови інструментального середовища (CASE-засобу) розробки ПЗ з адаптивним механізмом трасування вимог;

– програмна реалізація прототипу запропонованого CASE-засобу;

– експериментальне дослідження ефективності запропонованого підходу та розробка практичних рекомендацій для впровадження в реальні проекти по розробці та супроводу програмного забезпечення.

Об'єктом дослідження є гнучкі процеси розробки та супроводу компонентного програмного забезпечення складних інформаційних систем.

Предметом дослідження є моделі та інформаційні технології адаптивного трасування вимог в гнучких процесах розробки та супроводу ПЗ.

Методи дослідження. Наукові та практичні аспекти дисертації базуються на концепціях прикладного системного аналізу, і, зокрема, на застосуванні кібернетичних принципів управління складними системами та технологічними процесами для структурування гнучких процесів розробки програмного забезпечення; на використанні математичного апарату загальної теорії множин і методів матричних обчислень, експертних методів для моделювання та розробки адаптивних процедур управління вимогами та проектними артефактами; на методах об'єктно-орієнтованого аналізу та синтезу ПЗ з використанням уніфікованої мови системного моделювання UML для формалізації процедур проектування інструментальних програмних засобів; на застосуванні методів статистичного аналізу експериментальних даних для кількісної оцінки ефективності запропонованого підходу.

Наукова новизна одержаних результатів:

1) вперше запропонована знання-орієнтована модель адаптивного процесу трасування вимог, яка на відміну від існуючих підходів використовує механізм побудови сфокусованого проектного інтерфейсу розробника ПЗ на основі застосування функції ступеню інтересу, що дозволяє скоротити витрати часу та

підвищити точність опрацювання вимог при їх імплементації у відповідні проектні артефакти;

2) удосконалено загальну організаційно-технологічну схему гнучкої розробки ПЗ за методологією Scrum за рахунок застосування запропонованого механізму трасування вимог і комплексу експертних процедур, що надає можливість отримання кількісних показників при визначенні якості опрацювання текстових специфікацій вимог і встановлення їх пріоритетності та, в кінцевому рахунку, забезпечує підвищення ефективності трасування вимог в цілому;

3) удосконалено інформаційну технологію та методику оцінки ефективності трасування вимог та проектних артефактів в гнучких процесах розробки програмного забезпечення, шляхом побудови еталонної архітектури та патернів проектування для відповідного інтегрованого інструментального середовища, що дозволило провести експериментальне дослідження запропонованого підходу, оцінити працездатність та надати рекомендації щодо практичного впровадження в реальних софтверних проектах;

4) дістали подальшого розвитку підходи до трасування вимог в гнучких процесах розробки складних програмних систем на основі застосування кібернетичних методів управління технологічним процесом розробки ПЗ за умов наявності постійних змін в проектному середовищі та відсутності великого обсягу проектної документації.

Практичне значення одержаних результатів для вирішення задач програмної інженерії полягає в тому, шо розроблено модельно-технологічний інструментарій, включаючи програмно реалізований прототип CASE-засобу, який дозволяє застосувати запропонований механізм адаптивного трасування вимог в гнучких процесах створення ПЗ, зокрема, на платформі відкритого середовища розробки Eclipse.

Результати використані в компанії Bit media e-learning solution GmbH & Co KG (Грац, Австрія), запроваджені в навчальному процесі кафедри АСУ НТУ «ХПІ» у дисциплінах «Основи проектування ПЗ», «Аналіз вимог до ПЗ», «Методи та засоби автоматизації процесів життєвого циклу ПЗ». Ці результати були також використані при проведенні дистанційного навчального практикуму з сучасних проблем програмної інженерії в Альпен-Адріа університеті (Клагенфурт, Австрія), що підтверджено відповідними актами та довідками про використання цих результатів.

Особистий внесок здобувача. Усі наукові результати, викладені в дисертації, отримані здобувачем особисто. Серед них: процедура визначення нечітких значень для оцінки стану вимог у багатовимірному просторі критеріїв; формалізована схема трасування вимог в контексті методології Scrum для гнучкої розробки ПЗ; знання-орієнтована модель адаптивного процесу трасування вимог та прикладна інформаційна технологія для її реалізації; комплексна процедура формування динамічного каталогу вимог для проектної ітерації із урахуванням їх пріоритезації та непротиріччя; шаблони проектування для інтеграції систем управління вимогами та середовищ швидкої розробки ПЗ.

Апробація результатів дисертації. Результати досліджень доповідалися та обговорювалися на: 10-й Міжнародній конференції «УкрПРОГ-2012» (Київ, 2012), 7-й Міжнародній конференції ICTERI-2011 (Херсон, 2011), 6-й Міжнародній конференції ISTA-2008 (Austria, Klagenfurt, 2008), IX Міжнародній конференції «Системний аналіз та інформаційні технології» (Київ, 2007), Міжнародних конференціях «Інформаційні технології: наука, техніка, технологія, освіта, здоров'я» (Харків, 2011, 2012), на наукових семінарах кафедри автоматизованих систем управління НТУ «ХПІ», Інституту прикладної інформатики Альпен-Адріа університету (Австрія, Клагенфурт), кафедри теоретичної та прикладної інформатики Харківського національного університету ім. В. Н. Каразіна та кафедри комп'ютерних систем і мереж Національного аерокосмічного університету ім. М.Є. Жуковського «Харківський авіаційний інститут».

Публікації. По результатам дисертаційного дослідження опубліковано 9 друкованих робіт, з них 5 – у наукових фахових виданнях України, 4 – у збірниках матеріалів і тез доповідей міжнародних конференцій.

Структура роботи. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Повний обсяг дисертації складає 173 сторінки, включаючи 39 рисунків по тексту, 22 таблиці по тексту, список використаних джерел із 109 найменувань на 10 сторінках, 5 додатків на 29 сторінках.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі розкрито сучасний стан проблем програмної інженерії із зосередженням особливої уваги на питаннях гнучкої розробки ПЗ, обґрунтована актуальність теми, та сформульована мета і поставлені основні задачі дисертаційної роботи. Визначені об'єкт, предмет та методи дослідження, наводиться наукова новизна і практична цінність отриманих результатів.

У **першому розділі** систематизовані основні проблеми інженерії вимог (requirements engineering) та визначені деякі їх особливості в гнучких методологіях розробки ПЗ, які мають на меті забезпечити можливість ефективної реалізації, подальшої експлуатації та модернізації (реінжинірингу) ПЗ в умовах постійних змін вимог користувачів щодо його функціональності та показників якості. Проаналізовано основні напрямки сучасних досліджень в області розробки гнучких (agile-) методологій створення ПЗ і підкреслено поширення тенденції щодо використання кібернетичних принципів в цих напрацюваннях: як для організації процесів розробки ПЗ в цілому, так і, зокрема, для створення механізмів управління вимогами в них. Відзначено певне протиріччя, що існує у гнучких підходах, тому що позитивний ефект їх застосування значною мірою забезпечується за рахунок відмови розробників від необхідності вже на початку виконання проекту мати великий обсяг проектної документації. Але саме ця обставина ускладнює процеси управління вимогами, і, зокрема, трасування вимог, та контролю щодо рівня якості їх втілення в кінцевому програмному продукті. Існуючий наразі в більшості гнучких методологій підхід до вирішення цієї про-

блеми за рахунок збільшення числа проектних ітерацій є екстенсивним шляхом, який не дозволяє виявити резерви процесу трасування вимог.

Як один із інтенсивних шляхів підвищення ефективності управління вимогами, і, в першу чергу, процесу трасування вимог, пропонується застосування методів інтелектуальної обробки даних, що не потребує наявності значного обсягу проектної документації. До них відносяться, зокрема, відповідні методи аналізу слабо-структурованої текстової інформації, експертні методи ранжування багатокритеріальних альтернатив, а також відповідні моделі трасування вимог (requirements traceability models). Ці моделі є засобами встановлення зв'язків між різними артефактами життєвого циклу вимог у проекті розробки ПЗ (наприклад, їх текстовими специфікаціями, діаграмами варіантів використання, фрагментами коду тощо), для відстежити ступеню впливу змін у вимогах на характеристики функціональності та якості кінцевого програмного продукту. Наведені існуючи засоби, що застосовуються для безпосередньої реалізації трасування вимог і побудовано їх можливу класифікацію (рис. 1).

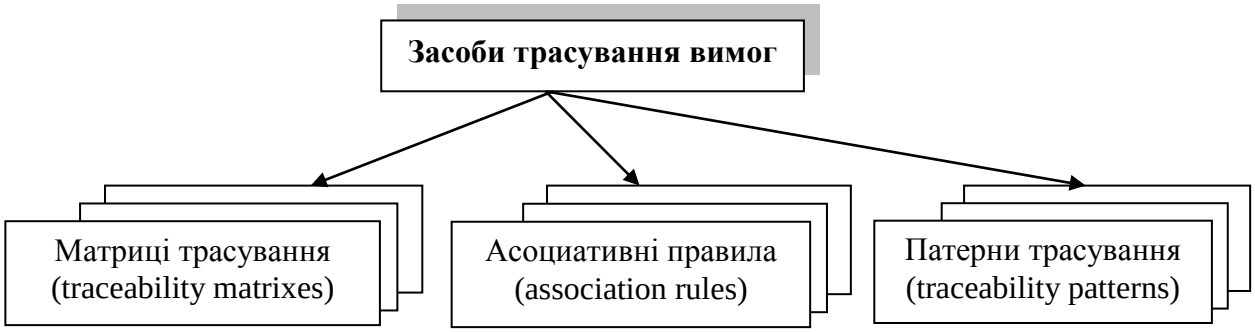

Рис. 1. Класифікація підходів до трасування вимог

Другий розділ присвячено дослідженню методологічних основ розробки перспективних моделей та технологій підвищення ефективності трасування вимог (requirements traceability) в гнучких процесах розробки програмного забезпечення, і розробці загальної схеми адаптивного трасування вимог на прикладі найбільш поширених agile-методології, а саме методології Scrum.

Проаналізовано найбільш поширені сучасні концептуальні моделі процесів трасування вимог, сформульовано основні методологічні принципи їх побудови та запропоновано узагальнюючий формалізований опис таких моделей із використанням фреймових специфікацій. На підставі цього, а також із урахуванням попередньо проаналізованих методів інтелектуальної обробки даних, визначено клас задач управління вимогами, які вирішуються в подальшому в дисертаційні роботі, а саме побудова механізмів адаптивного трасування вимог за умов змін у проектному середовищі на основі застосування кібернетичних принципів і знання-орієнтованих методів обробки даних. Виходячи з цього, формулюються наступні комплексні задачі, постановки та розв'язання яких утворюють методологічне підґрунтя для проведення досліджень у дисертаційній роботі, а саме:

Задача 1. Формалізація та дослідження процесів управління вимогами в гнучких підходах до розробки ПЗ, зокрема, в методології Scrum, для ідентифікації їх кількісних параметрів та забезпечення можливості застосування механізмів зворотного зв'язку (feedback loop).

Задача 2. Розробка моделей та процедур адаптивного трасування вимог в гнучких процесах створення програмного забезпечення із використанням інтелектуальних методів обробки даних.

Задача 3. Розробка та експериментальне дослідження ефективності використання інформаційної технології адаптивного трасування вимог у вигляді програмного CASE-засобу, що імплементує запропонований механізм адаптивного трасування вимог

Для розв'язання *задачі 1* в роботі проаналізовані деякі суттєві особливості процесів управління вимогами в гнучких процесах ПЗ, що зокрема, досить чітко простежуються на прикладі методології Scrum, схему якої наведено на рис. 2. Вимоги спочатку формуються у вигляді каталогу вимог до програмного продукту (Product Backlog – *PB*), який складається замовником проекту (Product Owner – *PO*) разом із командою розробників проекту (Scrum Team – *ST*). Потім ці вимоги приоритизуються *PO* – з точки зору їх важливості для бізнес-логіки всієї системи, та оцінюються групою *ST* – відносно трудомісткості їх подальшої реалізації.

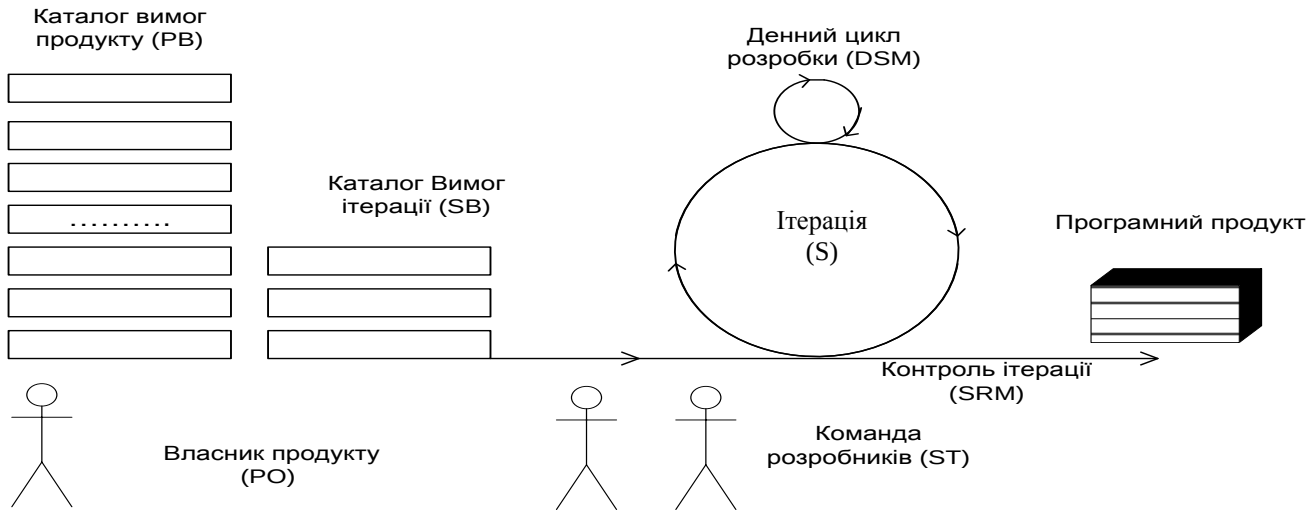

Рис. 2. Основні етапи та артефакти процесу розробки ПЗ за методологією Scrum

При цьому нові вимоги можуть додаватися в *PB* протягом виконання всього проекту. Процес розробки ПЗ відбувається ітеративно, на початку кожної ітерації (Sprint – *S*) з каталогу вимог *PB* формується їх деяка підмножина, яка фіксується в журналі вимог спринту (Sprint Backlog – *SB*). Вимоги, що є зафіксованими в *SB*, не можуть бути змінені протягом цієї ітерації за рішенням ззовні (напр., за рішенням *PO*). Але вони можуть бути скориговані самими розробниками (членами *ST*), якщо в ході виконання поточної ітерації *S* знайдені такі помилки, які можуть завадити її успішному завершенню, або якщо в ході ітерації уявлення про властивості кінцевого продукту значно розширилися і це ви-

магає корекції вимог в *SB*. Для оперативного управління процесом розробки в ході поточної ітерації *S* щодня проводиться робоча нарада (Daily Scrum Meeting – *DSM),* в ході якої всі члени *ST* (при необхідності, і за участю *PO*) обговорюють поточний стан проекту та коригують свої уявлення про хід реалізації вимог у *SB*. В кінці кожної ітерації *S* проводиться демонстраційний мітинг (Sprint Review Meeting – *SRM*), в ході якого: 1) члени *ST* представляють *PO* отримані результати роботи, 2) відбувається редагування загального каталогу вимог *PB* із урахуванням стану вимог в журналі поточного спринту *SB*; 3) формується новий журнал вимог *SB* для наступної ітерації (*S+1*).

Схема на рис. 2 визначає лише організаційні процедури для подолання зазначеного вище протиріччя в процесі управління та трасуванням вимог і тому, з метою розробки технологічних механізмів для вирішення цієї задачі, пропонується реструктурувати її із застосуванням таких принципів сучасної програмної кібернетики (software cybernetics) як побудова нових контурів управління в окремих фазах виконання процесу розробки програмного забезпечення і використання в цих контурах кількісним метрик для забезпечення ефективного зворотного зв’язку.

В модифікованій схемі Scrum-методу розробки ПЗ (рис. 3) введено 2 нових технологічних контури управління із зворотним зв’язком, а саме: (I) контур управління трасуванням вимог у процесі безпосереднього програмування (тобто щоденного виконання завдань проекту), для забезпечення ефекту сфокусованого інтерфейсу розробника ПЗ, (II) контур управління процедурами приоритезации та оцінки якості вимог (requirements priority and quality estimation), що уможливлює формування динамічного каталогу вимог *PB*.

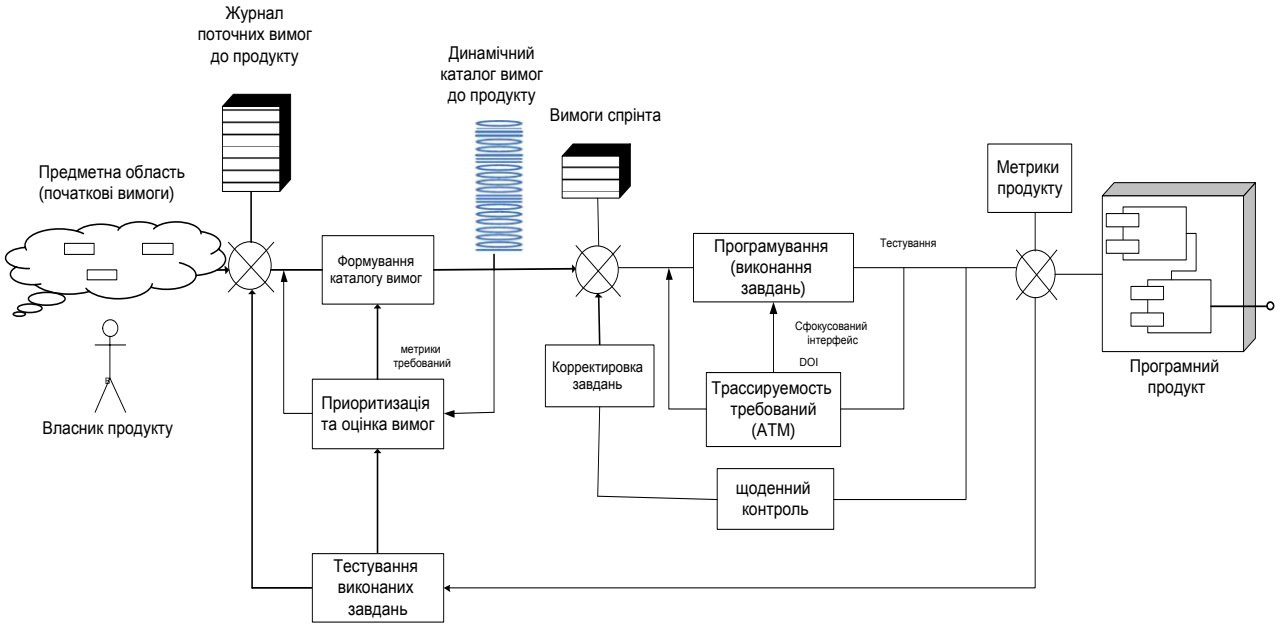

Рис. 3. Кібернетичний підхід до управління гнучким процесом розробки ПЗ

Для комплексного розв’язання *задачі 2*, враховуючи схему на рис. 3, розроблено відповідні механізми обробки даних щодо вимог для застосування їх у контурах управління (I) – (II).

Для реалізації контура управління (I), в контексті класифікації, яка представлена на рис. 1, проаналізовано існуючи підходи до побудови матриці трасування (Traceability Matrix – *TM*), яка традиційно розглядається як відношення

$$TM : (R \times F \to \{0,1\}), \tag{1}$$

де R – це множина вимог; F – множина програмних артефактів.

Такий підхід дозволяє лише визначити підмножину тих артефактів, що пов'язані з певними вимогами, із застосуванням бінарних значень: "0" – "зв'язок є", "1" – "зв'язку немає", але при цьому не враховує питому вагу або важливість того чи іншого артефакту у проекті, і він є типовим для монолітних методологій розробки ПЗ. Проаналізовані деякі інші варіанти побудови *TM*, зокрема, так званої розвинутої *TM (*Advanced Traceability Matrix – *ATM)*, яка на відміну від виразу (1), подається у наступному вигляді

$$ATM : (R \times F \to [0,1]), \tag{2}$$

коли елементи цієї матриці відображають зв'язок певного проектного артефакту із кожною з вимог до ПЗ шляхом подання відповідного дійсного числа, що належить до закритого інтервалу їх значень [0, 1]. Для обчислення елементів матриці (2) застосовуються кількісні значення відповідної функції ступеня інтересу (Degree-Of-Interest – *DOI)* розробника ПЗ, яка в загальному випадку може бути визначена як

$$DOI(a_i) = \sum_{j=1}^{N} (e_j^{a_i}) \cdot k_j, \tag{3}$$

де: $a_i \in A$, A – множина артефактів проекту; $e_j \in E$, E – множина подій в процесі розробки; $k_j \in K$, K – множина вагових коефіцієнтів для кожної події (визначається експертним шляхом). Це дозволяє реалізувати у відповідному інструментальному середовищі так званий сфокусований інтерфейс розробника завдань (task-focused developer's interface), тобто в інтерфейсі будуть відображені тільки ті файли проекту, що мають для нього ступень інтересу, вищий за певне значення DOI (яке, в свою чергу, може бути визначена евристичним шляхом). Такий підхід визначає елементи матриці ATM згідно виразу (2), але при цьому не враховує таких досить важливих особливостей гнучких процесів розробки ПЗ: (a) ітеративний характер усього процесу, причому саме із збільшенням числа ітерацій проекту уточняються вимоги користувачів щодо кінцевого програмного продукту; (b) наявність в групі розробників різних за видами діяльності осіб, наприклад: адміністратор баз даних (database admin), розробник коду (code developer), дизайнер інтерфейсу користувача (user interface designer) тощо, які: по-перше, можуть змінювати свої ролі (робочі профілі) і, по-друге, мають різні вагові коефіцієнти відносно ступеню свого інтересу до певних вимог та відповідних артефактів проекту. Тому запропонована модель адаптивної матриці трасування вимог (ADaptive Traceability Matrix – ADTM), значення елементів якої, на відміну від елементів матриці ATM із виразу (2), мають враховувати як саму наявність додаткових

чинників впливу (a) – (b) так і можливість їх постійних змін при виконанні проекту за однією з гнучких методологій.

Для реалізації контура управління (II) (рис. 3) розроблена комплексна експертна процедура, яка дозволяє динамічно (тобто із урахуванням змін у проекті) формувати каталог продукту *PB* на основі кількісних оцінок показників якості текстових специфікацій вимог та визначення їх приоритетності, що б, у кінцевому рахунку, цілелеспрямовано формувати каталог вимог *SB* для кожної наступної ітерації Scrum-проекту.

Задачу 3 пропонується розв'язати шляхом використання достатньо апробованого в сучасній програмній інженерії підходу, а саме: розробки проектних шаблонів (design pattern) для побудови еталонної програмної архітектури (reference software architecture) програмного засобу для адаптивного управління вимогами.

В **третьому розділі** представлені детальні моделі, процедури та проектні рішення, які забезпечують можливість реалізації всіх складових комплексу *задач 1 – 3*.

Алгоритмічна модель (АМ) процесу побудови адаптивної матриці трасування вимог ADTM подана у вигляді кортежу

$$AM = \langle InfBase, ToM, Workflow\,(ADTM)\rangle, \qquad (4)$$

де $InfBase$ – інформаційний базис моделі, ToM (Time-oriented Metric) – час-орієнтована метрика для визначення ступеню інтересу розробника щодо певного проектного артефакту, $Workflow(ADTM)$ – алгоритм побудови матриці ADTM.

Інформаційний базис моделі *AM* визначається кортежом множин

$$InfBase = \langle D, R, F, S, P\rangle, \qquad (5)$$

де $D = \{d_i\}, i = \overline{1, I}, I = |D|$ – множина розробників проекту; $R = \{r_j\}, j = \overline{1, J}, J = |R|$ – множина вимог до ПС; $F = \{f_k\}, k = \overline{1, K}, K = |F|$ – множина файлів або артефактів проекту, $S = \{s_l\}, l = \overline{1, L}, L = |S|$ – множина проектних сесій; $P = \{p_m\}, p = \overline{1, M}, M = |P|$ – множина робочих профілів, або ролей розробників ПС у проекті. Під метрикою *ToM* у виразі (4) мається на увазі саме метрика програмного забезпечення (software metrics), тобто певна міра, що дозволяє отримати чисельні значення деякої властивості програмного продукту або його специфікації. Один із найпростіших, але в той же час досить логічно-обґрунтований, спосіб визначити таку метрику для оцінки питомої ваги елементів матриці ADTM базується на тому, що певні файли (проектні артефакти) вимагають більше часу при роботі над ними при реалізації відповідних вимог до ПС. Така час-орієнтована метрика дозволяє відрізнити ті дії розробників ПС, які вносять лише невеликі («косметичні») проектні зміни від тих, які забезпечують значну програмну доробку. Графічно це ілюструє діаграма, яка наведена на рис. 4.

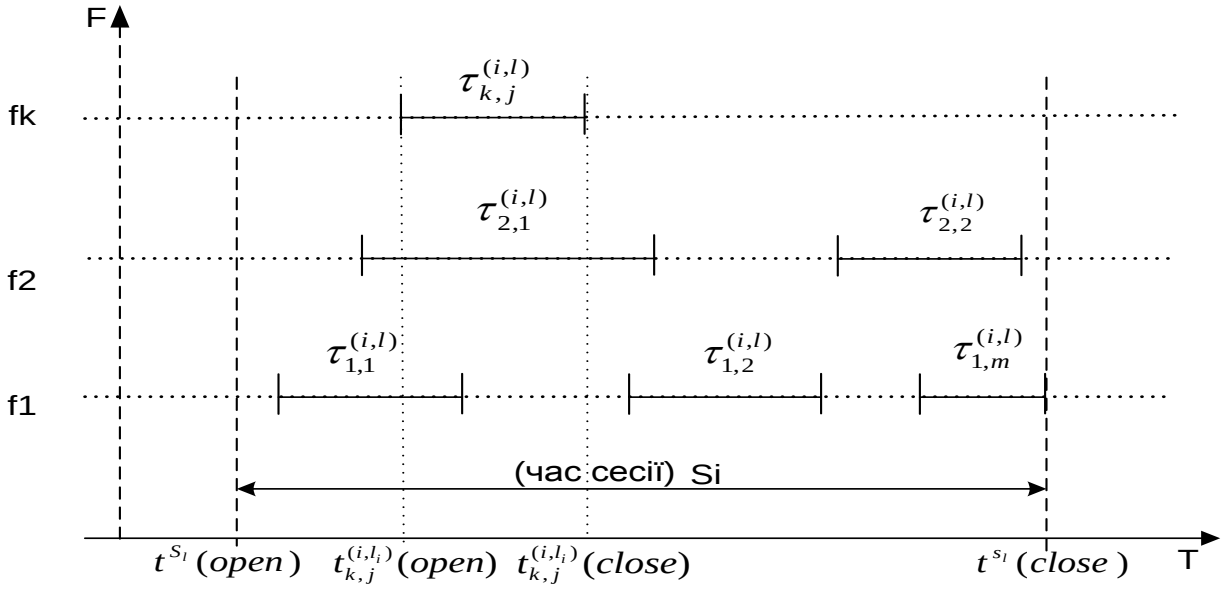

Рис. 4. Час-орієнтована метрика ступеню інтересу (графічна інтерпретація)

Виходячи з діаграми на рис. 4, кількісно ступень інтересу певного розробника проекту до окремого файлу при реалізації відповідної вимоги може бути розраховано з використанням наступного виразу

$$\tau_{k,j}^{(i,l_i)} = \frac{t_{k,j}^{(i,l_i)}(close) - t_{k,j}^{(i,l_i)}(open)}{t^{S_l}(close) - t^{s_l}(open)}, \tag{6}$$

де $\tau_{k,j}^{(i,l_i)}$ – інтервал часу виконання розробником $d_i \in D$ проекту відповідних дій над k-м файлом $f_k \in F$ під час реалізації вимоги $r_j \in R$ протягом сесії $s_l \in S$; $t_{k,j}^{(i,l_i)}(open)$ – момент часу початку роботи над *j*-ою вимогою *i*-м розробником; $t_{k,j}^{(i,l_i)}(close)$ – момент часу для закінчення роботи над *j*-ою вимогою *i*-м розробником; $t^{s_l}(close)$ – час закінчення *li* - проектної сесії; $t^{S_l}(open)$ – час початку *li* - сесії.

Таким чином, метрика *ToM* із виразу (4), в кінцевому рахунку, є множиною значень, отриманих за допомогою формули (6), а саме:

$$ToM = \{\tau_{k,j}^{(i,l_i)}\}, \tag{7}$$

де визначення відповідних індексних змінних відповідає виразу (5). Алгоритм $Workflow(ADTM)$ із виразу (4), який на підставі інформаційного базису (5) із використанням метрики (7) забезпечує побудову матриці ADTM наступним чином:

Крок 1. Для кожного розробника $\forall d_i \in D$, що приймає участь в проекті, будується його локальна матриця $ADTM_{ijk} = \| \tau_{i,j,k,l} \| \cdot \gamma$, де $\gamma = \{\gamma_l\}_{l=1}^{L}$, $\gamma_l = 1/L \quad l = \overline{1,L}$, елементи якої розраховуються за формулою

$$\tau_{j,k}^{(i)} = \sum_{l=1}^{|S|} (\tau_{j,k}^{(i)})_l \cdot (1/|S|) \tag{8}$$

Крок 2. На підставі локальних агрегованих локальних матриць всіх розробників, шляхом осереднення значень їх елементів будується агрегована сумарна матриця $ADTM_{jk} = \| \tau_{i,j,k,l} \| \cdot \lambda$, де $\lambda = \{\lambda_i\}_{i=1}^{I}$ $\lambda = \{\lambda_i\}_{i=1}^{I}$ $i = \overline{1,I}$, коефіцієнти якої обчислюють за формулою

$$\hat{\tau}_{k,j} = \sum_{i=1}^{|D|} \bar{\tau}_{k,j}^{(i)} \cdot (1/|D|). \quad (9)$$

Крок 3. Відповідно до множини можливих робочих профілей учасників проекту $P = \{p_m\}$, із отриманої на попередньому етапі колекції матриць відбираються їх окремі підмножини: $D = D(p_1) \cup D(p_2) \cup ... D(p_M)$, та на підставі цього будуються окремі локальні агреговані матриці для кожного профілю розробника ПЗ

$$ADTM_{k,j,D(p_i)} = \sum_{j=1}^{|D(p_i)|} (ADTM[F,R,D(p_i)])_j / |D(p_i)|). \quad (10)$$

При цьому для спрощення моделі зроблено припущення, що кожний розробник може належить тільки до однієї групи робочих профілей, тобто $D(p_i) \cap D(p_j) = \varnothing$.

Крок 4. На підставі отриманих на попередніх кроках результатів, із урахуванням того, що в процесі розробки ПЗ за технологією Scrum один і той же виконавець проекту може змінювати свої робочі профілі, тобто мати різну ступінь інтересу до певних артефактів, остаточно значення елементів матриці *ADTM* знаходяться за наступною формулою

$$ADTM'_{j,k,D(\bar{p}_i)} = (ADTM_{j,k,D(\bar{p}_i)} \cdot K_{role} + ADTM_{j,k,i}) / (K_{role} + 1). \quad (11)$$

Таким чином, в результаті виконання алгоритму, визначеного за формулами (8)–(11), побудовано адаптивну матрицю трасування вимог *ADTM*, елементи якої обчислюються за формулою (11), що враховує вплив чинників (a)–(b) на особливості опрацювання вимог та відповідних проектних артефактів у гнучкому процесі розробки ПЗ. Це дає змогу запропонувати алгоритм побудови сфокусованого інтерфейсу розробника ПЗ, який відіграє роль зворотного зв'язку у першому контурі управління при виконанні завдань по розробці ПЗ в загальній схемі виконання Scrum-проекту (рис.3). Цей алгоритм у вигляді спрощеної UML-діаграми дій представлено на рис. 5.

Для реалізації другого контуру управління в кібернетичній схемі Scrum-проекту (рис. 3) розроблена комплексна процедура, яка дозволяє динамічно формувати каталог вимог продукту *PB*, для чого послідовно застосовуються: 1) метод латентного семантичного аналізу – для первинної обробки текстових специфікацій вимог і видалення в них дублювання та надлишковості даних; 2) метод аналізу ієрархій – для приоритезації окремих вимог.

Ця технологічна процедура у вигляді структурної діаграми в нотації IDEF0 наведена на рис. 6.

У **четвертому розділі** розроблено інформаційну технологію для реалізації та дослідження адаптивної схеми трасування вимогами в інтегрованому середовищі розробки ПЗ за гнучкою методологією Scrum, наведено результати

обробки статистики програмних експериментів та надано практичні рекомендації щодо впровадження запропонованого підходу.

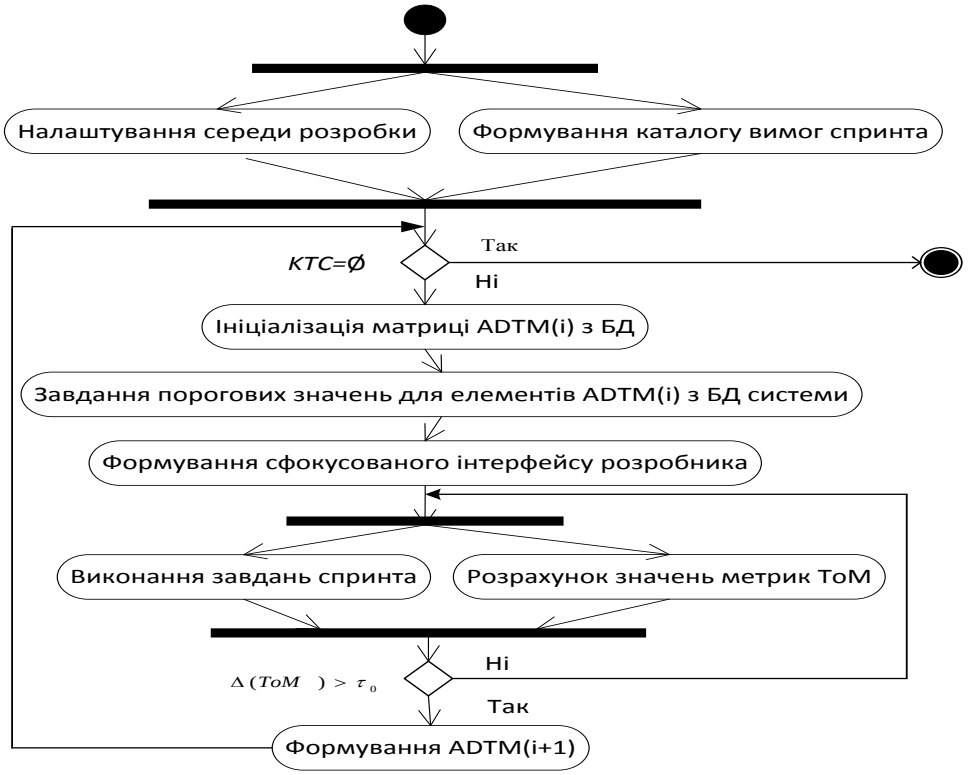

Рис. 5. Алгоритм побудови сфокусованого інтерфейсу розробника ПЗ

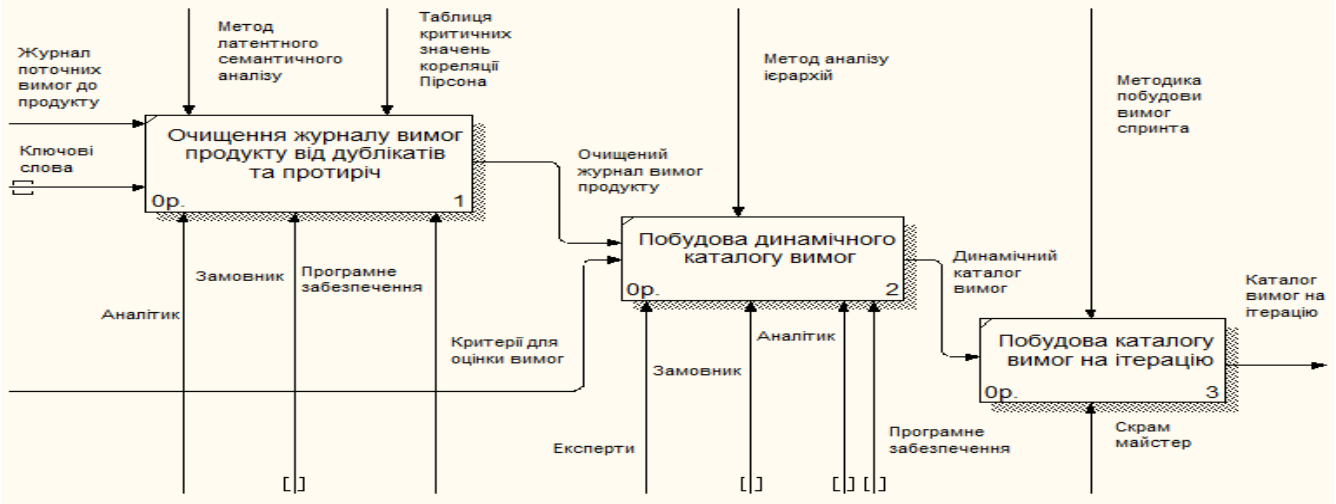

Рис. 6. Логічна схема процедури формування каталогу вимог

Для програмної реалізації технології розроблено прототип інтегрованого CASE-засібу, що поєднує в собі функціональні можливості інструментального середовища *Eclipse IDE*, системи управління вимогами (це компонент *Issue Tracker*) та підсистеми *ReqMIT*, що імплементує запропонований у розділі 3 ал-

горитм побудови матриці ADTM. На рис. 7 показана компонентна програмна архітектура цього CASE-засобу в нотації UML 2.0.

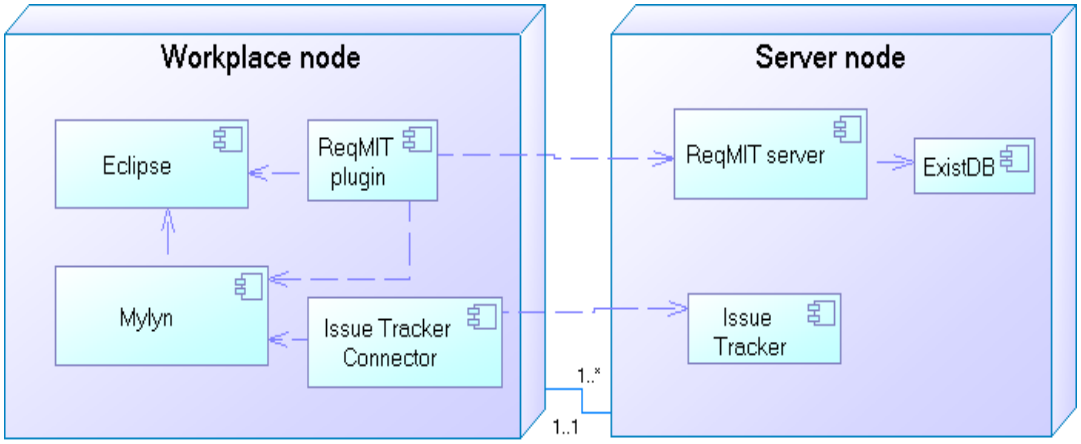

Рис. 7. Компонентна архітектура інтегрованого CASE-засобу

Клієнтській компонент *ReqMIT_plugin* використовується для побудови локальної матриці трасування для кожного розробника проекту, яка потім передається на сервер. Крім того, цей компонент, в залежності від ролі розробника та його проектного інтерфейсу, фокусує останній саме на тих файлах проекту, які є релевантними для роботи з відповідними вимогами. Серверний компонент *ReqMIT_server* використовується для зберігання даних щодо вимог, прив'язки вимог до завдань, а також обчислює ступінь інтересу розробника до відповідних проектних артефактів.

Для комплексної оцінки ефективності розробленого модельно-технологічного інструментарію адаптивного трасування вимог було запропоновано 2-х етапну методику, а саме: 1) оцінка ефективності безпосередньо процесу трасування вимог шляхом побудови матриці ADTM; 2) оцінка можливого підвищення продуктивності праці розробників програмного продукту за рахунок застосування ефекту сфокусованого проектного інтерфейсу із використанням матриці трасування. Для отримання оцінок на першому етапі були використані такі показники як *точність* (precision – *p*) та *повнота* (recall – *r*) інформаційного пошуку. Точність *p* – це відсоток коректно визначених взаємозв`язків між вимогами та артефактами із загальної кількості знайдених, тоді як повнота *r* – це відсоток таких визначених взаємозв`язків із загальної їх кількості, які дійсно в подальшому були імплементовані в проекті. Ці значення обчислюються у наступний спосіб:

$$p = \frac{tp}{tp + fp} \times 100\% \; ; \quad r = \frac{tp}{tp + fn} \times 100\%, \tag{12}$$

де *tp* – істинно позитивні, розглядаються як цікаві і що використовуються; *fp* – хибно позитивні, розглядаються як цікаві, але не використовуються; *fn* – хибно негативні, розглядаються як не цікаві, але використовуються. Середні значення цих показників склали, відповідно, 43% та 61% , що є цілком прийнятним для позитивної оцінки працездатності запропонованого підходу (рис. 8). Для

визначення загальної ефективності інформаційного пошуку звязків вимог та проектних артефактів застосовано збалансовану гармонійну згортку цих параметрів, що визначається за допомогою міри Ван-Різбергена (або F_1 – міри) $F_1 = 2pr \times 1/(p+r) \times 100\%$, і яка також має позитивний тренд, що показано відповідним графіком на рис. 8.

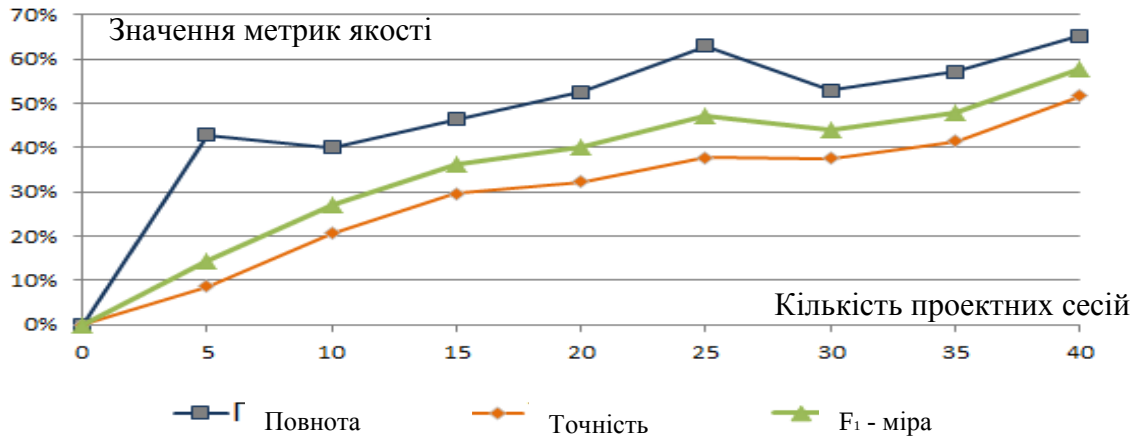

Рис. 8. Результати обчислень показників точності, повноти та F_1 – міри

Для здійснення другого етапу визначення ефективності – отримання оцінки можливого зростання продуктивності праці розробників ПЗ здійснено ряд програмних експериментів, у яких за допомогою засобу ReqMIT у середовищі швидкої розробки Eclipse створювався візуальний ефект сфокусованого інтерфейсу розробника проекту. При цьому, із застосуванням різних порогових значень для оцінки елементів матриці трасування ADTM, відбувалося фільтрування тільки тих проектних файлів, які були релевантними для реалізації відповідних вимог щодо функціональності програмного продукту, і що, в кінцевому рахунку, зменшувало час їх опрацювання. Для кількісної оцінки цього ефекту було використано «*коефіцієнт редагування*» (edit ratio). Це співвідношення (в %) кількості часу, який було витрачено розробником безпосередньо на операції кодування вимог – #(Tedit), до загальної кількості часу, який було витрачено на операції пошуку відповідних файлів – # (Tsearch) у процесі роботи над виконанням певного завдання, тобто

$$edit\ ratio = \frac{\#(Tedit)}{\#(Tsearch)} \times 100\%. \qquad (13)$$

Результати цих експериментів довели, що застосування сфокусованого інтерфейсу розробника проекту із пороговими значеннями у діапазоні від 0,1 до 0,4 забезпечують зростання коефіцієнту редагування, визначеного за формулою (13), у межах від 4,69 % до 15,37 % відповідно, у порівнянні із процесом опрацювання вимог без застосування запропонованої адаптивної технології їх трасування.

У **додатках** до дисертаційної роботи наведені документи про використання отриманих результатів досліджень, приклади проектної документації та сти-

сла інструкція по роботі із CASE-системою *ReqMIT*, а також детальні дані проведених програмних експериментів.

ВИСНОВКИ

У дисертаційній роботі вирішена науково-практична задача розробки комплексу моделей та прикладної інформаційної технології для адаптивного трасування вимог в гнучких процесах розробки програмного забезпечення (ПЗ), зокрема, із використанням методології Scrum

В процесі виконання роботи отримано наступні обґрунтовані наукові та практичні результати.

1. Проведено аналіз процесів накопичення та обробки вимог до програмного забезпечення у гнучких процесах його розробки, особливу увагу при цьому приділено вирішенню задачі підвищення ефективності трасування вимог. Запропоновано класифікацію відповідних підходів, що дозволяє в подальшому коректно ставити та розв'язувати задачі створення адаптивного механізму трасування вимог в процесах розробки ПЗ за методологією Scrum.

2. Із застосуванням кібернетичних принципів формалізовано організаційну схему виконання типового Scrum-процесу розробки ПЗ, що дозволяє визначити в ньому два нових технологічних контури управління вимогами, що дозволяє як забезпечити безпосередньо трасування вимог при розробці поточних проектних завдань щодо розробки програмного продукту, так і проводити попереднє їх опрацювання перед кожною проектною ітерацією.

3. Для забезпечення функціонування першого контуру управління розроблена знання-орієнтована математична модель адаптивного процесу трасування вимог, яка на відміну від існуючих підходів, використовує механізм побудови сфокусованого проектного інтерфейсу розробника ПЗ на основі застосування функції ступеню інтересу, що дозволяє скоротити витрати часу та підвищити точність опрацювання вимог при їх імплементації у відповідні проектні артефакти.

4. Для другого контуру управління запропонована комплексна експертна процедура, яка дозволяє динамічно, тобто із урахуванням змін у проекті, формувати каталог вимог щодо програмного продукту на основі кількісних оцінок показників якості обробки текстових специфікацій вимог та визначати їх приоритетність, щоб, у кінцевому рахунку, цілелеспрямовано формувати каталог вимог для кожної наступної ітерації Scrum-проекту.

5. Розроблені шаблони (патерни) проектування, що дозволили програмно реалізувати прототип інструментального засобу (CASE-засобу) із вбудованим адаптивним механізмом трасування вимог, який поєднує в собі функціональні можливості типових систем управління вимогами та інтегрованих середовищ розробки ПЗ, зокрема, відкритої платформи Eclipse.

6. Розроблено прикладну інформаційну технологію та методику оцінки ефективності застосування запропонованого механізму трасування вимог за критеріями повноти та точності їх опрацювання, проведено експериментальне дослідження ефективності цього підходу та сформульовані практичні рекомен-

дації щодо його впровадження в реальні проекти по розробці та супроводу ПЗ із застосуванням гнучких методологій.

7. Одержані в роботі теоретичні та практичні результати використані при виконанні держбюджетних тем у НТУ «ХПІ», при виконанні проектів в компанії Bit media e-learning solution GmbH & Co KG (Грац, Австрія), а також запроваджені в навчальному процесі кафедри АСУ НТУ «ХПІ» при викладанні дисциплін «Основи проектування ПЗ», «Аналіз вимог до ПЗ», «Методи та засоби автоматизації процесів життєвого циклу ПЗ».

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Гамзаев Р.А**.** Нечеткий подход к решению задачи анализа системных требований в процессах разработки и сопровождения информационных систем / Н.В. Ткачук, Р.А.Гамзаев // Автоматизированные системы управления и приборы автоматики. – Харьков: ХНУРЭ. – 2005. – Вып. 134. – С. 64 -70.

Здобувач розробив процедуру визначення нечітких значень для оцінки стану вимог у багатовимірному просторі відповідних критеріїв.

2. Гамзаев Р.А. Комплекс моделей технологии прототипирования для адаптивного проектирования компонентных программных решений / А.А Земляной, Н.В. Ткачук, Р.А. Гамзаев // Вісник Національного технічного університету «ХПІ» – Харків: НТУ «ХПІ». – 2008. – № 5. – С.97-107.

Здобувачем проаналізовані можливості застосування різних технологій прототипування для валідації функціональних вимог до ПЗ.

3. Гамзаев Р.А. Особенности процессов управления требованиями в гибких методологиях разработки программного обеспечения / Н.В. Ткачук, Р.А. Гамзаев // Радиоелектронні і комп'ютерні системи. – Харків: Національний аерокосмічний університет ім. М.Є. Жуковського «Харківський авіаційний інститут». – 2010. – № 4 (45) – С. 200 - 204.

Здобувач формалізував схему трасування вимог в контексті методології Scrum для гнучкої розробки ПЗ.

4. Gamzayev R. O. Models and Tools for Effectiveness Increasing of Requirements Traceability in Agile Software Development / Tkachuk M. V., R.O.Gamzayev,H.C.Mayr, V.O. Bolshutkin // Проблемы программирования. – К.: НАН України. – 2012. – № 2–3 (спец. выпуск). – С.252 – 260.

Здобувачем досліджені можливості використання функції ступеню інтересу для трасування вимог та інформаційна технологія для її реалізації.

5. Гамзаев Р.О. Модель та інформаційна технологія побудови адаптивної матриці трасування вимог в гнучких процесах розробки програмного забезпечення / Р.О.Гамзаев, М.В.Ткачук // Вісник НТУ «ХПІ». – Харків, 2013 – № 2 (976). – С. 49 – 60.

Здобувач розробив знання-орієнтовану модель адаптивного процесу трасування вимог.

6. Gamzayev R.A. Towards Prototyping-based Technology for Adaptive Software Development / M.V. Tkachuk, A.A. Zemlyano, R.A. Gamzayev //

R. Kashek et. al. (Eds.): UNISCON 2008, LNBIP 5: Springer-Verlag Berlin Heidelberg, 2008. – pp. 508-518.

Здобувачем виконан аналітичний огляд підходів та інструментальних засобів прототипування програмних рішень з метою аналізу вимог щодо ПЗ.

7. Gamsaev R. An Approach to Knowledge-based Data Handling in Complex Process Control Systems / R.Gamzayev,D.Kuklenko,M. Tkachuk. // Lecture Notes in Informatics (LNI) Proceedings, Series of the German Informatics Society (GI),Volume P-63. Printed in Bonn. – 2005. – p.p. 61-72.

Здобувачем розроблена модель асоціативних правил та програмна архітектура системи інтелектуальної обробки даних.

8. Гамзаев Р.А. Концепція та архітектурні рішення для інтеграції систем управління вимогами та засобів підтримки процесу розробки програмного забезпечення / М.В.Ткачук, Р.А.Гамзаев, В.О.Большуткін // Матеріали XIX Міжн. науково-практ. конференції «Інформаційні технології: наука, техніка, технологія, освіта, здоров'я», НТУ «ХПІ», Харків, 1–3 червня 2011. – с. 28.

Здобувачем запропоновані патерни проектування для інтеграції систем управління вимогами та середовищ швидкої розробки ПЗ.

9. Гамзаев Р.А. Процедура построения динамического каталога требований при разработке программного обеспечения по методологии Scrum / М.В. Ткачук, М.Г. Гнатенко, Р.А. Гамзаев // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: Тези доповідей XX міжнародної науково-практичної конференції, Ч.І (15–17 травня 2012р., Харків) / за ред. проф. Товажнянського Л.Л. – Харків, НТУ «ХПІ». – С.30.

Здобувачем запропонована комплексна процедура формування динамічного каталогу вимог для проектної ітерації із урахуванням їх приоритезации та непротиріччя.

АНОТАЦІЇ

Гамзаєв Р.А. Моделі та інформаційна технологія трасування вимог в гнучких процесах розробки програмного забезпечення. – На правах рукопису.

Дисертація на здобуття вченого ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Національний технічний університет «Харківський політехнічний інститут», Харків. – 2013.

У роботі розвязано задачу створення моделей та інформаційної технології для адаптивного трасування вимог в гнучких процесах розробки програмного забезпечення (ПЗ) на прикладі методології Scrum. Праналізовані основні існуючи підходи до трасування вимог, запропонована їх класифікація та надано узагальнюючий формалізований опис таких моделей із використанням фреймових специфікацій. Формалізовано організаційну схему виконання типового Scrum-проекту, в якій визначено два нових технологічних контури управління: 1) контур управління трасуванням вимог у процесі виконання поточних завдань щодо розробки ПЗ, 2) контур управління процедурами попереднього

опрацювання текстових специфікацій вимог.

Розроблені знання-орієнтована модель адаптивного трасування вимог на основі функції ступеню інтересу та експертна процедура формування динамічного каталогу вимог, із урахуванням змін у проекті, на основі кількісних оцінок показників якості текстових специфікацій та визначення їх приоритетності. Реалізовано прототип програмного CASE-засобу розробки ПЗ із вбудованим адаптивним механізмом трасування вимог. Запропоновано та апробовано комплексну методику для експериментальної оцінки його повноти та точності, а також для оцінки ефективності застосування сфокусованого інтерфейсу розробника програмних завдань.

Ключові слова: знання-орієнтована модель, інформаційна технологія, програмне забезпечення, трасування вимог, функція ступеню інтересу, сфокусований інтерфейс розробника.

Гамзаев Р.А. Модели и информационная технология трассировки требований в гибких процессах разработки программного обеспечения. – На правах рукописи.

Диссертация на соискание ученой степени кандидата технических наук по специальности 05.13.06 – информационные технологии. – Национальный технический университет «Харьковский политехнический институт», Харьков, 2013.

В работе решена задача создания моделей и информационной технологии для адаптивного управления требованиями в гибких процессах разработки программного обеспечення (ПО) на примере методологии Scrum. Проанализированы основные существующие подходы к трассировки требований, предложена их классификация и представлено обобщающее формализованное описание таких моделей с использованием фреймовых спецификаций. Формализована организационная схема выполнения типового Scrum-проекта, в которой определены два новых технологических контура управления:1) контур управления трассировкой требований в процессе выполнения текущих задач по разработке ПО, 2) контур управления процедурами предварительной обработки текстовых спецификаций требований.

Разработаны знание-ориентированная модель адаптивной трассировки требований на основе функции степени интереса и экспертная процедура формирования динамического каталога требований, с учетом изменений в проекте, на основе количественных оценок показателей качества текстовых спецификаций и определения их приоритетности. Реализован прототип программного CASE-средства разработки ПО со встроенным адаптивным механизмом трассировки требований. Предложена и апробирована комплексная методика для экспериментальной оценки его полноты и точности, а также для оценки эффективности использования сфокусированного интерфейса разработчика программных заданий.

Ключевые слова: знание-ориентированная модель, информационная технология, программное обеспечение, трассировка требований, функция степени интереса, сфокусированный интерфейс разработчика.

Gamzayev R.A. Models and information technology for requirements traceability in agile-software development. - Manuscript.

Dissertation for the candidate's degree on the speciality 05.13.06 − information technologies. − National Technical University «Kharkiv Polytechnic Institute», Kharkiv, 2013.

The thesis is devoted to solve the problem to develop models and information technology for adaptive requirements management in agile-software development, in particular, on the example of Scrum-methodology.

In the *fisrt chapter* the main tasks of requirements management in agile-development are considered, a special attention is paid to the problem of requirements traceabiluty in software development process. The classification of existing appoaches to this issue is elaborated, which includes such techniques as tracability matrix, associate rules and traceability patterns, their advantages and shortcommings are considered. The contradiction existing in most part of agile-approach is mentioned, namely: although the positive effect of their usage, in general, is ensured basically by eliminating of needs to have a large amount of project documentation (including detailed specifications for user requirements), exactly this circumstance complicates a requerement traceability as an important factor to provide the quality of target software product.

In the *second chapter* the methodological background for research objectives is elaborated basing on a cybernatics approach which becomes a growing trend in modern software engineering. With respect to this approach and taking into account the results of performed analysis on requirements management issues the following list of complex tasks to be resolved is formulated:

Task 1. Formalization, development and investigation of requirements management issues in agile-software development, especially in Scrum-methodology, in order to identify their quantitative parameters and to enable the usage of feedback control loops within such process.

Task 2. Elaboration of mechanisms to realize an adaptive requirements traceability using intelligent data processing and advanced software tools for requirements traceability.

Task 3. Development and testing of information technology to support the proposed approach with respect to its quality attributes.

To solve these tasks comprehensively, an organizational scheme of Scrum-project is formalized that allows to define within its framework two new technological control loops for requirements management. To implement the first control loop is proposed to elaborad a knowledge-oriented model for ADaptive Traceability Matrix (ADTM), which should provide for software developer a task-focused project interface basing on the degree-of-interest (DOI) function. For the second loop an expert decision-making procedure has to be used that allows to build

a requirements catalog in Scrum-project dynamically, taking into account current project's changes, and finally to form a subset of requirements for each next project iteration (or for each sprint in terms of Scrum-methodology).

In the *third chapter* the detailed models and appropriate software design solutions for the proposed approach are presented. To manage a process of an ADTM-building the formalized model is elaborated which is given as a tuple of three components: 1) a structured information base, 2) a time-oriented metrics, and 3) an algorithm to calculate DOI-function values, basing on data related to requirements and projects artifacts. To implement an expert procedure for constructing a dynamic requirements catalog in the second control loop the following methods are used consistently*:* 1) a latent semantic analysis method for primary processing of requirements textual specifications to eliminate their possible duplicates and data redundancy; 2) a analytic hierarchical method for requirements prioritizing in this catalog in order to form a subset of selected requirements for each project iteration. To support the proposed approach with appropriate CASE-tool the design patterns for its architecture are elaborated which allow to combine functionality of requirements management systems with integrated development environment (e.g. IDE Eclipse).

In the *forth chapter* the informational technology and the method to assess the serviceability of the developed requirements traceability mechanism are presented and discussed. The appropriate software is implemented as client-server application, where the client-side components are responsible for data gathering in order to form a local ADTM of each developer, basing on his / her individual time-oriented metrics, and to transfer this data to the server-side component. This one forms a global ADTM, calculates integrated values of DOI-functions, which are sent back to a client-side components and which are finally used to provide task-focused project's interface for a concerned developer.

The testing of the proposed approach was performed in 2 phases: 1) to estimate a correctness of adaptive traceability matrix, 2) to evaluate how to increase a developer's productivity by using task-focused interface. At the first test-phase the quality parameters of information retrieval – accuracy and completeness, – were used, and they have got values 43% and 61% accordingly, besides that to estimate a common effectiveness of this information retrieval the F-measure (or Van Rijsbergen's measure) was used, and it has shown a positive results. At the second test-phase, to evaluate an increasing of software productivity due to applying task-focused developer's interface, the *edit coefficient* was calculated. It is equal to the ratio of the amount of editing time (time for coding) versus amount of time spent by developer for files searching. Depend on the different thresholds used in task-focused interface, the edit coefficient was grown in the range between 4,69 % and 15,37 % respectively.

Key words: knowledge-oriented model, information technology, software, requirements traceability, degree of interest function, developer's task-focused interface.