

Д. ГОБОВ, Н. ШЕВЧЕНКО

ВИЗНАЧЕННЯ АРХІТЕКТУРИ ВИМОГ ДО ІТ-РІШЕННЯ ЯК БІЗНЕС-АНАЛІТИЧНОГО ПРОДУКТУ

Предметом дослідження в статті є підходи до розуміння та тлумачення поняття архітектури вимог до ІТ-рішення. **Мета роботи** – надати визначення архітектурі вимог до ІТ-рішення для кращого розуміння й подальшого впровадження цього концепту в практику розроблення програмного забезпечення. У статті передбачається виконання таких **завдань**: проаналізувати підходи до класифікації вимог з метою встановлення спільних та відмінних характеристик і їх впливу на визначення архітектури вимог; проаналізувати стандарти з інженерії вимог для пошуку визначення та будь-яких зав'язків з архітектурою вимог; розглянути підходи до побудови набору вимог та використання шаблонів для їх опису з метою виявлення оптимальних стратегій для бізнес-аналітичних процесів та їх впливу на архітектуру вимог. Використовуються такі **методи**: компаративний аналіз, зокрема порівняння та узагальнення різних класифікацій вимог з огляду на їх вплив на визначення архітектури вимог; аналіз та узагальнення підходів до інженерії вимог для виявлення їх впливу на процес формування архітектури вимог; системний аналіз понять «вимога» та «архітектура вимог»; метод порівняння та аналогій у вивченні питання використання шаблонів для опису вимог у контексті побудови архітектури вимог. Крім того, впроваджено метод аналогового моделювання для виявлення зав'язків між якістю вимог та їх архітектурою, який дав змогу визначити оптимальні підходи до побудови архітектури вимог, що відповідають вимогам якості програмного забезпечення. **Досягнуті результати**: досліджено підходи до розуміння вимог у процесі розроблення програмного забезпечення; наведено загальноприйнятні класифікації вимог та характеристики якісних вимог; проаналізовано зв'язки ланцюжка "вимоги – якість вимог – підходи до опису вимог – стандарти – архітектура вимог" з подальшою логічною проекцією вивченого контексту на архітектуру вимог; запропоновано авторське визначення поняття "архітектура вимог". **Висновки**. Архітектура вимог може базуватися на різних архітектурних фреймворках, залежить від методологій розроблення програмного забезпечення, але водночас вона може варіюватися відповідно до галузевих стандартів і потреб конкретного проєкту, також архітектура вимог може використовувати певні шаблони документів, проте не обмежується ними.

Ключові слова: бізнес-аналіз; вимоги до програмного забезпечення; документування вимог; подання вимог; точка зору на структуру вимог; архітектура вимог.

Вступ. Визначення проблеми

Бізнес-аналіз – це діяльність, що забезпечує можливість змін підприємства внаслідок визначення потреб і рекомендацій рішень, які приносять користь зацікавленим сторонам. Бізнес-аналіз допомагає організації формулювати потреби та обґрунтовувати зміни, а також проєктувати та описувати корисні рішення. Крім того, бізнес-аналіз – це наукова дисципліна з чітко визначеними принципами, концепціями та практиками, що в сукупності допомагають фахівцям із бізнес-аналізу та організаціям створювати бажані результати [1]. Ланкою-посередником між формулюванням потреби та обґрунтуванням змін є набір завдань щодо роботи з вимогами: виявлення, аналіз, валідація та управління.

У сфері інформаційних технологій бізнес-аналіз є ключовою діяльністю, спрямованою на досягнення змін і вдосконалення підприємства через ідентифікацію потреб і надання рекомендацій, зокрема щодо

розроблення, модифікації, упровадження програмного забезпечення. Якісні продукти бізнес-аналізу (артефакти – будь-які об'єкти, що стосуються рішення та створюються в межах зусиль бізнес-аналізу) допомагають підприємствам обґрунтовувати зміни та впроваджувати ефективні рішення. Якісні вимоги є критерієм успішності продукту та визначають його спроможність приносити користь зацікавленим сторонам. Неякісні вимоги можуть призвести до серйозних наслідків, наприклад, позначитися на бюджеті, термінах виконання проєкту й на рівні задоволення стейкхолдерів.

Вивченню критеріїв якості вимог до програмного забезпечення присвячено чимало наукових праць і практичних дослідів, а вимоги до якості визначені багатьма спеціалізованими стандартами (наприклад, *ISO29148* та *IREB*) [2, 3]. Однак, як зазначається в роботі [4], навіть за умови масштабного вивчення підходів до визначення якості вимог, питання критеріїв якості залишається відкритим, особливо

з погляду фахівців, які безпосередньо працюють з вимогами. У статті [4] одним із напрямів покращення якості вимог визнано використання шаблонів і контрольних показників якості, проведення ретроспективи вимог.

Відповідно до ВАВОК [5] "вимога" – це придатне для використання подання потреби, і це "подання" починає своє існування як результат діяльності бізнес-аналітика вже на стадії виявлення або збору інформації. Вимоги зосереджені на розумінні того, яка цінність може бути отримана внаслідок її виконання. "Дизайн" – придатне для використання подання рішення, що визначає, як саме рішення зможе приносити користь зацікавленим сторонам. В ІТ-проектах цей термін використовується в контексті технологій та передбачає визначення того, як технології будуть використовуватись для виконання завдань бізнесу. Дизайн може містити дизайн користувацьких інтерфейсів, архітектурні моделі, моделі бізнес-процесів. Для чіткого розмежування документів, що створюють бізнес-аналітики та фахівці в галузі рішень (архітектори, розробники), замість терміна "дизайн" часто застосовують словосполучення "вимоги до рішення". За змістом, це подання може бути документом (або набором документів) і здатне значно змінюватися залежно від обставин.

Робота з аналізу вимог та визначення дизайну рішення передбачає:

- специфікацію та моделювання вимог для докладного опису набору вимог чи дизайну з використанням аналітичних методів;

- верифікацію вимог для перевірки того, що набір вимог або дизайну опрацьовано досить детально для використання конкретною зацікавленою стороною, внутрішньо несуперечливий і має високу якість;

- валідацію вимог для перевірки того, що набір вимог або дизайну корисний для бізнесу й підтримує цілі та завдання організації;

- визначення архітектури вимог для структурування всіх вимог і дизайну так, щоб вони підтримували кінцеву бізнес-мету зміни та успішно працювали як єдине ціле;

- визначення варіантів вирішення для виявлення, вивчення та опису різних можливих шляхів задоволення потреб бізнесу;

- аналіз потенційної цінності та рекомендацію рішення, що передбачає оцінювання бізнес-цінності, пов'язаної з потенційним рішенням, і порівняння

різних варіантів, зокрема компроміси, для визначення та рекомендації варіанта рішення, що приносить найбільшу загальну користь.

За кожним із перелічених завдань закріплюється певний результат, на досягнення якого спрямована діяльність бізнес-аналітика: (1) визначені та змодельовані вимоги, (2) верифіковані вимоги, (3) валідовані вимоги, (4) архітектура вимог, (5) варіанти дизайну, (6) рекомендації рішення. У цьому разі, з огляду на змістовне подання вимог, результатом, що формалізує, візуалізує та підсумовує результати (1)–(3) та (5), (6), є саме архітектура вимог. Архітектура вимог використовується також і в інших бізнес-аналітичних завданнях: пріоритезація вимог, оцінювання їх змін, виокремлення та моделювання вимог, визначення варіантів дизайну [6], а також у процесі комунікації бізнес-аналітичної інформації, підтримки вимог в актуальному стані, трасуванні, затвердженні вимог тощо. Повний набір вимог та зв'язків між ними (архітектура вимог) використовується у визначенні варіантів дизайну, що здатні задовольнити весь набір вимог [1]. Вимоги зосереджуються на потребах, а дизайн – на рішенні. Різниця між вимогами та дизайном не завжди очевидна, бо для виявлення, моделювання та аналізу вимог і дизайну впроваджуються одні й ті самі техніки.

На архітектуру вимог автори ВАВОК звертають увагу в третій версії стандарту, виокремлюючи активність із визначення архітектури в окреме завдання. Також у глосарії третьої версії стандарту сформульовано відповідне визначення [5]: архітектура вимог – "вимоги щодо ініціативи та взаємозв'язки між ними". Це структура всіх вимог, метою якої є сумісне забезпечення вимогами досягнення цілей. Архітектура вимог поєднує окремі моделі та специфікації для гарантії того, що всі вимоги є єдиною системою, спрямованою на досягнення бізнес-мети та отримання цінності зацікавленими сторонами. Автори ВАВОК другої версії для узгодження поглядів різних зацікавлених сторін пропонували формувати документ із вимогами (*Requirements Document, Requirements Package*), який визначався як набір вимог, згрупований у документ чи презентацію для взаємодії із стейкхолдерами. У ВАВОК третьої версії *Requirements Package* також визначається як артефакт бізнес-аналізу. Водночас архітектура вимог, зважаючи на визначення, призначена для того, щоб показати, як елементи співпрацюють разом для підтримки бізнес-вимог, і структурувати їх для узгодження різних поглядів

зацікавлених сторін. Тут постає очевидне важливе питання, яке автори статті пропонують розглянути: чи є архітектура вимог альтернативою, частиною або містить пакет вимог (*Requirements Package*), а якщо ні, то якого значення архітектура набуває в контексті вимог до програмного забезпечення.

Аналіз останніх досліджень і публікацій

Глосарій BABOK надає широке визначення архітектури вимог (*requirements architecture*), уникаючи термінів "структура", "система", "документ" та будь-які обмеження щодо "вимог", які б додали змістовності цьому поняттю та уможливили його

використання як трафарету до архітектури вимог до програмного забезпечення. У праці [7] зазначено, що "архітектура вимог передбачає визначення того, як будуть згруповані та пов'язані вимоги й набори вимог до програмного забезпечення". У наведеному визначенні варто звернути увагу на термін "набір вимог", що містить не лише вимоги та зав'язки між ними, але й певні об'єкти.

Як бачимо, контекст уточнює термін, додаючи певні обмеження. З цього аспекту важливо зрозуміти спочатку контекст, у якому сформулюються вимоги, а вже потім визначатися з тим, що таке архітектура цих вимог, зважаючи, що вибір фреймворку / методології безпосередньо впливає на її формування (рис. 1).

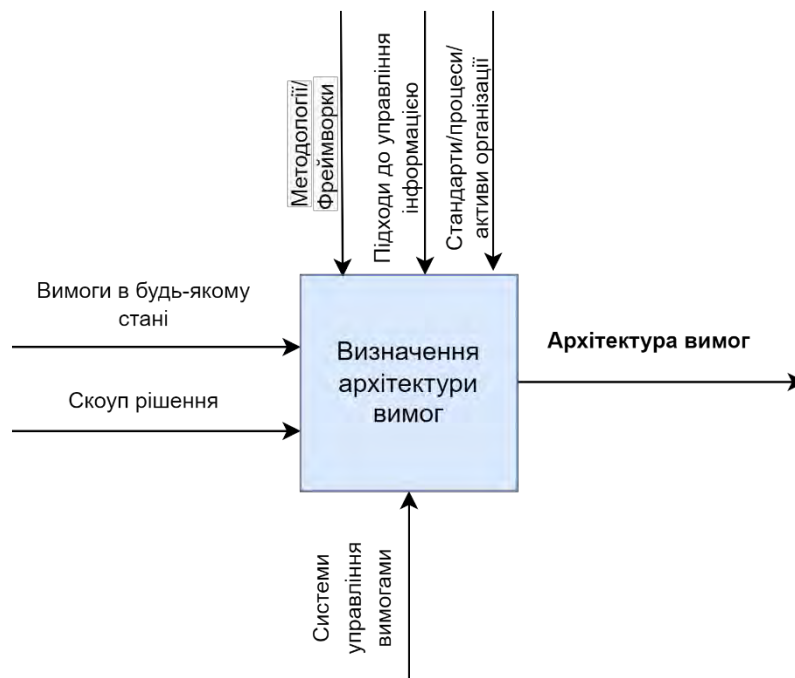


Рис. 1. Контекстна діаграма "Визначення архітектури вимог" [1]

Більшість досліджень, спрямованих на вивчення сучасних практик роботи бізнес-аналітика під час документування вимог, зосереджені на підходах та техніках опису вимог і елементів дизайну, і не розглядають їх у сукупності як пов'язану структуру. У праці [8] досліджено поточні практики в українській IT-індустрії щодо використання технік аналізу й моделювання вимог і впливу на них контексту проекту. У межах дослідження виявлено найбільш популярні техніки бізнес-аналізу, призначені для створення елементів архітектури вимог – діаграми класів, діаграми бізнес-процесів, сценарії використання. У роботі [9] проаналізовано сумісне

використання цих моделей, але встановлення зв'язків між ними не вивчалось. Результати дослідження щодо змісту документів "Специфікація вимог до програмного забезпечення" в польській IT-індустрії подані в статті [10]. У цьому дослідженні підтверджено, що зміст специфікацій вимог залежить від предметної галузі проекту, методу фінансування, обраної методології, а найчастішим елементом у документах є опис вимог щодо інтеграції із зовнішніми системами. Також вивчалися підходи до трасування вимог, що дає змогу відстежувати взаємні залежності між вимогами, результатами виявлення, тест-кейсами, архітектурними рішеннями

та кодом. У роботі [11] наведені результати систематичного огляду досліджень, присвячених трасуванню вимог на джерела їх походження (протоколи робочих зустрічей, результати інтерв'ю та аналізу наявних систем). За результатами дослідження виявлено, що найбільш критичними проблемами, які виникають під час трасування вимог, є неадекватна підтримка версій посилань трасування, погана адаптованість для конкретних потреб проєкту та відсутність довіри до можливо неправильно створених посилань. Останні дві проблеми безпосередньо пов'язані з інтеграцією трасування в архітектуру вимог. У роботі [12] досліджено питання взаємозв'язків між документами вимог і сформульовано концептуальну модель *T-Reqs*, що допомагає описати відношення різних типів між документами. Автори наголошують, що їхня модель є здебільшого теоретичною та її перевірка в межах проєктної діяльності є наступним етапом розвитку.

Визначення не вирішених раніше частин загальної проблеми. Мета роботи, завдання

Вимоги є вхідною інформацією для визначення архітектури, але водночас існує зворотний зв'язок: архітектура вимог впливає на визначення та моделювання вимог. Тому доцільно в межах цієї статті дослідити підходи до розуміння вимог у процесі розроблення програмного забезпечення, загальноприйнятій класифікації вимог і характеристики якісних вимог із подальшою логічною проєкцією вивченого контексту на архітектуру вимог та запропонувати визначення поняття "архітектура вимог".

Мета дослідження – надати визначення архітектурі вимог до ІТ-рішення для кращого розуміння та подальшого впровадження цього концепту в практику розроблення програмного забезпечення.

Завдання дослідження: проаналізувати підходи до класифікації вимог з метою виокремлення спільних і відмінних характеристик та їх впливу на розуміння архітектури вимог; проаналізувати стандарти з інженерії вимог для пошуку визначення та будь-яких зв'язків з архітектурою вимог; розглянути підходи до побудови набору вимог і використання шаблонів для їх опису з метою виявлення оптимальних стратегій для бізнес-аналітичних процесів та їх впливу на архітектуру вимог.

Матеріали й методи

Під час пошуку та надання власного визначення архітектурі вимог до ІТ-рішення використані методи компаративного аналізу, зокрема порівняння та узагальнення різних класифікацій вимог з огляду на їх вплив на визначення архітектури вимог, методи аналізу та узагальнення підходів до інженерії вимог для виявлення їх впливу на процес формування архітектури вимог, методи системного аналізу понять "вимога" та "архітектура вимог", а також методи порівняння та аналогій у вивченні питання використання шаблонів для опису вимог у контексті побудови архітектури вимог. Крім того, використаний метод аналогового моделювання для виявлення зв'язків між якістю вимог та їх архітектурою, що дав змогу визначити оптимальні підходи до побудови архітектури вимог, які відповідають вимогам якості програмного забезпечення.

Результати досліджень та їх обговорення

Вимога – це твердження, що перекладає або виражає потребу й пов'язані з нею обмеження та умови [2]. К. Вігерс у роботі [13] дає більш повне визначення цього терміна: "Вимоги – це специфікація того, що має бути реалізовано. У них описана поведінка системи, її властивості чи атрибути, що можуть використовуватися як обмеження під час розроблення системи". Вимоги декомпонують та уточнюють, трансформуючи бізнес-вимоги у функціональні та нефункціональні.

У табл. 1 наведена класифікація вимог за К. Вігерсом [13] та сукупністю знань з бізнес-аналізу BABOK [1], PMI [14], IREB [3].

Отже, вимоги до програмного забезпечення містять три рівні: бізнес-вимоги, користувацькі та функціональні й нефункціональні вимоги. На кожному рівні вимоги документуються, але документ – це необов'язково паперовий чи електронний документ; це контейнер, де зберігаються знання про вимоги [13] (рис. 2).

К. Вігерс використовує конкретні назви артефактів для документації вимог, але водночас наголошує, що назви в таких контейнерах можуть бути різними, важливо – це досягти домовленості, які види інформації міститимуться в них та як ця інформація буде організована.

Таблиця 1. Класифікація вимог

PMI (у частині вимог до продукту)	BAVOK	IREB (у частині вимог до продукту)	К. Вігерс
<u>Бізнес-вимога</u> (визначення відсутнє).	<u>Бізнес-вимога</u> – формулювання цілей, завдань і результатів, що описують, чому зміна була ініційована.	<u>Бізнес-вимога</u> – вимога, основана на бізнес-меті, завданнях або потребах, що мають бути досягнуті за допомогою використання системи або комплексу систем.	<u>Бізнес-вимога</u> – високорівнева бізнес-мета підприємства чи замовника.
<u>Вимоги зацікавленої сторони</u> (визначення відсутнє).	<u>Вимоги зацікавленої сторони</u> – описують потреби зацікавлених сторін, які необхідно задовольнити, щоб виконати бізнес-вимоги.	–	<u>Вимоги зацікавленої сторони</u> – завдання, які певні класи користувачів мають змогу вирішувати в системі, або певні атрибути продукту.
–	<u>Системні вимоги</u> передбачають функціональні та нефункціональні вимоги.	–	<u>Системні вимоги</u> – вимоги верхнього рівня до продукту, що містить багато підсистем (ПЗ) чи ПЗ та обладнання.
–	<u>Функціональні вимоги</u> – описують можливість, що має рішення з погляду поведінки та інформації, з якою рішення працюватиме.	<u>Функціональні вимоги</u> – опис результату або поведінки, що мають бути реалізовані функціями системи.	<u>Функціональні вимоги</u> – опис поведінки системи, що необхідна в певних умовах.
<u>Вимоги до якості</u> (визначення відсутнє).	<u>Нефункціональні вимоги</u> , чи вимоги до якості сервісу – описують умови, за яких рішення має залишатися функціональним, або якості, необхідні для рішення.	<u>Вимоги до якості (нефункціональні обмеження)</u> – вимоги, що описують обмеження / проблеми, не охоплені функціональними вимогами.	<u>Нефункціональні вимоги</u> – опис властивостей чи особливостей, необхідних для системи, або обмеження, яким має відповідати (виконувати) система. <u>Атрибути якості</u> – вид нефункціональних вимог, що описують характеристику сервісу або продуктивності продукту.
–	<u>Перехідні вимоги</u> (вимоги перехідного періоду) – описують можливості, необхідні для рішення, або умови, яким воно має відповідати, щоб забезпечити перехід із поточного стану в майбутній.	–	–

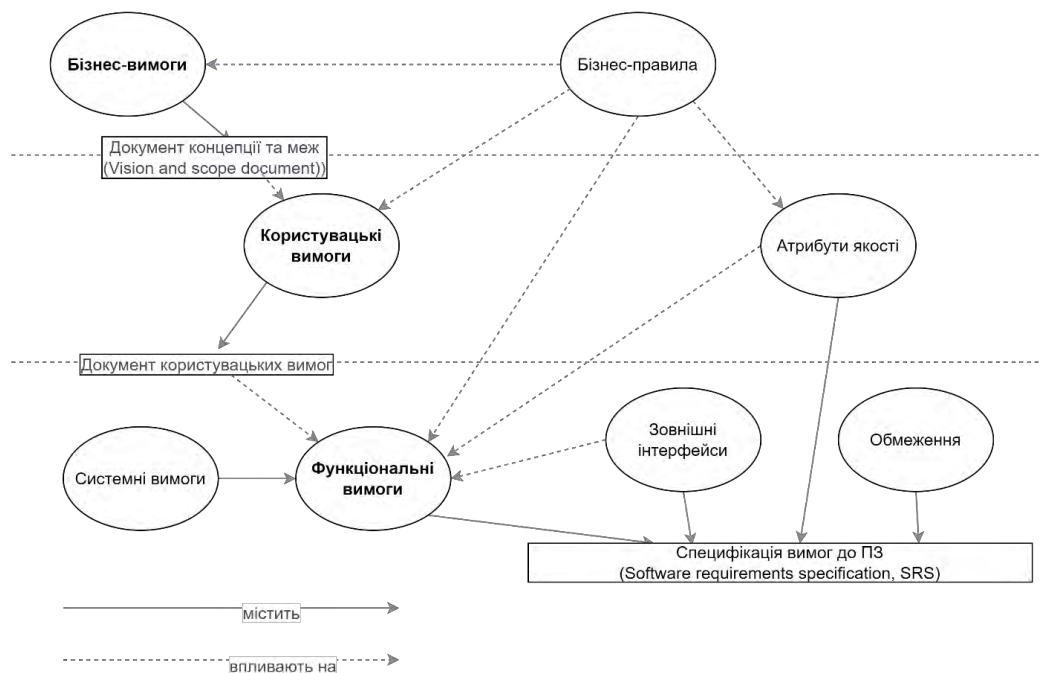


Рис. 2. Зв'язок типів інформації для формування вимог [13]

На практиці використовують різні способи організації інформації про вимоги (табл. 2): від написаних природною мовою детальних специфікацій до опису вимог формальною мовою.

Крім зазначених стандартних підходів до опису вимог (табл. 2), поступово в практику впроваджуються й новітні підходи до опису вимог та використання шаблонів. Наприклад, у роботі [16] автори пропонують неконтрольований підхід до розпізнавання шаблонів

для опису вимог із допомогою визначення загальних синтаксичних структур вимог.

На вибір способу подання вимог впливає методологія розроблення програмного забезпечення, а також стандарти, яких дотримується організація-розробник чи які визначені замовником. Крім способу подання вимог, методології та деякі стандарти визначають артефакти (контейнери), у яких зберігаються вимоги (табл. 3).

Таблиця 2. Способи подання вимог

Формат опису вимог	Приклади подання вимог
Документація, у якій використовується чітко структурована природна мова [Natural Language Processing for Requirements Engineering] (INCOSE). Easy Approach to Requirements Syntax (EARS) – нотація, призначена для полегшення запису та документування вимог у чіткій і стислій формі за допомогою природної мови [15].	Software requirements specification Use case User story
Графічні моделі, що ілюструють процеси трансформації стану системи та їх зміни, взаємодії даних, логічні потоки, класи об'єктів і відношення між ними.	Use-case diagram Data Flow Diagram Context Diagram Business Process Model Business Model Canvas Lean Model Canvas
Формальні специфікації, де вимоги визначені за допомогою математично точних, формальних логічних мов.	Vienna Development Method Spec# RAISE Specification Language

Таблиця 3. Артефакти, у яких зберігаються вимоги

Підхід / методологія / фреймворк	Артефакти	Архітектурний елемент
Waterfall	Software requirements specification	Use case Графічні моделі
RUP (Rational Unified Process)	Software requirements specification	Use case User Story Графічні моделі
SCRUM	Product Backlog Roadmap	User Story
SAFe (Scaled Agile Framework)	Product Backlog Roadmap	User Story Enabler Story
ISO/IEC/IEEE	Stakeholder requirements specification System requirements specification Software requirements specification Software quality assurance plan	Use case Структурований опис природною мовою

Як бачимо з табл. 3, результатами бізнес-аналітичної діяльності найчастіше стають такі артефакти, як *Software requirements specification* (специфікація вимог до програмного забезпечення) та *Product Backlog* (беклог продукту), а як архітектурні елементи цих артефактів застосовуються *Use case* (сценарії використання), графічні моделі та *User Story* (користувачка історія) з обов'язковим визначенням зав'язків і залежностей. У цьому разі вимоги до якості програмного забезпечення можуть описуватися за допомогою власних специфічних артефактів [17].

У зазначеному контексті доречно навести ще один термін з праці [1], а саме "структура вимог".

Структура вимог (*Requirements Structure*) дає змогу отримувати вимоги на різних рівнях абстракції чи деталізації та вказує на необхідність установаження логіки вимог. Як приклад, у табл. 4 наведена структура вимог для *Waterfall*- та *Agile*-підходів [18, 19].

Жорсткість і стиль документації безпосередньо залежать від мети та контексту, у якому документація створюється. Наприклад, *Waterfall*-методологія зазвичай передбачає побудову детальної специфікації вимог. *ISO/IEC/IEEE* надає таке визначення специфікації вимог до програмного забезпечення: *Software requirements specification* – структурований набір основних вимог (функцій, продуктивності,

проектних обмежень та атрибутів) програмного забезпечення та його зовнішніх інтерфейсів [2]. Згідно з *IEEE Recommended Practice for Software Requirements Specification*, документ *SRS* має містити детальні вимоги й відповідати певному набору критеріїв якості. Звісно, дотримання рекомендацій та вимог стандартів до побудови *SRS* не завжди

є обов'язковою умовою розроблення програмного забезпечення, але застосування шаблонів, побудованих за певними стандартами, значно полегшує бізнес-аналітичну роботу та підтримує якість її результатів [20]. Для побудови якісної *SRS* у стандартах визначені певні умови та рекомендації (табл. 5).

Таблиця 4. Структура вимог

Рівень	<i>Waterfall</i>	<i>Agile</i>	Акцент
<i>V High</i>	Бізнес-цілі	Бізнес-цілі	Отримання цінності
<i>High</i>	Бізнес-вимоги	Епіки	Планування
<i>Mid</i>	Вимоги стейкхолдерів	<i>User story</i>	Розуміння потреб стейкхолдерів
<i>Low</i>	Вимоги до програмного забезпечення	<i>Task / Acceptance criteria</i>	Допомога розробникам у проектуванні рішення

Таблиця 5. Аналіз стандартів (вимоги до документації)

Документація	Вимоги до документації	Стандарт
1	2	3
Вимоги до програмного забезпечення		
<i>Software Requirements Specifications.</i>	Коректність (<i>Correct</i>); однозначність (<i>Unambiguous</i>); завершеність (<i>Complete</i>); послідовність (<i>Consistent</i>); пріоритетованість (<i>Ranked for importance and/or stability</i>); можливість перевірки (<i>Verifiable</i>); можливість змін (<i>Modifiable</i>); відстежуваність (<i>Traceable</i>).	<i>Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998.</i>
<i>Stakeholder requirements specification, System requirements specification, Software requirements specification.</i>	Стандарт визначає структуру документа, характеристики якісних вимог, характеристики набору вимог.	<i>ISO/IEC/IEEE 29148:2018(en) Systems and software engineering – Life cycle processes – Requirements engineering.</i>
Вимоги до якості програмного забезпечення		
Документ з вимогами – "документ, що містить будь-яку комбінацію вимог або правил, що необхідно виконати за допомогою готового для застосування програмного продукту (<i>RUST</i>)". Ці документи можуть бути технічними звітами, стандартами, переліком вимог (або специфікацією моделі вимог) для різних типів користувачів або статутом чи положенням, введеним керівним або регулювальним органом.	Повнота: достатність інформації для забезпечення якісного використання програмного забезпечення. Коректність: відповідність вимогам основних цільових користувачів. Узгодженість: відсутність суперечностей у межах документа та з іншими документами, наприклад, описом продукту. Зрозумілість: зрозумілість для широкого кола користувачів із застосуванням термінології та стилю, зрозумілих спеціальній аудиторії.	ДСТУ <i>ISO/IEC 25051:2016</i> . Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (<i>SQuaRE</i>). Вимоги до якості готового для застосування програмного продукту (<i>RUSP</i>) та інструкції щодо його тестування (<i>ISO/IEC 25051:2014, IDT</i>).
Шаблон плану проекту з оцінювання якості, що містить вимоги до якості програмних засобів і систем, а також інші вимоги. Джерелом інформації для цього розділу є специфікація вимог до якості програмних засобів або систем(и).	Ефективність специфікації вимог до якості.	ДСТУ <i>ISO/IEC 25001:2016</i> . Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (<i>SQuaRE</i>). Планування та керування (<i>ISO/IEC 25001:2014, IDT</i>).

Продовження таблиці 5

1	2	3
Моделі якості, які можна застосовувати для визначення вимог, валідування повноти визначення вимог. Специфікація вимог до якості.	Прямих вимог до документації не вказано. Доречно виокремити характеристику якості програмного продукту, що можна перекласти на вимоги до документації: <i>покриття контексту</i> . Підхарактеристики: <i>повнота контексту</i> (важливо, щоб продукт був використаний у всіх необхідних умовах застосування) та <i>гнучкість</i> (дає змогу застосовувати продукт у нових умовах). Вилучено підхарактеристику "відповідність", оскільки відповідність нормам і правилам є складником загальних вимог системи, а не складником власне якості. Але цю підхарактеристику доречно спроектувати на документацію, зважаючи що "вимога має пристосовувати модель, обґрунтовуючи будь-які зміни".	ДСТУ ISO/IEC 25010:2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (<i>SQuaRE</i>). Моделі якості системи та програмних засобів (<i>ISO/IEC 25010:2011, IDT</i>).
Документація, що містить вимоги до якості систем програмних засобів.	Вимоги щодо відповідності нормам і правилам можуть використовуватись для формування вимог, але вони виходять за межі вимог до якості.	ДСТУ ISO/IEC 25022:2019. Інженерія систем і програмних засобів. Вимоги до якості систем програмних засобів та їх оцінювання (<i>SQuaRE</i>). Вимірювання якості під час застосування (<i>ISO/IEC 25022:2016, IDT</i>).
Документація, що містить вимоги до якості систем програмних засобів.	Якість системи / програмного продукту – це рівень відповідності заявленим та очікуваним потребам різних груп стейкхолдерів, що дає змогу отримати цінність.	ДСТУ ISO/IEC 25023:2019. Інженерія систем і програмних засобів. Вимоги до якості систем програмних засобів та їх оцінювання (<i>SQuaRE</i>). Вимірювання якості систем та програмних продуктів (<i>ISO/IEC 25023:2016, IDT</i>).

Крім рекомендацій щодо структури *SRS* та вимог до побудови структурованого документа, у стандартах також подано приклади синтаксису вимог, визначено характеристики якісних вимог, а в *ISO29148* наведено ознаки якісного набору вимог. У документі [2] набір вимог поданий як цілісна система, що має відповідати певним характеристикам і забезпечувати досягнення конкретної мети, а отже, вочевидь, має ознаки архітектури вимог. Стандарт наводить такі характеристики набору вимог: повнота (*Complete*, набір вимог не потребує подальшого розширення, оскільки містить усе необхідне для визначення системи або системного елемента), послідовність (*Consistent*, відсутні суперечності та дублювання вимог, використовується єдина термінологія), доступність (*Affordable*, повний набір вимог може бути реалізований як рішення з огляду на наявні умови), обмеженість (*Bounded*, набір вимог не виходить за межі того, що достатнє для задоволення потреб користувача). Зазначений підхід до побудови набору вимог відповідає методології *Waterfall* і дає змогу їх

ретельно перевірити для уникнення змін і зростання вимог ("розповзання вимог") упродовж життєвого циклу, що може вплинути на вартість, графік або якість системи.

Гнучкі ж підходи підтримують мінімально достатню документацію, яка створюється відповідно до потреби, а не за визначеними моделями документації. Цей підхід допомагає документам ефективно враховувати зміни, зберігаючи водночас низькі витрати на їх розроблення та підтримку. Проте обов'язкова документація, необхідна для аудиту чи регуляторної звітності, наявна в кожному циклі постачання. Важливо, щоб документи відповідали виявленим потребам і приносили цінність, яка перевищує витрати на їх розроблення та підтримку. Але умова щодо якості вимог і якості набору вимог, що міститимуться в цих документах, зберігається. Для оцінювання якості вимог, описаних у форматі *user story*, використовуються критерії INVEST: *I* – *Independent* – незалежність, *N* – *Negotiable* – обговорюваність (дискусійність), *V* – *Valuable* – цінність,

E – *Estimable* – вимірність, *S* – *Small* – маленький розмір, *T* – *Testable* – можливість протестувати / перевірити [21]. З детальним описом викликів щодо розроблення якісних вимог у проєктах гнучкого розроблення можна ознайомитися в роботах [23, 25].

Про важливість документування вимог під час реалізації різних підходів до виконання бізнес-аналітичних завдань наголошується і в праці [24]. Кожна вимога документується, щоб чітко визначити, що потрібно реалізувати, на основі цих вимог формується відповідний каталог / набір вимог, повнота якого залежить від організаційних стандартів і підходів до змін. Як зазначалось вище, набір вимог формує пакет вимог, який не завжди є офіційним документом. Це можуть бути історії користувачів і пов'язана інформація. Формальність і рівень деталізації в пакеті вимог залежать від життєвого циклу проєкту та інтересів зацікавлених сторін, що визначаються під час планування [14]. Схожі завдання виконує і архітектура вимог, що описує вимоги, моделі та іншу інформацію про продукт або пов'язані елементи.

З огляду на результати проведеного аналізу необхідно зазначити наявність визначених / запропонованих характеристик якості вимог, наявність визначених / запропонованих ознак якісного набору вимог, наявність визначених / запропонованих вимог до структури документів, що містять інформацію про вимоги до програмного забезпечення. Зважаючи на контекстну діаграму "Визначення архітектури вимог" (рис. 1), усе це дає змогу забезпечити якість вхідних параметрів для побудови архітектури вимог.

ВАВОК [1] наголошує, що кожна вимога має бути сформульована один раз і додана до архітектури вимог так, щоб весь набір можна було оцінити на предмет повноти. Крім того, архітектура вибудовується з огляду на *точки зору* (*viewpoints*), наборів угод / домовленостей, що визначають, як подаються вимоги, як вони організуються та як будуть пов'язані. Також *точки зору* мають брати до уваги інтереси певних груп зацікавлених сторін, а набір скоординованих взаємоповнених *точок зору* забезпечуватиме основу для оцінювання повноти та зв'язності вимог.

Точки зору можуть передбачати стандарти, моделі даних, нотації, моделі бізнес-процесів, варіанти використання та/або досвід користувача (*user experience*) тощо.

Отже, *точка зору* – спосіб / формат надання інформації зацікавленим особам з огляду на їх інтереси; *подання* – це опис фактичних вимог чи дизайнів відповідно до певної точки зору. *Подання вимог* (*Views on requirements*) є складником системи управління вимогами та застосовуються для валідації вимог до програмного забезпечення [3].

Архітектура допомагає переконатися, що набір вимог повний, завдяки структуруванню вимог відповідно до різних *точок зору*. У наборі не має бути втрачених, несумісних з іншими вимог, або вимог, що суперечать одна одній. Архітектура вимог має брати до уваги будь-які залежності між вимогами, що можуть стати на заваді досягненню цілей зацікавлених сторін. Крім того, компонентом архітектури вимог є архітектура інформації, оскільки вона описує, як пов'язана вся бізнес-аналітична інформація, і визначає взаємовідносини таких видів інформації, як вимоги, дизайни, типи моделей і результати виявлення. Для побудови архітектури вимог використовуються архітектурні фреймворки – сукупність *точок зору*, стандартних для галузі, галузевого сектора або організації, наприклад, заздалегідь установлені шаблони, з яких можна розпочати визначення архітектури.

Як бачимо, наведені у ВАВОК підходи до побудови архітектури вимог досить узагальнені й не завжди корелюють із галузевими стандартами, наприклад, з *ISO29148* щодо опису дизайну в структурі *SRS*. Також на момент написання статті авторам не вдалося знайти рекомендацій та кращих практик побудови якісної архітектури вимог до програмного забезпечення, а також загальноприйнятого визначення поняття "архітектура вимог до програмного забезпечення".

Отже, за результатами аналізу вхідних параметрів, можливих архітектурних елементів і вимог до їх якості, наявних загальних рекомендацій щодо побудови набору вимог пропонується надати визначення поняттю "архітектура вимог".

Розглянемо спочатку етимологію поняття "архітектура". "Архітектура (лат. *architectura*, від грец. *ἀρχιτέκτων* – провідний будівничий) – зодчество, мистецтво зводити будівлі відповідно до вимог міцності, краси й доцільності; у ширшому значенні – частина матеріальної культури (самі споруди), сфера матеріального виробництва" [25]. З погляду системного аналізу з цього визначення виокремимо складну систему – "культуру", її структурні елементи – "споруди", умови функціонування – "відповідність

вимогам", спроектуємо їх на архітектуру вимог та її архітектурні елементи – *подання*, та на цій основі надамо визначення архітектурі вимог до програмного забезпечення. Також у формулюванні визначення взято до уваги, що архітектура вимог використовується для розуміння, які моделі підходять для предметної галузі, допомагає організувати вимоги до структур відповідно до груп зацікавлених сторін, забезпечує візуалізацію взаємодії та залежностей вимог, а також загальну спрямованість вимог на досягнення спільних цілей і прийняття компромісних рішень щодо вимог.

Архітектура вимог – це організована з огляду на контекст і бачення зацікавлених сторін сукупність різноманітного *подання*, що дає змогу висувати вимоги до продукту (програмного забезпечення) як до цілісної системи.

Доречно звернути увагу на такі припущення:

1) архітектура вимог може передбачати / базуватися на використанні певних шаблонів документів, але не обмежується ними;

2) дизайн є елементом архітектури.

У контексті цієї статті зазначені припущення подаються авторами без доведення та обґрунтування і стануть предметом наступних досліджень.

Висновки й перспективи подальшого розвитку

Досліджуючи поняття "архітектура вимог" як бізнес-аналітичного продукту, автори статті пройшли

шлях від систематизації та узагальнення різноманітних підходів до визначення поняття до виокремлення його сутності в контексті розроблення програмного забезпечення. Результати аналізу показали, що архітектура вимог відіграє важливу роль у забезпеченні повноти, узгодженості та якості вимог, а також у розумінні потреб зацікавлених сторін. Виявлено, що архітектура вимог може базуватися на різних архітектурних фреймворках, залежить від методологій розроблення програмного забезпечення, але водночас вона може варіюватися залежно від галузевих стандартів і потреб конкретного проекту. Також архітектура вимог може використовувати певні шаблони документів, проте вона не обмежується ними.

Крім того, проведені дослідження довели необхідність подальшої роботи в цій сфері, бо відкритими залишаються питання стандартизації поняття "архітектура вимог" та розроблення рекомендацій щодо його практичного застосування. Надалі можливе розширення дослідження щодо аналізу впливу архітектури вимог на процес розроблення програмного забезпечення та його якості. Також важливо провести порівняльний аналіз різних методів та інструментів побудови архітектури вимог для виявлення найефективніших підходів. Загалом подані в статті здобутки відкривають перспективи для подальших досліджень у частині побудови архітектури вимог, що має значний потенціал для вдосконалення процесів розроблення програмного забезпечення та підвищення його якості.

Список літератури

1. A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide). International Institute of Business Analysis, Toronto, Ontario, Canada, 2015. 514 p.
2. Systems and software engineering – Life cycle processes – Requirements engineering. ISO/IEC/IEEE 29148:2018(en). URL: <https://doi.org/10.3403/30295695u> (дата звернення: 10.12.2023).
3. Pohl K. Requirements engineering: fundamentals, principles, and techniques. *Springer Publishing Company*. 2010. 182 p. DOI: <http://dx.doi.org/10.5860/choice.48-3304>
4. Lind E., Gonzalez-Huerta J., Alégroth E. Requirements Quality vs. Process and Stakeholders' Well-Being: A Case of a Nordic Bank. *Lecture Notes in Business Information Processing*. 2023. № 472. P. 17–37. DOI: https://doi.org/10.1007/978-3-031-31488-9_2
5. BABOK® Guide v3 Glossary – Ukrainian Translation. URL: <https://www.iiba.org/globalassets/standards-and-resources/glossary/files/babok-v3-glossary-ukrainian.pdf> (дата звернення: 10.12.2023).
6. Gobov D., Yanchuk V. Network Analysis Application to Analyze the Activities and Artifacts in the Core Business Analysis Cycle. *Proceedings of the 2nd International Informatics and Software Engineering Conference (IISEC)*. IEEE. 2021. P. 1–6. DOI: <https://doi.org/10.1109/IISEC54230.2021.9672373>
7. Diev S. Structuring complex requirements. *ACM SIGSOFT Software Engineering Notes*. 2007. №32(2). P. 1–5. DOI: <https://doi.org/10.1145/1234741.1234755>
8. Gobov D., Huchenko I. Modern Requirements Documentation Techniques and the Influence of the Project Context: Ukrainian IT Experience. *Lecture Notes on Data Engineering and Communications Technologies*. 2022. vol 134. P. 260–270. DOI: https://doi.org/10.1007/978-3-031-04812-8_22

9. Gobov D. Practical Study on Software Requirements Specification and Modelling Techniques. *International Journal of Computing*. 2023. №22(1). P. 78-86. DOI: <https://doi.org/10.47839/ijc.22.1.2882>
10. Rączkowska-Gzowska K., Walkowiak-Gall A. What Should a Good Software Requirements Specification Include? Results of a Survey. *Foundations of Computing and Decision Sciences*. 2023. № 48(1). P. 57–81. DOI: 10.2478/fcds-2023-0004
11. Mucha J., Kaufmann A., Riehle D. A systematic literature review of pre-requirements specification traceability. *Requirements Engineering*. 2024. №1. P. 1–23. DOI: <https://doi.org/10.1007/s00766-023-00412-z>
12. Großer K., Riediger V., Jürjens J. Requirements document relations: A reuse perspective on traceability through standards. *Software and Systems Modeling*. 2022. №21(6). P. 1–37. DOI: <https://doi.org/10.1007/s10270-021-00958-y>
13. Wiegers K., Beatty J. *Software Requirements (Developer Best Practices)*. 3rd Edition, Microsoft Press, 2013, 672 p.
14. The PMI Guide to Business Analysis. Project Management Institute, Inc, Newtown Square, Pennsylvania, 2018. 444 p.
15. Gregory S. Requirements for the New Normal: Requirements Engineering in a Pandemic. *IEEE Software*. 2021. vol. 38, No. 2. P. 15–18. DOI: 10.1109/MS.2020.3044403
16. Sonbol R., Rebdawi G., Ghneim N. Learning software requirements syntax: An unsupervised approach to recognize templates. *Knowledge-Based Systems*. 2022. Volume 248. 108933 p. DOI: <https://doi.org/10.1016/j.knosys.2022.108933>
17. Fratini K, et.al. Requirements quality research: a harmonized theory, evaluation, and roadmap. *Requirements engineering*. 2023. Volume 28. P. 507–520. DOI:<https://doi.org/10.1007/s00766-023-00405-y>
18. Noreika K., Gudas S. Causal Knowledge Modelling for Agile Development of Enterprise Application Systems. *Informatica*. 2023. №34(1). P. 121–146. DOI: 10.15388/23-INFOR510
19. Palomares C., et. al. The state-of-practice in requirements elicitation: an extended interview study at 12 companies. *Requirements Engineering*. 2021. №26. P. 273–299. DOI: <https://doi.org/10.1007/s00766-023-00399-7>
20. Gobov D., Zuieva O. Identifying the dependencies between it project context and business analysis document content. *Innovative technologies and scientific solutions for industries*. 2023. №2(24). P. 39–53. DOI: 10.30837/ITSSI.2023.24.039
21. Dellsén E., Westgårdh K., Horkoff J. Invest in Splitting: User Story Splitting Within the Software Industry. *Lecture Notes in Computer Science*. 2022. Vol 13216. P. 115-130. DOI: https://doi.org/10.1007/978-3-030-98464-9_10
22. Rasheed A., et. al. Requirement Engineering Challenges in Agile Software Development. *Mathematical Problems in Engineering*. 2021. Volume 2021. Article ID 6696695. P. 1–18. DOI: <https://doi.org/10.1155/2021/6696695>
23. Atoum I, et.al. Challenges of Software Requirements Quality Assurance and Validation: A Systematic Literature Review. *IEEE Access*. 2021. Vol. 9. P. 137613–137634. DOI: <https://doi.org/10.1109/ACCESS.2021.3117989>
24. Paul D., Cadle J., Yeates D. (editors). *Business Analysis. Third edition*, British Computer Society, 2014. 401 p.
25. С. К. Кілессо. Архітектура. *Енциклопедія сучасної України*. [Електронний ресурс] за ред.: І. М. Дзюба, А. І. Жуковський, М. Г. Железняк [та ін.]; НАН України, НТШ. К.: *Інститут енциклопедичних досліджень НАН України*, 2001. URL : <https://esu.com.ua/article-44834>

References

1. A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide). International Institute of Business Analysis, Toronto, Ontario, Canada, 2015. 514 p.
2. Systems and software engineering – Life cycle processes – Requirements engineering. ISO/IEC/IEEE 29148:2018(en). available at: <https://doi.org/10.3403/30295695u> (last accessed: 10.12.2023).
3. Pohl, K. (2010), "Requirements engineering: fundamentals, principles, and techniques". *Springer Publishing Company*. 182 p. DOI: <http://dx.doi.org/10.5860/choice.48-3304>
4. Lind, E., Gonzalez-Huerta, J., & Alégroth, E. (2023), "Requirements Quality vs. Process and Stakeholders' Well-Being: A Case of a Nordic Bank". *Proceedings of International Conference on Software Quality*, P. 17–37. DOI: https://doi.org/10.1007/978-3-031-31488-9_2
5. International Institute of Business Analysis. "BABOK® Guide v3 Glossary – Ukrainian Translation", available at: <https://www.iiba.org/globalassets/standards-and-resources/glossary/files/babok-v3-glossary-ukrainian.pdf> (last accessed: 10.12.2023).
6. Gobov, D., Yanchuk, V. (2021), "Network Analysis Application to Analyze the Activities and Artifacts in the Core Business Analysis Cycle", *Proceedings of the 2nd International Informatics and Software Engineering Conference (IISEC)*, IEEE. P. 1–6. DOI: <https://doi.org/10.1109/IISEC54230.2021.9672373>
7. Diev, S. (2007), "Structuring complex requirements", *ACM SIGSOFT Software Engineering Notes*, №32(2), P. 1–5. DOI: <https://doi.org/10.1145/1234741.1234755>

8. Gobov, D., Huchenko, I. (2022), "Modern Requirements Documentation Techniques and the Influence of the Project Context: Ukrainian IT Experience", *Lecture Notes on Data Engineering and Communications Technologies*, Vol 134, P. 260–270. DOI: https://doi.org/10.1007/978-3-031-04812-8_22
9. Gobov, D. (2023), "Practical Study on Software Requirements Specification and Modelling Techniques", *International Journal of Computing*, №22(1), P. 78–86. DOI: <https://doi.org/10.47839/ijc.22.1.2882>
10. Rączkowska-Gzowska, K., Walkowiak-Gall, A. (2023), "What Should a Good Software Requirements Specification Include? Results of a Survey", *Foundations of Computing and Decision Sciences*, № 48(1), P. 57–81. DOI: 10.2478/fcds-2023-0004
11. Mucha, J., Kaufmann, A., Riehle, D. (2024), "A systematic literature review of pre-requirements specification traceability", *Requirements Engineering*, №1, P. 1–23. DOI: <https://doi.org/10.1007/s00766-023-00412-z>
12. Groß, K., Riediger, V., Jürjens, J. (2022), "Requirements document relations: A reuse perspective on traceability through standards", *Software and Systems Modeling*, №21(6), P. 1–37. DOI: <https://doi.org/10.1007/s10270-021-00958-y>
13. Wiegers K., Beatty J. (2013), *Software Requirements (Developer Best Practices). 3rd Edition*. Microsoft Press, 672 p.
14. Project Management Institute (2018), *The PMI Guide to BUSINESS ANALYSIS*. PMI, Newtown Square, Pennsylvania, 444 p.
15. Gregory, S. (2021), "Requirements for the New Normal: Requirements Engineering in a Pandemic", *IEEE Software*, vol. 38, No. 2, P. 15–18. DOI: 10.1109/MS.2020.3044403
16. Sonbol, R., Rebdawi, G., Ghneim, N. (2022), "Learning software requirements syntax: An unsupervised approach to recognize templates", *Knowledge-Based Systems*, Volume 248, 108933 p. DOI: <https://doi.org/10.1016/j.knosys.2022.108933>
17. Frattini, K., et.al. (2023), "Requirements quality research: a harmonized theory, evaluation, and roadmap", *Requirements engineering*, Volume 28. P. 507–520. DOI: <https://doi.org/10.1007/s00766-023-00405-y>
18. Noreika, K., Gudas, S. (2023), "Causal Knowledge Modelling for Agile Development of Enterprise Application Systems", *Informatica*. №34(1). P. 121–146. DOI: 10.15388/23-INFOR510
19. Palomares, C., et. al. (201), "The state-of-practice in requirements elicitation: an extended interview study at 12 companies", *Requirements Engineering*. №2. P. 273–299. DOI: <https://doi.org/10.1007/s00766-023-00399-7>
20. Gobov, D., Zuieva, O. (2023), "Identifying the dependencies between it project context and business analysis document content", *Innovative technologies and scientific solutions for industries*, №2 (24), P. 39–53. DOI: 10.30837/ITSSI.2023.24.039
21. Dellsén, E., Westgårdh, K., Horkoff, J. (2022), "Invest in Splitting: User Story Splitting Within the Software Industry", *Lecture Notes in Computer Science*, Vol 13216, P. 115–130. https://doi.org/10.1007/978-3-030-98464-9_10
22. Rasheed, A., et. al. (2021), "Requirement Engineering Challenges in Agile Software Development", *Mathematical Problems in Engineering*, Volume 2021, Article ID 6696695. P. 1–18. DOI: <https://doi.org/10.1155/2021/6696695>
23. Atoum, I., et.al. (2021), "Challenges of Software Requirements Quality Assurance and Validation: A Systematic Literature Review", *IEEE Access*, Vol. 9, P. 137613–137634. DOI: <https://doi.org/10.1109/ACCESS.2021.3117989>
24. Paul, D., Cadle, J., Yeates, D. (2014), *Business Analysis. Third edition*, British Computer Society, 401 p.
25. Kileso, S.K. (2001), "Architecture. Encyclopedia of Modern Ukraine", edited by: I. M. Dzyuba, A. I. Zhukovsky, M. G. Zheleznyak [and others], National Academy of Sciences of Ukraine, National Academy of Sciences. K.: Institute of Encyclopedic Research of the National Academy of Sciences of Ukraine. available at: <https://esu.com.ua/article-44834>.

Надійшла 24.02.2024

Відомості про авторів / About the Authors

Гобов Денис Андрійович – кандидат технічних наук, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", старший викладач кафедри інформатики та програмної інженерії, Київ, Україна; e-mail: d.gobov@kpi.ua; ORCID ID: <https://orcid.org/0000-0001-9964-0339>

Шевченко Наталя Юрївна – кандидат економічних наук, доцент, ТОВ "Технічний Університет "Метінвест Політехніка", доцент кафедри цифрових технологій та проєктно-аналітичних рішень, Запоріжжя, Україна; e-mail: nataliya.shevchenko@mipolytech.education; ORCID ID: <https://orcid.org/0000-0002-3085-710>

Gobov Denis – PhD (Engineering Sciences), National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Associate Professor at the Department of Computer Science and Software Engineering, Kyiv, Ukraine.

Shevchenko Natalia – PhD (Economic Sciences), Associate Professor, Technical University "Metinvest Polytechnic" LLC, Associate Professor at the Department of Digital technologies and project-analytical solutions, Zaporizhzhia, Ukraine.

DEFINITION OF THE REQUIREMENTS ARCHITECTURE FOR IT SOLUTION AS A BUSINESS ANALYTICS PRODUCT

The **subject** of research in the article is approaches to understanding and interpreting the concept of the architecture of requirements for an IT solution. The **purpose** of the work is to define the architecture of software requirements for a better understanding and further implementation of this concept in the practice of software development. The article addresses the following **tasks**: analyze approaches to the classification of requirements to determine common and distinctive characteristics and their impact on the definition of the architecture of requirements; examine requirements engineering standards to find the definition and any relationship to the requirements architecture; consider approaches to building a set of requirements and using templates to describe them to identify optimal strategies for business analytical processes and their impact on the architecture of requirements. The following **methods** are used: comparative analysis, including comparison and generalization of different classifications of requirements taking into account their impact on the definition of requirements architecture, review of literature and standards related to requirements engineering, as well as analysis of their impact on the process of formation of requirements architecture, analysis of documentation with the aim identifying optimal strategies for building the requirements architecture, as well as analyzing the use of templates for describing requirements and their importance in defining the requirements architecture. In addition, the requirements quality criteria analysis results were interpolated to identify relationships between the quality of requirements and their architecture, which made it possible to determine the optimal approaches to building the architecture of requirements that meet the requirements of software quality. The following **results** were obtained: approaches to understanding requirements in software development were investigated, generally accepted classifications of requirements and characteristics of quality requirements were given, and links of the chain "requirements – quality of requirements – approaches to description of requirements – standards – architecture of requirements" were analyzed with further logical interpolation of the studied context on the architecture of requirements, the author's definition of the concept of architecture of requirements is proposed. **Conclusions.** Requirements architecture can be based on different architectural frameworks and depends on software development methodologies, but at the same time, it can vary depending on industry standards and the needs of a specific project. Also, requirements architecture can use certain document templates, but it is not limited to them.

Keywords: business analysis; software requirements; documentation of requirements; requirements view; requirements viewpoint; requirements architecture.

Бібліографічні описи / Bibliographic descriptions

Гобов Д. А., Шевченко Н. Ю. Визначення архітектури вимог до ІТ-рішення як бізнес-аналітичного продукту. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024. № 1 (27). С. 26–38.

DOI: <https://doi.org/10.30837/ITSSI.2024.27.026>

Gobov, D., Shevchenko, N. (2024), "Definition of the requirements architecture for IT solution as a business analytics product", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (27), P. 26–38.

DOI: <https://doi.org/10.30837/ITSSI.2024.27.026>
