УДК 004.4:004.75:005.8

# ЗАСТОСУВАННЯ МЕТОДОЛОГІЇ AGILE ТА DEVOPS У СИСТЕМНОМУ АНАЛІЗІ: СИНЕРГІЯ ТА ПРОТИРІЧЧЯ

Токар В.Р. [1]; Макаліш Б.Д. [2]; Сиротенко Б.С.[3]; Лучик В.Є., д.е.н., професор[4]
[1,2,3,4]Харківський національний університет внутрішніх справ
[1,2,3,4]Україна, Кам'янець-Подільський
[1] vitaliktokar622@gmail.com; [2] bohdan.makalish@gmail.com; [3] baret7412@gmail.com
[4] luchik.vasil@gmail.com
[2] https://orcid.org/0009-0002-2500-2734; [4] https://orcid.org/0000-0002-1997-0272

***Анотація.*** *Під час дослідження розроблено підхід до інтеграції методологій Agile та DevOps у процес системного аналізу, що дозволяє визначити та систематизувати ключові точки синергії та внутрішні протиріччя. Запропоновано модель гнучкого архітектурного каркасу, що поєднує стратегічне попереднє проектування з ітеративною деталізацією вимог, а також механізм узгодження термінології між аналітиками та розробниками через інтеграцію документації в конвеєр CI/CD. Такий підхід мінімізує технічний борг та ризики втрати узгодженості вимог у швидко-динамічних середовищах розробки.*
***Ключові слова****: системний аналіз; Agile; DevOps; CI/CD; синергія; протиріччя; технічний борг; архітектура.*

***Вступна частина.*** Проблема оцінювання ефективності системного аналізу в умовах сучасних гнучких та девопс-орієнтованих методологій розробки полягає у значній трансформації його класичних інструментів і процесів. Швидкі ітерації Agile, автоматизація DevOps та акцент на безперервній поставці цінності часто суперечать традиційним підходам до формування вимог, проектування архітектури та ведення документації, що призводить до внутрішніх напруг, комунікаційних розривів і, як наслідок, падіння якості кінцевого продукту. Існуючі методики не завжди достатньо ефективно балансують між необхідністю швидкої адаптації та забезпеченням системної цілісності та надійності [1].

***Метою дослідження*** є ефективність процесу системного аналізу шляхом виявлення та аналізу синергетичних ефектів і ключових протиріч між принципами Agile, практиками DevOps і класичними методами аналізу, а також розробка на цій основі практичних рекомендацій щодо їх гармонізації. Для досягнення мети було поставлено завдання проаналізувати, як безперервна взаємодія в Agile трансформує збір вимог; дослідити вплив автоматизації CI/CD на аналітичну документацію; оцінити роль моніторингу DevOps у переоцінці вимог; виявити конфлікт між ітеративністю та архітектурним плануванням; та запропонувати шляхи подолання термінологічних розбіжностей між учасниками команди.

***Основна частина.*** Ітеративний підхід Agile, заснований на принципі безперервної взаємодії з замовником, кардинально реорганізує класичний процес системного аналізу, трансформуючи його з жорсткої послідовності в гнучку, циклічну модель. На зміну водоспадному підходу, де вимоги фіксуються на початку та рідко переглядаються, приходить динамічний процес постійного уточнення через щоденні комунікації, спринт-ревю та регулярне оновлення Product Backlog. Це не лише зменшує ризик створення продукту, що не відповідає очікуванням, але й залучає замовника до активної участі, перетворюючи його на частину команди, що, втім, вимагає від нього готовності до гнучкості та постійної включеності в процес. Така синергія забезпечує безцінну здатність швидко адаптуватися до змін, проте

створює нові виклики, пов’язані з необхідністю підтримки стабільності та цілісності системи в умовах постійної еволюції вимог [1].

Ця динаміка, породжена Agile, знаходить своє логічне продовження та посилення в практиках DevOps, де автоматизація конвеєрів CI/CD надає нову роль документації системного аналізу. Ізольовані артефакти та специфікації перетворюються на живий, інтегрований компонент розробки, вбудований безпосередньо в робочий процес. Вимоги, описані в Backlog, безпосередньо впливають на створення автоматизованих тестів, які перевіряють код на кожному етапі, а системи контролю версій, як-от Git, забезпечують синхронізацію змін у документації та коді. Це підвищує не лише швидкість розробки, але й якість, оскільки будь-яка невідповідність вимогам виявляється миттєво, а документація стає динамічним джерелом істини, а не статичним архівом. Таким чином, DevOps не лише підтримує ітеративність Agile, але й надає їй технологічний фундамент, що робить процес більш керованим і передбачуваним.

Потужним каталізатором цього симбіозу виступають інструменти моніторингу та зворотного зв’язку DevOps, які замикають петлю між експлуатацією системи та аналізом вимог. Такі системи, як Prometheus або Grafana, надають системному аналітику об’єктивні дані про реальну поведінку продукту в продакшені, виявляючи недоліки, непередбачені сценарії використання та нові потреби користувачів. Ця інформація в реальному часі стає безцінним джерелом для переоцінки та коригування вимог, які миттєво, завдяки тому ж конвеєру CI/CD, перетворюються на оновлення продукту. Створюється безперервний цикл удосконалення, де дані моніторингу безпосередньо впливають на еволюцію вимог, що суттєво знижує технічний борг і забезпечує високу релевантність продукту. Однак ця ж динаміка створює і головну суперечність: конфлікт між швидкістю ітерацій та необхідністю ретельного архітектурного планування.

Саме в цій точці виникає фундаментальне протиріччя між гнучкістю Agile та класичними засадами системного аналізу, що вимагають повноцінного попереднього проектування архітектури. Швидкі спринти, орієнтовані на негайну доставку цінності, часто не залишають часу для глибокого аналізу довгострокових наслідків архітектурних рішень, що може призводити до накопичення технічного боргу та створення фрагментованих, важко інтегрованих рішень. Ітеративна розробка, позбавлена стратегічного архітектурного бачення, ризикує створити систему, яка, хоч і відповідає поточним вимогам, виявляється нееластичною для майбутнього масштабування. Ця напруга між необхідністю швидкої адаптації та забезпеченням структурної цілісності є центральним викликом інтеграції Agile у системний аналіз, що часто вимагає гібридних підходів, таких як попереднє визначення базової архітектури з подальшою гнучкою ітеративною розробкою [2, 3].

Ця суперечність ще більше загострюється через комунікаційний бар’єр, який виникає через різницю в термінології та глибині документації між системними аналітиками та розробниками. Класичні, детальні специфікації аналітиків часто суперечать Agile-принципу «працюючий продукт над вичерпною документацією», що призводить до семантичних розбіжностей у тлумаченні ключових понять, як-от «інкремент» чи «готовість». У середовищі DevOps ця проблема посилюється, оскільки автоматизовані процеси та інструменти моніторингу, такі як Infrastructure as Code, можуть еволюціонувати окремо від формалізованих аналітичних артефактів. Відсутність єдиного словника та узгодженості в глибині опису вимог створює реальну загрозу втрати узгодженості, коли технічна реалізація поступово віддаляється від бізнес-очікувань. Таким чином, подолання цієї комунікаційної прірви та вироблення спільної культури відповідальності стає критичним умовою для реалізації

справжньої синергії між системним аналізом, Agile та DevOps, дозволяючи командам не лише швидко реагувати на зміни, але й будувати надійні, масштабовані та цінні продукти [4].

***Висновки.*** Проведене дослідження підтверджує, що інтеграція Agile та DevOps у системний аналіз має двоїстий характер, породжуючи як потужну синергію, так і глибокі внутрішні протиріччя. До ключових синергій можна віднести трансформацію процесу вимог у ітеративну модель завдяки постійній взаємодії з замовником, інтеграцію документації в автоматизовані конвеєри розробки для підвищення якості та швидкості, а також замикання циклу зворотного зв'язку за допомогою інструментів моніторингу для оперативного виправлення помилок. Одночасно основним протиріччям залишається конфлікт між вимогою до швидких ітерацій та необхідністю ретельного архітектурного планування, яке посилюється комунікаційними бар'єрами через різницю в термінології. Для подолання цих викликів запропоновано реалізовувати гібридні підходи, такі як створення гнучкого архітектурного каркасу на початку проекту та впровадження практик спільної відповідальності для узгодження словника. Подолання цих протиріч відкриває шлях до створення стійких, адаптивних і якісних інформаційних систем, що повністю реалізують потенціал гнучких та девопс-методологій.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] QALight. Agile – QALight. https://qalight.ua/baza-znaniy/agile-dev/
[2] NIX. (2024, 4 листопада). CI/CD: що це, як пов’язано з DevOps, переваги, найкращі практики. https://www.nixsolutions.com/ua/blog/for-developer/ci-cd-shho-cze-yak-povyazano-z-devops-perevagy-najkrashhi-praktyky/
[3] KEY4. (2025, 5 лютого). DevOps (Девопс): що це таке і де застосовується. https://key4.com.ua/uk/blog/devops-devops-chto-eto-takoe-i-gde-primenyaetsya/
[4] QATestLab. (2023, 3 жовтня). Гнучка методологія розробки ПЗ – Agile. https://training.qatestlab.com/blog/technical-articles/flexible-software-development-methodology-agile/

Tokar V. R., Makalish B. D., Syrotenko B. S., Luchyk V. E.

**Application of agile and devops methodologies in system analysis: synergy and contradictions**

***Abstract.*** The study develops an approach to integrating Agile and DevOps methodologies into the system analysis process, allowing the identification and systematization of key synergy points and internal contradictions. A model of a flexible architectural framework is proposed, combining strategic preliminary design with iterative requirements detailing, as well as a mechanism for terminology alignment between analysts and developers through the integration of documentation into the CI/CD pipeline. This approach minimizes technical debt and the risk of requirements inconsistency in fast-paced development environments.
***Keywords:*** system analysis; Agile; DevOps; CI/CD; synergy; contradictions; technical debt; architecture.