

В.В. Любченко

АРХІТЕКТУРНІ МЕТРИКИ: ОГЛЯД ПОТЕНЦІАЛУ ДЛЯ ПОКРАЩЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розроблення програмного забезпечення залежить від добре продуманої архітектури, яка впливає на всю систему. У цьому дослідженні розглядається використання метрик архітектурного проектування, які кількісно оцінюють різні аспекти архітектури. Ми провели систематичне картографічне дослідження, проаналізувавши існуючу літературу, щоб зрозуміти, як ці метрики використовуються зараз. Наші висновки показують обмежений, але багатообіцяючий ландшафт. Існують метрики для оцінки стабільності, зрозумілості, модульності та безпеки проекту. Зокрема, кілька досліджень вивчають вимірювання «здатності генерувати варіанти» архітектури та її потенціалу генерувати різні проєктні варіанти. Однак існує прогалина у безпосередньому використанні цих метрик для прогнозування несправностей. Існуючі дослідження зосереджені переважно на «зворотному» ефекті, коли метрики з більш пізніх етапів розроблення (наприклад, характеристики коду) використовуються для виявлення архітектурних проблем. Загалом, це дослідження підкреслює недостатньо використаний потенціал метрик архітектурного проектування. Хоча поточні дослідження демонструють ефективність відносно простого набору метрик для різних цілей, подальші дослідження є виправданими. Майбутні зусилля повинні бути спрямовані на накопичення даних і дослідження моделей для використання цих метрик з метою прогнозування, що в результаті підвищить якість програмного забезпечення та покращить процеси розробки.

Ключові слова: метрики архітектурного проектування, систематичне дослідження публікацій, оцінювання проєкту програмного забезпечення, атрибути якості архітектури.

V. Liubchenko

ARCHITECTURE METRICS: A SURVEY OF THE POTENTIAL TO IMPROVE SOFTWARE

Software development hinges on well-crafted architecture, a foundational phase that influences the entire system. This study investigates using architectural design metrics, which quantify various aspects of architecture. We conducted a systematic mapping study, analysing existing literature to understand how these metrics are currently employed. Our findings reveal a limited yet promising landscape. Metrics exist to assess design stability, understandability, modularity, and security. Notably, several studies explore gauging the "option-generation ability" of an architecture and its potential to generate different design choices. However, a gap exists in directly using these metrics for fault prediction. Existing research primarily focuses on the "reverse" effect, where metrics from later development stages (like code) are used to identify architectural issues. Overall, this study highlights the underutilised potential of architectural design metrics. While current research demonstrates the effectiveness of a relatively simple set of metrics for various purposes, further exploration is warranted. Future efforts should delve into data accumulation and investigate models for using these metrics for predictive purposes, ultimately enhancing software quality and development processes.

Keywords: software architecture design metrics, systematic mapping study, software design evaluation, architectural quality attributes.

Вступ

Розроблення програмного забезпечення (ПЗ) значною мірою покладається на архітектурне проектування – ключовий етап, який суттєво впливає на кінцеву систему. По суті, архітектурне проектування передбачає створення високорівневої абс-

тракції топології, функціональності та поведінки системи, що закладає основу для детального проектування та реалізації. Існує багато визначень архітектури ПЗ, але всі вони, як правило, стосуються компонентів, модулів та їхніх взаємозв'язків.

Керуючись усталеними моделями проектування, найкращими практиками та міркуваннями зацікавлених сторін, архітектурне проектування має надавати пріоритет дотриманню перевірених методологій. Цей процес часто передбачає пошук компромісів для узгодження суперечливих цілей. Ретельно спланований дизайн є незамінним на всіх етапах створення та розвитку системи, особливо при внесенні змін та адаптації.

Якісна архітектура слугує основою для поточної розробки ПЗ. Логічно припустити, що атрибути архітектурного дизайну формують якості окремих артефактів розробки та ПЗ в цілому. Якби ми могли кількісно описати ці залежності та впливи, то могли б передбачити характеристики цих артефактів і всієї програмної системи.

Набори даних, які зазвичай використовуються в емпіричних дослідженнях програмної інженерії, такі як репозиторії NASA, переважно містять характеристики коду та метрики дефектів. Отже, доречно дослідити використання та призначення метрик, що описують архітектурний дизайн, в опублікованих дослідженнях.

Самі собою метрики не визначають якість архітектури як хорошу чи погану. Однак вони слугують цінними допоміжними засобами для архітекторів, висвітлюючи потенційні проблемні елементи та сфери, які потребують вдосконалення в їхніх проєктах. Інструмент аналізу архітектури ПЗ обчислює різні метрики, дотримуючись певних рекомендацій щодо метрик ПЗ, а саме:

- метрики мають бути зрозумілими та простими для розуміння та застосування.
- метрики повинні відповідати очікуванням та практичному досвіду інженерів.
- метрики повинні давати однозначні результати.
- метрики мають підтримувати математичну узгодженість.
- метрики мають бути застосовні до різних мов програмування.
- метрики повинні пропонувати дієві ідеї для покращення якості та продуктивності ПЗ.

Основна мета цієї статті – узагальнити існуючий досвід використання метрик архітектурного проектування. Крім того, ми прагнемо визначити перспективні сфери їхнього застосування для покращення процесу розроблення ПЗ, так і кінцевого продукту.

1. Метод дослідження

Ми провели систематичне дослідження публікацій, щоб проаналізувати існуючі публікації про використання метрик архітектурного дизайну ПЗ. Це дослідження є цінним інструментом для вивчення еволюції досліджень у цій галузі з плином часу. Нижче ми описуємо методологію, яка була використана у процесі проведення цього дослідження.

Основною метою цього дослідження є систематизація метрик, що характеризують дизайн архітектури та методи їхнього застосування в процесах програмної інженерії, з фокусом на моделях якості та мета-моделях. Для досягнення цієї мети ми дослідили простір метрик проектування ПЗ та їх підтримку для процесів програмної інженерії. Це дослідження має на меті надати огляд сучасних досліджень з оцінки архітектури ПЗ, зосереджуючись на метриках, визначених для архітектурних діаграм. Питання дослідження є наступними:

RQ1. Які метрики застосовуються до архітектурного дизайну? Це питання є ключовим, оскільки воно формує основу нашого розуміння того, як вимірюється та оцінюється архітектурний дизайн.

RQ2. Для яких цілей вони застосовуються?

Щоб відповісти на ці запитання, було розроблено процес пошуку.

Ми провели пошук у чотирьох довідкових базах даних: IEEE Xplore, ACM Digital Library, ScienceDirect та Elsevier, щоб знайти статті про метрики проектування архітектури. Основними ключовими словами були "дизайн архітектури ПЗ" та "метрики".

Пошук охоплював поля назви, анотації та ключові слова, а також сформульований пошуковий рядок:

*("software architecture design") AND
(("metrics" OR "measuring"))).*

Цей рядок було закодовано відповідно до синтаксичного формату кожної відповідної бази даних.

Пошук у базах дав початковий набір з 97 статей. Ці статті пройшли фільтрацію у два етапи на основі попередньо визначених критеріїв відбору

На першому етапі ми використовували ряд критеріїв включення та виключення для розгляду та відбору статей. До розгляду включалися статті написані англійською мовою, опубліковані в журналах або матеріалах конференцій, тема яких пов'язана з проектуванням та вимірюванням архітектури ПЗ. З розгляду були виключені статті, написані як систематичні дослідження публікацій, або предметом яких є не ПЗ, а архітектура апаратної інфраструктури. Також були виключені статті про метрики призначені для конкретної характеристики якості, такої як безпека.

На другому етапі фільтрація базувалася на ознайомленні з повним текстом. Як наслідок були відфільтровані, наприклад, роботи з метрик подібності, метрик детального проектування та програмного коду, залежностей між властивостями коду та "запахом" архітектури, а також бази знань про архітектуру.

Після процедури відбору для систематичного картографічного дослідження було відібрано лише 19 первинних робіт, що становить лише 19,6% від початкового набору даних. Така обмежена кількість підкреслює очевидний брак значного досвіду успішного застосування таких метрик.

2. Результати аналізу

Перша метрика, яка нас цікавить, – це вартість, пов'язана з еволюцією архітектурних проєктів [1]. Еволюція програмної архітектури передбачає зміни в компонентах та з'єднувачах. Для оцінки еволюції автори запропонували метрики на основі операцій перезапису: додавання, видалення та модифікація. Автори визначили вартість змін як зважену суму вартості кожної операції. Метрика досить складна, її застосування дуже специфічне.

Того ж року в [2] було описано цілу низку архітектурних метрик, зокрема, зв'язність, зчеплення, складність сервісів, кількість послуг на компонент, fan-in і fan-out, глибина сценарію. Це дослідження цікаве тим, що більшість перелічених метрик були використані в багатьох пізніших дослідженнях. Також тут зроблено спробу встановити зв'язок з фазою аналізу вимог: метрика глибини сценарію орієнтована на складність сценаріїв використання, а не на внутрішню складність самої архітектури.

Зупинимося на метриках зчеплення, які залишаються затребуваними протягом багатьох років. В [3] запропоновано визначити зчеплення на основі кількості вимог, пов'язаних з програмним компонентом, кількості функцій та середній взаємодії між функціями в межах компонентів. В [4] введено метрику відсутності зчеплення, яка вимірює схожість між операціями в межах певних сервісів та їхній взаємозв'язок. В [5] емпірично досліджено практичну життєздатність метрик зчеплення на рівні сервісів у контексті застосування архітектури мікросервісів.

В [6] було представлено набір метрик стабільності та модульності проєкту. Зокрема, автори розробили метрику волатильності для оцінки стабільності рішень на основі кількості умов навколишнього середовища, що впливають на них, та широти їхнього впливу. Для модульності автори представили метрику «Рівень незалежності», спрямовану на вимірювання здатності проєкту сприяти пошуку і заміні модулів.

В [7] було представлено комплексну метрику «Баланс компонентів», яку утворюють два показники: розбиття системи, яке оцінює ступінь декомпозиції системи на розумну кількість компонентів, та рівномірність розмірів компонентів, яка оцінює узгодженість розмірів компонентів у системі. У наступному дослідженні [8] ті ж автори розглянули дві метрики рівня архітектури: баланс компонентів та профілі залежностей.

В [9] було введено спеціальну метрику для комплексної оцінки розміру системи та зусиль, необхідних для управління її загальною архітектурою. Абсолютний

індекс програмних елементів визначає загальну кількість програмних елементів, що використовуються в системі.

Дослідження [10] зосереджене на взаємодіях в архітектурі. Введено дві метрики: коефіцієнт взаємодії компонента (відношення вихідних взаємодій до вхідних) та відсоток взаємодії між компонентами (відношення суми вхідних та вихідних взаємодій до максимальної кількості потенційних взаємодій).

На відміну від попередніх досліджень, у [11] запропоновано використовувати фундаментальні метрики: кількість компонентів, кількість з'єднувачів (незалежно від їхнього спрямування) та кількість елементів (сукупність компонентів та з'єднувачів). Автори назвали ці метрики зрозумілістю, пропонуючи прямий підхід до оцінки зрозумілості системи.

У дослідженні [12] було зібрано та структуровано попередню компіляцію архітектурних метрик відповідно до точок зору та доменів. Тим не менше, дослідження дійшло висновку, що оцінка загальної корисності архітектурних метрик залишається відкритим дослідницьким питанням.

Цікавим є [13], яке зосередилося на кількісній оцінці архітектурного занепаду, спираючись на метрики когерентних взаємодій, нестабільності та якості модуляризації. Автори також визначили нові метрики для виявлення архітектурних запахів: двонаправлений зв'язок компонентів, щільність та покриття архітектурного запаху. Ці метрики допомагають інженерам передбачати помилки системи.

У [14] автори класифікували встановлені метрики в ієрархічну структуру і ввели нові агреговані метрики, що охоплюють структурну якість і системні характеристики, зокрема гнучкість. Це дозволило оцінити ефективність метрик на етапі проектування архітектури.

У [15] розглядалися метрики зв'язків між вимогами та архітектурним проектом: Traced Requirements per Component визначає кількість вимог на компонент, а Traced Components per Requirement оцінює кількість компонентів на вимогу. Ці зале-

жності можуть бути індикаторами дефектів ПЗ.

У [16] представлено метрики рівня архітектури для оцінки якості: рівень роз'єднання (Decoupling Level), що оцінює розділеність системи на автономні модулі, і вартість розповсюдження (Propagation Cost), що оцінює щільність зв'язку всередині системи. Також використовувались метрики баланс компонентів і профілі залежностей. Ефективність цих метрик була підтверджена практиками.

Дослідження [17] зосереджене на метриках безпеки в архітектурній моделі, оцінюючи вразливість точок входу в композитних елементах. Це дослідження акцентує увагу на вивченні безпеки як атрибуту якості.

В [18] розглядаються метрики для архітектури мікросервісів, що встановлюють зв'язок між архітектурними метриками та метриками часу виконання. Це робить можливим проведення оцінок під час проектування. Проте, нещодавня робота [19] вказує, що загальноживані метрики, такі як зв'язність, розмір, зчеплення та складність, також придатні для оцінки архітектури на основі сервісів.

3. Таксономія архітектурних метрик

Кластеризація метрик, що використовуються в архітектурному проектуванні, утворює таксономію метрик. Надамо опис цієї таксономії.

1. Кластер метрик структурної цілісності фокусується на основних структурних аспектах архітектури, включаючи те, наскільки добре компоненти організовані та взаємодіють один з одним, а також наскільки стабільною та збалансованою є система.

1.а. Група метрик згуртованості оцінює ступінь взаємозв'язку та ефективної спільної роботи елементів у межах модуля або компонента.

1.а.1. Архітектурна реляційна когерентність оцінює ступінь, за якого компоненти або модулі в архітектурі ПЗ функці-

онально пов'язані між собою та ефективно працюють разом.

1.a.2 Зв'язність визначає, наскільки добре елементи в межах компонента пов'язані один з одним та виконують єдину сфокусовану функцію.

1.a.3 Показник відсутності зв'язності ідентифікує компоненти з елементами, які погано пов'язані один з одним.

1.a.4. Коефіцієнт зв'язних взаємодій – це співвідношення взаємодій всередині компонента у порівнянні з взаємодіями з іншими компонентами.

1.b. Група метрик складності оцінює складність конструкції та взаємодій системи, що може вплинути на ремонт-придатність та зрозумілість.

1.b.1. Двонаправлений зв'язок компонентів вимірює кількість залежностей між двома компонентами, де кожен з них залежить від іншого.

1.b.2 Зчеплення – це ступінь взаємозалежності між компонентами.

1.b.3 Рівень роз'єднання визначає ступінь незалежності між компонентами.

1.c. Група метрик залежностей аналізує взаємозв'язки та залежності між різними компонентами, які можуть впливати на надійність та модульність системи.

1.c.1. Профілі залежностей описують типи та шаблони залежностей між компонентами.

1.c.2. Включення (Fan-in) визначає кількість компонентів, які залежать від певного компонента.

1.c.3. Розширення (Fan-out) визначає кількість компонентів, від яких даний компонент не залежить.

1.c.4. Витрати на поширення оцінюють зусилля, необхідні для того, щоб зміни в одному компоненті поширилися на інші компоненти, на які вони впливають.

1.d. Група метрик розміру та однорідності оцінює розподіл розмірів та однорідність компонентів, що може вплинути на керованість та збалансованість архітектури.

1.d.1. Баланс компонентів визначає, чи є розподіл функціональності між компонентами збалансованим або асиметричним.

1.d.2. Рівномірність розміру компонентів визначає узгодженість у розмірі компонентів, що вимірюється кількістю методів, розміром інтерфейсів компонент тощо.

1.d.3 Кількість компонентів / з'єднувачів / операцій визначає загальну кількість відповідних елементів у системі.

1.d.4. Кількість послуг компонента визначає кількість послуг, що надаються одним компонентом.

1.e. Група метрик стабільності оцінює, наскільки архітектура є стійкою до змін та наскільки добре вона підтримує свою цілісність з плином часу.

1.e.1 Стабільність архітектури вимірює стійкість архітектури до змін та збоїв.

1.d.2. Нестабільність вимірює схильність архітектури до збоїв та помилок.

2. Група метрик адаптивності та еволюційності розглядає здатність системи адаптуватися до змін та еволюціонувати з часом, а також деталізацію її компонентів, що впливає на гнучкість та масштабованість.

2.a. Група метрик адаптивності вимірює легкість, з якою система може бути адаптована до нових вимог або умов.

2.a.1 Абсолютний індекс адаптивності системи був розроблений для кількісної оцінки ступеню, до якого програмна система може адаптуватися до змін у вимогах, середовищі або технологіях.

2.a.2 Показник вартості еволюції оцінює зусилля, необхідні для модифікації архітектури в майбутньому.

2.b. Група метрик волатильності вимірює частоту та ступінь змін в системі, вказуючи на те, наскільки динамічною та схильною до змін є архітектура.

2.b.1. Волатильність проєкту вимірює, як часто архітектурний проєкт потрібно змінювати через внутрішні або зовнішні фактори.

2.c. Група метрик деталізації оцінює рівень деталізації та декомпозиції сервісів або компонентів, що впливає на гнучкість та масштабованість.

2.c.1. Метрика гранулярності сервісів оцінює розмір та обсяг сервісів у програмній архітектурі.

2.с.2. Показник розбиття системи вимірює загальний стан та ремонтпридатність системної архітектури.

3. Кластер метрик складності та ремонтпридатності досліджує складність системи, яка впливає на те, наскільки архітектура є ремонтпридатною та зрозумілою. Він також визначає «запахи» архітектури, які можуть вказувати на більш глибокі проблеми.

3.a. Група складності оцінює складність дизайну та взаємодій системи, що може вплинути на ремонтпридатність та зрозумілість.

3.a.1 Складність сервісів вимірює складність функціональних можливостей, що надаються сервісами в архітектурі.

3.a.2. Показники складності взаємодії «вхід-вихід» вимірюють складність взаємодії між компонентом та його зовнішнім середовищем.

3.a.3. Масштаб впливу визначає ступінь, за якого зміна в одному компоненті впливає на інші компоненти.

3.b. Група метрик «архітектурних запахів» визначає проблеми або «запахи» в архітектурі, які можуть вказувати на глибші проблеми або області для покращення.

3.b.1. Покриття архітектурного «запаху» вимірює відсоток шаблонів проектування, визначених як потенційні проблеми.

3.b.2. Щільність архітектурного «запаху» підраховує кількість потенційних проблем, виявлених на компонент проекту.

4. Кластер метрик відстежуваності та інших атрибутів якості включає метрики, пов'язані з відстежуваністю вимог та компонентів, а також атрибути якості, які не вписуються в інші групи.

4.a. Група метрик простежуваності вимірює, наскільки добре вимоги та компоненти можуть бути простежені в системі, що впливає на підзвітність та зрозумілість.

4.a.1. Кількість простежуваних компонентів на одну вимогу показує, скільки компонентів пов'язано з однією вимогою.

4.a.2. Кількість відстежених вимог на компонент показує, за скільки вимог відповідає один компонент.

4.b. Група різних метрик включає інші метрики, які не вписуються в інші категорії, але все одно дають цінну інформацію про архітектуру.

4.b.1. Глибина сценарію (Depth of Scenario) визначає кількість компонентів, задіяних у конкретному користувацькому сценарії.

4.b.2. Рівень незалежності (Independence Level) вимірює, наскільки автономними є компоненти.

4.b.3. Якість модульності вимірює, наскільки добре архітектура розділена на незалежні, багаторазово використовувані модулі.

4.b.4. Захищені точки входу вимірюють кількість чітко визначених точок доступу для взаємодії з компонентом.

Обговорення та висновки

Дослідження показали, що метрики проектування архітектури ПЗ можуть бути цінними для дизайнерів. Ці метрики допомагають дизайнерам оцінити два важливих аспекти архітектури: її здатність генерувати різні варіанти проектів та її стабільність за різних зовнішніх умов.

Ми помітили, що рішення на основі метричних стратегій найчастіше використовуються для ідентифікації архітектурного занепаду. Ці метрики можуть визначити зростання архітектурної нестабільності з еволюцією системи, визначити ймовірність класів, що сприяють архітектурним невідповідностям, і діагностувати аномалії, чи є агломерації або окремі компоненти більш корельованими з архітектурними проблемами.

Ми також бачимо, що метрики архітектурного проектування можна використовувати для вимірювання атрибутів якості.

Хоча у вступі обговорювався потенціал метрик архітектурного проектування для прогнозування несправностей, жодне з існуючих досліджень не розглядає це питання безпосередньо. Існуючі дослідження зосереджені на взаємозв'язку між метриками різних етапів розробки. Однак ці дослідження аналізують «зворотній» ефект, коли метрики з більш пізніх етапів (напри-

клад, характеристики коду) застосовуються для виявлення архітектурних проблем, таких як ерозія дизайну або анти-патерни.

Підсумовуючи, можна сказати, що потенціал використання метрик архітектурного проектування використовується не повністю. Існуючі дослідження показали, що один набір відносно простих метрик може бути успішно використаний для різних архітектурних рішень. Тому немає принципових перешкод для використання цих метрик для прогнозування характеристик артефактів наступних етапів розробки та ПЗ в цілому. Отже, подальші дослідження мають бути спрямовані на накопичення даних та дослідження моделей, які можуть бути використані для прогнозування.

Література

1. M. Aoyama, Metrics and analysis of software architecture evolution with discontinuity, International Workshop on Principles of Software Evolution (IWPSE '02), 2002, P. 103–107. doi: 10.1145/512035.512059.
2. J. Muskens, M. R. V. Chaudron, R. Westgeest, Software architecture analysis tool: software architecture metrics collection, 3rd PROGRESS Workshop on Embedded Systems, 2002, P. 128–139.
3. H. Venkitachalam, J. Richenhagen, A. Schlosser, T. Tasky, Metrics for Verification and Validation of Architecture in Powertrain Software Development, First International Workshop on Automotive Software Architecture (WASA '15), 2015, P. 27–33. doi: 10.1145/2752489.2752496.
4. O. Al-Debagy, P. Martinek, A Metrics Framework for Evaluating Microservices Architecture Designs, *Journal of Web Engineering*, 2020. Vol. 19, no. 3–4. P. 341–370. doi: 10.13052/jwe1540-9589.19341.
5. M. G. Moreira, B. B. N. De França, Analysis of Microservice Evolution using Cohesion Metrics, 16th Brazilian Symposium on Software Components, Architectures, and Reuse (SBCARS '22), 2022, P. 40–49. doi: 10.1145/3559712.3559716.
6. K. Sethi, Y. Cai, S. Wong, A. Garcia and C. Sant'Anna, From retrospect to prospect: Assessing modularity and stability from software architecture, Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, 2009, P. 269–272. doi: 10.1109/WICSA.2009.5290817.
7. E. Bouwers, J. P. Correia, A. v. Deursen, J. Visser, Quantifying the Analyzability of Software Architectures, Ninth Working IEEE/IFIP Conference on Software Architecture, 2011, P. 83–92. doi: 10.1109/WICSA.2011.20.
8. E. Bouwers, A. van Deursen, J. Visser, Evaluating usefulness of software metrics: An industrial experience report, 35th International Conference on Software Engineering (ICSE), 2013, P. 921–930. doi: 10.1109/ICSE.2013.6606641.
9. D. Perez-Palacin, R. Mirandola, J. Merseguer, Software architecture adaptability metrics for QoS-based self-adaptation, Joint ACM SIGSOFT conference -- QoSA and ACM SIGSOFT symposium – ISARCS on Quality of software architectures – QoSA and architecting critical systems – ISARCS (QoSA-ISARCS '11), 2011, P. 171–176. doi: 10.1145/2000259.2000288.
10. U. Tiwari, S. Kumar, In-out interaction complexity metrics for component-based software, *SIGSOFT Softw. Eng. Notes*, 2014. Vol. 39(5). P. 1–4. doi: 10.1145/2659118.2659135.
11. S. Stevanetic, T. Haitzer, U. Zdun, Supporting Software Evolution by Integrating DSL-based Architectural Abstraction and Understandability Related Metrics, European Conference on Software Architecture Workshops (ECSAW '14), 2014. Article 19, P. 1–8. doi: 10.1145/2642803.2642822.

12. O. Zimmermann, Metrics for architectural synthesis and evaluation: requirements and compilation by viewpoint: an industrial experience report, Second International Workshop on Software Architecture and Metrics (SAM '15), 2015, P. 8–14.
13. D. Le, Architectural-Based Speculative Analysis to Predict Bugs in a Software System, 38th International Conference on Software Engineering Companion (ICSE-C), 2016, P. 807–810.
14. S. Orlov, A. Vishnyakov, Decision Making for the Software Architecture Structure Based on the Criteria Importance Theory, *Procedia Computer Science*, 2017. Vol. 104. P. 27–34. doi: 10.1016/j.procs.2017.01.050.
15. B. Nassar, R. Scandariato, Traceability Metrics as Early Predictors of Software Defects?, IEEE International Conference on Software Architecture (ICSA), 2017, P. 235–238. doi: 10.1109/ICSA.2017.12.
16. W. Wu et al., Software Architecture Measurement—Experiences from a Multinational Company, *Lecture Notes in Computer Science*, 2018. Vol. 11048. doi: 10.1007/978-3-030-00761-4_20.
17. M. Buitrago, I. Borne, J. Buisson, Deriving metrics for software architectures from the "protected entry points" security patterns, 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), 2023, P. 1473–1475. doi: 10.1145/3555776.3577816.
18. N. Knoll, R. Lichtenthäler, An Experimental Evaluation of Relations Between Architectural and Runtime Metrics in Microservices Systems, 13th International Conference on Cloud Computing and Services Science – CLOSER, 2023, P. 147–154. doi: 10.5220/0011728600003488.
19. M. H. Hasan, M. H. Osman, N. I. Admodisastro, M. S. Muhammad, From Monolith to Microservice: Measuring Architecture Maintainability, *International Journal of Advanced Computer Science and Applications*, 2023. Vol. 14(5). doi: 10.14569/IJACSA.2023.0140591.

Одержано: 08.04.2024

Внутрішня рецензія отримана: 21.04.2024

Зовнішня рецензія отримана: 26.04.2024

Про автора:

Любченко Віра Вікторівна,
Доктор технічних наук
ORCID: 0000-0002-4611-7832

Місце роботи автора:

Національний університет
«Одеська політехніка»,
пр. Шевченка, 1, Одеса, 65044, Україна.
E-mail: lvv@op.edu.ua
Телефон: +38-048-705-85-66