

DOI: 10.20535/kpissn.2021.4.261938

УДК 007: 658.5

П.П. Маслянюк*, І.В. Савчук
КПІ ім. Ігоря Сікорського, Київ, Україна
*corresponding author: mppdom@i.ua

DEVOPS – КОНЦЕПТ І СТРУКТУРНЕ ПРЕДСТАВЛЕННЯ

Проблематика. На сьогодні, поряд з величезною кількістю методів, технологій та засобів підвищення ефективності бізнесу, стрімкого розвитку набуває і застосування концепту DevOps (*Development & Operations*).

Зокрема, застосування в бізнесі концепту DevOps приводить практиків до питання доцільності застосування, формалізації цього процесу, та розробки, як правило, своєї власної системи DevOps зорієнтованої на досягнення визначених значень метрик діяльності Орг.С.

Саме тому автори цієї статті вважають, що тема наукового та прикладного обґрунтування та систематизації DevOps-концепту потребує більш ґрунтовного висвітлення не тільки з точки зору формалізації, але й з точки зору практичного продукування систем DevOps для будь якого виду продуктової діяльності Орг.С.

Мета дослідження. Метою цієї статті є дослідження суті, концепту та дефініції поняття DevOps, визначення сутностей концепту DevOps та формалізація уніфікованої моделі DevOps.

Методика реалізації. Методологія порівняльного аналізу наукових результатів та інженерних практик реалізації DevOps, систематизація класів сутностей концепту DevOps у формі діаграми Вена, графічне моделювання уніфікованої моделі DevOps.

Результати дослідження. Встановлено практичну цінність і результативність впровадження DevOps, відсутність теоретичних досліджень та обґрунтування методів інженерії систем DevOps, окрім прикладів окремих патернів та окремих архітектурних рішень DevOps для ІТ індустрії.

В статті обґрунтовано і запропоновано структурне представлення концепту DevOps у формі діаграми Вена як системи сутностей продукування, підтримки, процесів управління продукуванням і підтримки, та відношень між ними. Ці результати дозволяють обрати теоретичні і прикладні інструменти для формалізації уніфікованої моделі DevOps-систем.

Висновки. Уніфікована модель DevOps-системи у вигляді діаграми компонентів і формалізованих специфікацій інтерфейсів між цими компонентами та уніфікована модель DevOps-системи у вигляді діаграми діяльності цих компонентів дають можливість перейти до розробки обґрунтованих методів інженерії DevOps-систем.

Ключові слова: DevOps, модель DevOps, Системна інженерія DevOps, концепт DevOps, метрики DevOps, процеси DevOps, управління програмним продуктом, колаборація, конкурентна перевага.

Вступ

На сьогодні поряд із величезною кількістю методів, технологій та засобів підвищення ефективності бізнесу стрімкого розвитку набуває і концепт DevOps (*Development & Operations*).

Застосування концепту DevOps і його практична реалізація набувають наразі яскраво вираженого корпоративного характеру, тобто спрямованого на певний вид діяльності та корпоративну культуру тієї чи іншої Організаційної Системи (Орг.С) у самому широкому сенсі розуміння терміна «корпоративна культура».

Зокрема, застосування в бізнесі концепту DevOps приводить практиків до питання доцільності застосування, формалізації цього процесу та розробки, як правило, своєї власної системи DevOps, зорієнтованої на досягнення визначених значень метрик діяльності Орг.С.

У такому контексті власника бізнесу і топ-менеджмент Орг.С цікавлять насамперед повна вартість володіння такою системою DevOps і реальні прибутки від впровадження системи DevOps.

З іншого боку, ринок пропонує не так уже і багато теоретичних та прикладних розробок, на основі яких можна було б реалізувати

Рекомендуємо цитувати цю статтю так: П.П. Маслянюк, І.В. Савчук, «DevOps – Концепт і структурне представлення», *Наукові вісті КПІ*, № 4, с. 39–51, 2021. doi: 10.20535/kpissn.2021.4.261938.

Please cite this article as: P.P. Maslianko, I.V. Savchuk, “DevOps – concept and structural representation”, *KPI Science News*, no. 4, pp. 39–51, 2021. doi: 10.20535/kpissn.2021.4.261938.

фаховий проєкт системи DevOps для конкретної Орг.С.

Ця проблема виникла тому, що наукових публікацій за цим напрямом просто мізерна кількість, а у наявних мова йде переважно про статистичні дослідження впливу DevOps на бізнес у цілому або на окремі метрики Орг.С.

Саме тому автори цієї статті вважають, що тема наукового та прикладного обґрунтування та систематизації DevOps-концепту потребує більш ґрунтовного висвітлення з точки зору не тільки формалізації, але й практичного продукування систем DevOps для будь-якого виду продуктової діяльності Орг.С.

У цій статті досліджуються місце і роль DevOps для будь-якої стадії життєвого циклу продуктів, товарів та послуг, проводиться детальна декомпозиція всіх основних сутностей DevOps, визначаються ролі та вплив кожної з них.

Таким чином, ми спробуємо розкрити всі особливості феномену DevOps не тільки як засобу для організації взаємодії між організаційними одиницями Орг.С, а і як важливого й актуального інструменту для продукування систем DevOps для сучасної ІТ-індустрії.

Постановка задачі

Метою цієї роботи є дослідження суті, концепту та дефініції поняття DevOps, визначення сутностей концепту DevOps та формалізація уніфікованої моделі DevOps.

Суть концепту DevOps

Бачення суті концепту DevOps учасниками ринку та особливостей його застосування показано у багатьох публікаціях. Так, автори статті [1] детально аналізують результати застосування методології DevOps для страхової компанії.

Насамперед автори статті визначають, що *«Сенс DevOps, як висловили респонденти, розглядався як спосіб інтеграції ролей і навичок розробки та операцій ближче один до одного, щоб узгодити стимули ключових ролей, які беруть участь у розробці програмного забезпечення»* [1].

Далі вони вказують, що основна мотивація впровадження методології DevOps в організації — це отримання конкурентної переваги за рахунок застосування й автоматизації бізнес-процесів взаємодії розробників, служб підтримки та менеджменту Орг.С. Автори виділяють дві групи результатів від впровадження DevOps: групу результатів, що безпосередньо стосується конкрет-

них метрик ефективності діяльності Орг.С у цілому, та групу результатів, що стосується метрик оцінювання діяльності самої команди Орг.С.

Окрім результатів, у статті автори [1] вказують і на певні труднощі, пов'язані з впровадженням DevOps.

До першої групи результатів автори відносять такі якісні метрики:

- спільна робота команди, відсутність конфліктів і взаємний контроль, що дає можливість проводити часте/неперервне розгортання нових версій проєкту та впроваджувати нові можливості;

- можливість пришвидшувати процеси розробки та роботу над проєктом у цілому;

- покращення роботи з користувачами результатів розробки та отримання кращого досвіду від користувачів.

До другої групи результатів автори відносять позитивний вплив не тільки на продукти компанії, а й на саму команду:

- учасники команди мають більший рівень автономії та відчувають свою значущість;

- робота в команді мотивує, люди вивчають нові технології;

- учасники команди менше звинувачують один одного, оскільки їхня робота є командною.

Окрім цього, автори визначають і складнощі впровадження DevOps:

- необхідність впроваджувати нові технології та навчати персонал, зокрема інструментам хмарного хостингу, архітектури мікросервісів, а також проведення автоматизованого розгортання та моніторингу;

- необхідність залучати фахівців, знайомих із методологією DevOps.

У статті [2] автори спираються і описують традиційний цикл DevOps і зазначають, що переваги полягають у більшій кількості безперервних розгортань, розширеній автоматизації, кращій відповідності вимогам і загальних покращеннях.

У статті [3] компанія інформує про власний досвід використання методології бізнесом і всередині компанії. Це стаття-звіт щодо методології DevOps від компанії Atlassian яка випускає програмні продукти саме для впровадження методології DevOps (Jira, Bitbucket та інші). Наведені ними статистичні дані чітко вказують на позитивний вплив цієї методології на бізнес за рахунок:

- частіших релізів нових версій проєкту та його швидкого розгортання;

- більшої кількості проєктів, що закінчуються успіхом;

– швидшого налаштування роботи після виявлення несправностей.

У статті вказана і основна парадигма компанії Atlassian стосовно реалізації концепту DevOps – «краще багато маленьких оновлень, але часто, аніж велике та складне оновлення, але досить рідко» [3]. Цю парадигму компанія реалізує за рахунок автоматизації наскрізного бізнес-процесу «ти це зробив, ти це доставив, ти це запустив», тобто DevOps – «це не чиясь окрема робота, це робота усіх, хто працює в компанії» [3].

У звіті Гарвардського університету щодо практик застосування методології DevOps [4] автори зазначають, що наразі одним із ключових показників для бізнесу є якомога швидший запуск продукту з новою функціональністю для кінцевого користувача, але зараз небагато компаній роблять це дійсно добре.

Автори наводять багато інфографіки з приводу того, як оцінюють вплив методології DevOps різні компанії, абсолютна більшість із яких вважають, що вплив є значущим і позитивним.

Автори вказують на те, що впровадження методології – це насамперед проведення змін у самій компанії, управлінні, менеджменті бізнес-процесів і навіть корпоративній культурі.

Корисною є інформація від респондентів про те що, на їх думку, стримує розвиток впровадження DevOps в компанії, зокрема:

- невідповідні організаційні моменти в компанії;
- використання старих (legacy) технологій;
- супротив керівництва до змін;
- обмеженість навичок і правильного способу мислення у робітників;
- необхідність у покращенні безпеки з одночасним підвищенням рівня комунікацій.

Насамкінець автори підкреслюють, що застосування DevOps не є технологічним проектом. На їхню думку, DevOps насамперед полягає у проведенні змін відносно персоналу, організаційної структури, управління продуктивністю і навіть корпоративної культури, тобто всього того, що може фундаментально змінити організацію.

У щорічному огляді від компанії Puppet [5], яка займається випуском програмного забезпечення інструментів DevOps, встановлено, що в компаніях, де впроваджені технології та інструменти DevOps, команди розробки й підтримки мають чітко формалізовані обов'язки та високий рівень автономії щодо виконання цих обов'язків, і головне, вони мають чітко

визначені парадигми взаємодії та канали комунікації між командами [5].

Автори огляду виявили, що види труднощів на шляху впровадження DevOps змінюються від композиції технічних і культурно-організаційних до майже повністю культурних.

Автори огляду вважають, що ключем до максимально успішного рівня впровадження DevOps є застосування єдиної технологічної платформи DevOps для всіх команд компанії. Така платформа регламентує права та обов'язки всіх команд і комунікації між ними.

У статті [6] представлені результати систематичного дослідження, в якому розглянуто основні сфери, теми та цілі NFR (Non-Functional Requirements, нефункціональних вимог) у контексті DevOps, на яких фокусуються розробники при проектуванні DevOps-систем. У статті зазначається, що при дослідженні методології DevOps найчастіше зустрічається такий перелік ключових слів: *Performance, Scalability, Safety, Cross-team integration, Cross-Domain Governance, Regulatory Compliance* та багато інших.

У статті детально проаналізовано та наведено статистику по кожній із частин процесу DevOps. Наведено інформацію про те, що найбільше цікавить технічний, організаційний та бізнесовий склад компаній.

Найважливіші завдання, що можуть бути вирішені за допомогою NFR у DevOps, – це забезпечити безпечну та продуктивну роботу програмного забезпечення (зосередження на технічній якості), сприяти міжкомандній інтеграції експертів із різних галузей (організаційний фокус), з'ясувати відповідність програмного забезпечення нормативним вимогам (фокус на управлінні), використовувати взаємодію клієнта з програмним забезпеченням для ідей щодо перспективних функцій (зосередженість на клієнті), створювати та захищати частку ринку продукту (бізнес-фокус), оптимізувати конвеєр доставки (фокус на розробці) та полегшувати налагодження після розгортання (зосередження на роботі) [6].

Однією з ключових складових DevOps є колаборація. Автори статті [7] дають таке практичне визначення цього терміна: «Колаборація – це здатність людини ефективно робити свій внесок у роботу команди. Це передбачає наполегливість, внесок у знання команди, оцінку внеску інших та здатність до вирішення розбіжностей. Ефективна колаборація передбачає розподіл праці між учасниками, які ведуть активний дискурс, що приводить до об'єднання їхніх зусиль» [7].

У статті [8] автори роблять акцент на тому, що саме в DevOps вважається важливим і найчастіше цитованим: Automated Build, Continuous Deployment, Continuous Integration, Continuous Delivery, Cloud Computing, Source Code Version Control, Automated and Continuous Feedback, Infrastructure as Code.

Автори роблять висновок, що незважаючи на те що в статтях, де згадують DevOps, мова найчастіше йде про програмне забезпечення, все ж не всім вдається вдало застосовувати методологію DevOps, оскільки це не просто технологічний підхід до написання програмного коду, а створення певної моделі культури взаємодії в організації.

Автори наголошують на тому, що результати застосування DevOps проявляються різними формами покращень і перспективами для організацій, які розробляють програмне забезпечення, оскільки його впровадження є чимось трансформуючим для всієї компанії, оскільки залучаються люди, процеси та інструменти. Крім того, впровадження DevOps потребує реструктуризації корпоративної культури, набуття нових навичок, тіснішої співпраці та колаборації команд. Автори підкреслюють, що це нетривіальні задачі, оскільки впровадження DevOps вимагає дотримання низки вимог і дуже добре розробленого та структурованого плану [8].

Важливими є результати, викладені у статті [9], де автори пропонують таксономію рівнів досконалості реалізації DevOps і розглядають декілька патернів організації роботи команд DevOps, починаючи від найновіших, заснованих на досконалій взаємодії між командами Dev і Ops, до більш архаїчних, що не мають достатнього рівня комунікацій між командами.

Таксономія є результатом ґрунтового дослідження, в якому проаналізовано 31 багатонаціональну компанію, що спеціалізуються на розробці програмного забезпечення.

Запропонована таксономія може бути застосована для обґрунтування і реалізації конкретних проєктів DevOps та оцінювання поточного технологічного рівня зрілості DevOps компанії.

Поряд із коротким історичним оглядом методології DevOps, у статті [10] автори роблять огляд важливих складових DevOps і наводять діаграми, що характеризують процес DevOps-розробки. Автори наголошують на тому, що «DevOps – це нова оптимальна практика, якої слід дотримуватися в життєвому циклі розробки програмного забезпечення, щоб збільшити кількість випусків,

надійність, швидкість оновлення, ефективність використання відгуків клієнтів, підвищити якість і безпеку» [10].

Автори підкреслюють, що процес розробки DevOps будь-якої архітектури є нетривіальною задачею, має ітеративний характер, і наголошують, що Хмара є ключовою сутністю DevOps, призначеної для інтеграції команд, безперервної доставки, забезпечення безпеки та збору відгуків від користувачів. Насамкінець автори наголосили, що команда DevOps має бути достатньо кваліфікованою та гнучкою для проведення змін корпоративної культури.

Привертає увагу і стаття [11] із надзвичайно широкою географією і статистикою та висновками про те, як краще запроваджувати DevOps, чи потрібно її взагалі запроваджувати, чи привносить впровадження DevOps якийсь позитивний вплив на результати діяльності компанії.

Автори наводять перелік конкретних практик і принципів Agile, Lean і DevOps, які розглядаються як розширення підходів та інструментів реалізації взаємодії команд DevOps.

Автори встановили, що застосування практик і принципів Agile, Lean і DevOps дає змогу змінювати корпоративну культуру, відстежувати поточний стан стадій життєвого циклу програмного забезпечення та керувати станом розробки засобами автоматизації DevOps.

А застосування SCRUM, Kanban та Continuous Delivery забезпечує доставку, реалізацію процесів управління змінами, управління портфелем послуг (включаючи керування каталогом послуг), керування випуском і розгортанням та керування рівнем обслуговування.

Автори наголошують і на тому, що поряд із високим рівнем підготовки команди DevOps, має бути такий же рівень і керівника DevOps, який повинен працювати на постійній основі [11].

Серед публікацій, що стосуються суті концепту DevOps, є роботи, які описують метрики оцінювання результатів застосування DevOps.

Зокрема, автори статті [6] проаналізували наявні дослідження в галузі DevOps для того, щоб визначити основні вимоги до цієї методології, тобто було проведено аналіз *нефункціональних вимог*. Для цього методологію DevOps розділили на сім доменних областей:

– **Customer Focus Area** (область зосередженості клієнта) – вона концентрується на цінностях, які допомагають покращити досвід клієнта та відслідковувати покращення. Наведемо перелік орієнтованих на клієнта цінностей: *ідеї, які надходять від клієнтів для покращення бізнесу;*

перевірка заявленим вимогам; оцінювання важливості використання клієнтом; частота релізів; пріоритезація циклів розробки, орієнтованих на клієнта.

– **Development Focus Area** (область зосередженості розробки) – в основному тут йдеться про орієнтири на підвищення ефективності процесу розробки. Перелік орієнтованих на розробку цінностей: *вивірність процесу доставки; неперервна інтеграція; неперервна доставка; широкий вибір інструментів розробки; складність задань розробки.*

– **Operation Focus Area** (область зосередженості операцій) – тут йдеться про вимоги, які ставляться до програмного забезпечення і які мають виконуватись під час проведення операцій стосовно нього. Перелік цих вимог: *відслідковування артефактів під час виконання; моніторинг процесу зростання продукту; перевірка продукту після його розгортання.*

– **Governance Focus Area** (область зосередженості управління) – в цій площині розглядаються цінності, які є важливими для процесів управління і які мають бути дотримані процесом DevOps-розробки. Перелік цих цінностей: *забезпечення конфіденційності даних; виконання нормативних вимог; дотримання корпоративної архітектури.*

– **Technical Quality Focus Area** (область зосередженості забезпечення технічної якості) – ця область повністю концентрується на вимогах до продукту, які він має задовольняти, наприклад такі, як час виконання операцій або затрати на використання пам'яті в базі даних. Наведемо розгорнутий перелік цих цінностей і вимог: *зручність використання; стійкість продукту; можливість до залучення нових ресурсів; потужність; використання актуальних фреймворків; безпека даних; можливість розширення продукту; якість поставлених тестів; можливість до подальшого обслуговування продукту; розгортання продукту, орієнтоване на роботу з даними; безпека використання.*

– **Organization Focus Area** (область зосередженості організації) – тут йдеться про цінності та вимоги, які має задовольняти компанія для продукування орієнтованих на бізнес продуктів. Наведемо перелік: *своєчасна реакція на зміну ринку; текучість ресурсів; мотивація для продукування продуктів; комунікація команд в організації; інтеграція команд в організації; можливості для підвищення професійного рівня; охоплення різних доменних областей.*

– **Business Focus Area** (область зосередженості бізнесу) – ця область покриває класичні

вимоги бізнесу, адже його цікавить не те, як буде зроблено продукт, а те, яку цінність він у собі несе. Наведемо перелік таких цінностей і вимог: *середній час до отримання результату; вартість провадження; цінність створеної екосистеми; яку частку ринку займає продукт; актуальність продукту на ринку.*

Таким чином, у цій статті фактично описані класи метрик, необхідних для оцінювання результатів впровадження концепту DevOps.

У статті [12] автор аналізує метрики якості для успішного запуску програмного продукту. Метрики дають можливість визначити, чи буде користь від розробки продукту. Серед таких метрик:

– окупність інвестицій – тобто чи буде приносити продукт дохід від вкладених інвестицій та за який час;

– скільки ідей було згенеровано для того, щоб задовольнити потреби користувача;

– прогнозована комерційна цінність;

– скільки часу потрібно, щоб вийти на ринок;

– яку частину ринку буде займати продукт.

У публікації [13] зазначають важливі метрики оцінювання продукту, який вже працює, наприклад такі як:

– щомісячний регулярний дохід;

– середній дохід за одного користувача;

– вартість залучення нових клієнтів;

– відношення кількості активних клієнтів за день до кількості активних клієнтів за місяць;

– показник утримання клієнтів.

Важливими є і метрики, що оцінюють ефективність роботи команди DevOps. Так, у статті [14] автори вказують на те, що наразі існують чотири найпопулярніші ключові метрики, **Four Key Metrics (FKM)**, для визначення ефективності роботи DevOps і надають фреймворк для автоматизації вимірювання цих метрик. Потрібно зазначити, що ці метрики підходять тільки для IT-продуктів.

1. **Deployment Frequency** (частота розгортань) – ця метрика направлена на те, щоб прискорити до зменшення нагромадження в проєкті та прискорювати доставку нових функцій продуктів до кінцевих користувачів.

2. **Lead Time for Change** (час виконання змін) – пояснюється як кількість часу, затраченого на перехід від готового та перевіреного коду до успішно розгорнутого продукту на ринку. Ця метрика вважається важливою тому, що в разі несправності у продукті дає змогу мінімізувати час виправлення проблем.

3. **Time to Restore Service** (час відновлення сервісу) – ця метрика важлива тим, що у великих системах є невідворотним факт того, що будуть з'являтися помилки, і тому дуже важливо мінімізувати час відновлення роботи сервісу.

4. **Change Failure Rate** (рівень змін, які призвели до відмов) – це відсоток нововведень у програмне забезпечення, які призводять до погіршення функціональності або навіть відмови та потребують негайного виправлення (так звані hot-fix).

Таким чином, автори цієї статті пропонують чотири метрики для оцінювання власне роботи команди DevOps з організації розробки та експлуатації програмних продуктів.

На нашу думку, дослідження й аналіз наведених наукових і практичних результатів бачення суті концепту DevOps учасниками ринку дає змогу зробити декілька важливих узагальнювальних висновків.

1. Насамперед звертаємо увагу на множині ключових слів, частина з яких покладена авторами статей в основу онтологічних моделей предметної області DevOps. Зазначимо, що перелік цих ключових слів не є вичерпним!

Ключові слова: DevOps; DevOps system; DevOps model; DevOps adoption; DevOps life cycle; IT product life cycle; DevOps culture; System Design; System Engineering; System Analysis; колаборація; комунікація; неперервна інтеграція; розгортання; автоматизація; моніторинг; тестування; вимірювання метрик; швидкість розробки; гнучкість; конкурентна перевага; cross-team integration; cross-domain governance.

Тут іменники-сутності, іменники-метрики, дієслова-процеси.

2. Команди, тобто організаційні одиниці Орг.С, що задіяні у бізнес-процесах виробництва, підтримки та менеджменту, працюють як складові спільної моделі діяльності Орг.С на всіх стадіях життєвого циклу продуктів, товарів і послуг, у тому числі й розробки програмного забезпечення.

3. Спільна модель діяльності Орг.С накладає і спільну відповідальність за виконання всієї роботи.

4. Спільна модель діяльності Орг.С і спільна відповідальність за виконання всієї роботи надає можливість спільно контролювати та керувати бізнес-процесами діяльності Орг.С.

5. Спільна модель діяльності Орг.С, спільна відповідальність за виконання всієї роботи, спільний контроль і керування бізнес-процесами дає змогу формувати спільне та узгоджене

бачення продукту, його дизайн, конкретну імplementацію та сценарії використання.

6. Спільна модель діяльності Орг.С, спільна відповідальність за виконання всієї роботи, спільний контроль і керування бізнес-процесами, спільне та узгоджене бачення продукту, його дизайн, конкретна імplementація та сценарії використання дають змогу забезпечити можливість для частішого/неперервного розгортання нових версій проекту, проводити аналіз поточних версій і вчасно вносити необхідні зміни.

7. Незважаючи на практичну цінність і результативність впровадження DevOps, привертає увагу відсутність у публікаціях теоретичних досліджень та обґрунтування методів інженерії систем DevOps, окрім прикладів окремих патернів та окремих архітектурних рішень DevOps для IT-індустрії.

8. Більшість авторів підкреслюють, що DevOps є по суті системою автоматизації бізнес-процесів колаборації команд (організаційних одиниць) Орг.С, але при цьому не демонструє DevOps як систему.

9. Оскільки DevOps є по суті системою автоматизації бізнес-процесів колаборації команд (організаційних одиниць) Орг.С, то, на нашу думку, мають бути формалізовані, зокрема, й класи метрик для оцінювання результатів автоматизації бізнес-процесів колаборації.

9.1. Клас метрик оцінювання результатів колаборації бізнес-процесів інженерії продуктів, товарів та послуг [14]:

- частота розгортань (deployment frequency);
- час виконання змін (lead time to change);
- час на відновлення сервісу (time to restore service);
- рівень змін, які призвели до відмов (change failurerate).

9.2. Клас метрик оцінювання результатів колаборації бізнес-процесів підтримки продуктів, товарів і послуг [12, 13]:

- вартість залучення нових клієнтів;
- відношення кількості активних клієнтів за день до кількості активних клієнтів за місяць;
- показник утримання клієнтів;
- кількість згенерованих ідей для задоволення потреб користувача.

9.3. Клас метрик оцінювання результатів колаборації бізнес-процесів управління інженерією та підтримкою продуктів, товарів і послуг на всіх стадіях їх життєвого циклу [12, 13]:

- окупність інвестицій;
- прогнозована комерційна цінність;
- яку частину ринку буде займати продукт;

- скільки потрібно часу, щоб вийти на ринок;
- щомісячний регулярний дохід;
- середній дохід за одного користувача;
- вартість залучення нових клієнтів.

10. На наш погляд, дослідження та розробка методів і моделей інженерії систем DevOps можуть бути корисними і затребуваними для імплементації систем DevOps на всіх стадіях їх життєвого циклу. В будь-якому разі наявність методів і моделей інженерії систем DevOps могла б систематизувати й упорядкувати як процес розробки, так і процеси експлуатації систем DevOps.

DevOps – дефініція та структурне представлення

Для фахової дискусії проведемо аналіз дефініції терміна «DevOps» на основі існуючих поглядів на цю методологію, насамперед у наукових статтях.

Так, автори статті [1] наводять таке визначення терміна «DevOps»:

«DevOps – це набір можливостей інженерних процесів, які підтримуються культурними та технологічними факторами. Можливості визначають процеси, які організація має бути спроможна виконати, в той час як засоби реалізації забезпечують плавність, гнучкість та ефективність роботи».

Це визначення автори підкріплюють переліком елементів множин можливостей і способів їх реалізації інструментами DevOps для Орг.С ІТ-індустрії. Зокрема, до переліку можливостей відносяться:

- колаборативна та неперервна розробка;
- неперервна інтеграція та тестування;
- неперервний випуск і розгортання продукту;
- неперервний моніторинг інфраструктури та оптимізація;
- постійне відслідковування поведінки користувача та реакція на неї;
- відновлення роботи сервісів без затримок;
- неперервне вимірювання показників.

А як способи реалізації цих можливостей зазначають:

- автоматизацію збирання проекту;
- автоматизацію тестування;
- автоматизацію розгортання;
- автоматизацію моніторингу;
- автоматизацію реакції на несправності;
- менеджмент конфігурацій для кодування та інфраструктури;

- автоматизацію проведення вимірювань.

Розглянемо, як автори статті [2] дають визначення терміна «DevOps»:

«DevOps – це підхід, який поєднує традиційні ролі розробників програмного забезпечення та людей, які працюють над покращенням комунікацій у проекті задля підвищення частоти розгортання проекту та підтримки якості програмного забезпечення».

Це визначення є більш конкретним і торкається конкретних понять процесу DevOps-розробки. Серед наборів інструментів, якими оперує DevOps, автори виділяють:

- неперервну інтеграцію;
- неперервне розгортання проекту;
- неперервний моніторинг, насамперед програмного забезпечення;
- контроль версій;
- неперервне тестування.

Автори документу [3] намагаються описувати DevOps максимально просто, щоб донести основну суть цього процесу, і, як вони вважають, у DevOps фактором успіху № 1 є командна робота. В документі наголошується, що DevOps не є роботою якоїсь конкретної людини – це робота всіх працівників, які мають відношення до проекту. Саме за тісної, регулярної взаємодії формується культура праці в колективі, з'являється колективна відповідальність і, як наслідок, бажання брати активну участь в усіх етапах реалізації проекту. Це і декларується у визначенні в їхній інтерпретації:

«**DevOps** – це процес, у якому команди розробки та різних ІТ-операцій працюють разом. Він спрямований на покращення комунікації та колаборації таким чином, щоб процеси розробки, тестування, випуску та використання програмного продукту могли відбуватися швидше та надійніше».

У наступних працях автори описують DevOps з огляду на те, як це поняття характеризують респонденти в опитуваннях і що ці респонденти вважають найбільш вагомими частинами.

У статі [4] автори цікавились, що відзначають люди, які працювали з DevOps. За проведеним опитуванням, 86 % респондентів вважають головною можливістю в DevOps те, що за допомогою цієї методології відбувається пришвидшення розробки та випуску проектів. Компанії вважають, що DevOps надає їм конкурентну перевагу в таких аспектах, як:

- швидкість виходу нового проекту та функцій для кінцевого користувача;
- продуктивність розробки;

– якість розроблюваних продуктів і сервісу в цілому;

– задоволення роботою в командах, що займаються проєктом;

– швидкість впровадження інновацій.

І на основі цього автори доходять висновку, що «DevOps об'єднує команди розробки та операцій для підвищення швидкості та якості продукту, при цьому поєднуючи розробку, інтеграцію, тестування, випуск, моніторинг та управління».

У статті [5] компанія Puppet звітує про DevOps, як «про поняття, яке, можливо, ще не до кінця визначене». Як зазначають автори, ця методологія наразі є дуже популярною серед компаній і зазвичай комбінується з використанням гнучких підходів до розробки, таких як, наприклад, Agile. Окрім цього, автори додають, що залежно від специфіки задачі DevOps може гнучким чином підлаштовуватись під будь-яку предметну область. З урахуванням цього в статті наводяться три визначення того, що таке DevOps, а саме:

– «DevOps – це культурна парадигма та своєрідна об'єднана множина обов'язків, відповідно до цього кожен розробник також несе відповідальність за роботу операцій, а кожен, хто відповідальний за операції, – за участь у розробці»;

– «DevOps – це культурне зміщення, покликане покращити комунікації, колаборацію, інтеграції та автоматизацію в рамках всієї організації»;

– «DevOps – це об'єднання команди, що займається інфраструктурними рішеннями, з командою, що займається розробкою програмного забезпечення, в результаті якого вони разом займаються управлінськими рішеннями та навіть створенням нової інфраструктури».

Зазвичай, коли говорять про DevOps, мають на увазі таку множину сутностей, яку можна представити у вигляді діаграми Венна (див. [15]).

На нашу думку, дуже вдале означення концепту DevOps відносно ІТ-індустрії, наведено в роботі [16]: «DevOps – це набір практик, інструментів та культури, яка автоматизує та інтегрує процеси між командами розробки та командами ІТ-планування. DevOps спеціалізується на тому, що підвищує ефективність, комунікації, співпрацю та автоматизацію роботи ІТ-команд» [1]. Важливо те, що це визначення містить основні класи сутностей, необхідних для розуміння суті концепту DevOps, та суттєво доповнює представлення концепту DevOps у вигляді діаграми Венна (див. [15]).

Зокрема, це сутності інженерії продуктів, товарів і послуг, сутності бізнес-процесів під-

тримки продуктів, товарів і послуг та сутності колаборації бізнес-процесів управління інженерією та підтримкою продуктів, товарів і послуг на всіх стадіях їх життєвого циклу.

Аналізуючи всі ці визначення терміна «DevOps», можна помітити, що більшість авторів відзначають приблизно одне і те ж розуміння сутності концепту DevOps із деякими особливостями, які, швидше за все, обумовлені практичним досвідом авторів цих визначень.

На основі цього огляду виділимо набір слів, якими можна найточніше описати семантику означення концепту DevOps, а саме: *культурна парадигма, колаборація, комунікація, неперервна інтеграція, розгортання, автоматизація, моніторинг, тестування, вимірювання метрик, швидкість розробки, гнучкість, конкурентна перевага*.

Узагальнюючи результати цього короткого системного аналізу сутностей означення концепту DevOps, можна зробити висновок, що мова йде насамперед про модель системи організації процесів розробки та підтримки, принципи якої полягають у застосуванні певних наукових і практичних підходів для об'єднання наявних ресурсів та інтеграції роботи команд розробки й підтримки впровадження.

Концепт DevOps як система інтегрованих міждисциплінарних ресурсів колаборації бізнес-процесів управління інженерією та підтримкою продуктів, товарів та послуг на всіх стадіях їх життєвого циклу

Спираючись на результати системного аналізу сутностей представлення концепту DevOps у вигляді класичної діаграми Венна, дамо авторське визначення концепту DevOps як системи сутностей і відношень між ними. Для цього спочатку формалізуємо складові діаграми Венна з урахуванням класів сутностей предметної області DevOps та дамо їх чітке визначення.

На наш погляд, на підставі результатів аналізу предметної області DevOps, представлення концепту DevOps складається з трьох класів сутностей:

– Множини сутностей інженерії продуктів, товарів та послуг – Entities of Product Development, Set D;

– Множини сутностей підтримки продуктів, товарів та послуг – Entities of Product Support, Set S;

– Множини сутностей управління інженерією та підтримкою продуктів, товарів та послуг – Entities of Support and Engineering Control, Set P.

Далі дамо авторське визначення класів сутностей представлення предметної області концепту DevOps у вигляді діаграми Венна (рис. 1).

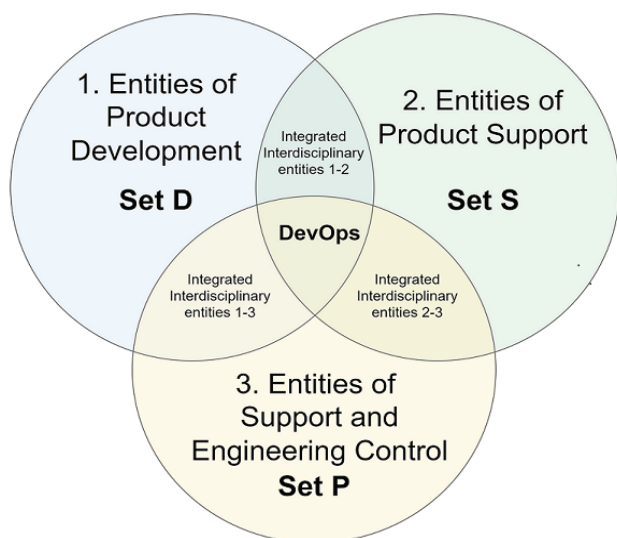


Рис. 1. Представлення класів сутностей концепту DevOps у вигляді діаграми Венна

1. Entities of Product Development, Set D – Множина сутностей інженерії продуктів, товарів та послуг для всіх стадій їх життєвого циклу визначається як множина D , $d \in D$, усіх наявних на ринку матеріальних і нематеріальних сутностей, що можуть бути прямо та/або опосередковано задіяні для інженерії визначеного класу, або об'єкта класу продуктів, товарів та послуг на конкретній стадії їх життєвого циклу.

2. Entities of Product Support, Set S – Множина сутностей підтримки продуктів, товарів та послуг для всіх стадій їх життєвого циклу визначається як множина S , $s \in S$, усіх наявних на ринку матеріальних і нематеріальних сутностей, що можуть бути прямо та/або опосередковано задіяні для підтримки життєдіяльності визначеного класу, або об'єкта класу продуктів, товарів та послуг на конкретній стадії їх життєвого циклу.

3. Entities of Support and Engineering Control, Set P – Множина сутностей управління інженерією та підтримкою продуктів, товарів та послуг для всіх стадій їх життєвого циклу визначається як множина P , $p \in P$, усіх наявних на ринку матеріальних і нематеріальних сутностей, що можуть бути прямо та/або опосередковано задіяні для досягнення мети управління інженерією та підтримкою життєдіяльності продуктів, товарів та послуг визначеного класу, або об'єкта

класу продуктів, товарів та послуг на конкретній стадії їх життєвого циклу.

Наприклад. Development processes – процеси, в ході яких відбуваються розробка, тестування, розгортання проєкту. Включають в себе: кодування; збирання проєкту; неперервне тестування, розгортання, моніторинг, інтеграцію коду; збір артефактів; контроль версій.

Наприклад. Support processes – у цей термін вкладається поняття про те, що проєкт має бути постійно під наглядом, неперервно розгортатися, випускатися, тестуватися, має здійснюватися регулярний контроль за якістю та відповідністю поставленим вимогам. Окрім цього, збирається зворотній зв'язок від користувачів продукту, і формується відповідна реакція на нього.

Наприклад. IT Operations – це найбільш широке поняття в DevOps. Воно включає як постійні корпоративні зустрічі між різними ланками співробітників, так і відповідальність за всі інші види комунікацій на проєкті. Крім цього, операції мають виконуватися із залученням робітників із різними вміннями: наприклад, у розгортанні проєкту мають брати участь як розробники, так і працівники, які займаються тестуванням, та люди, які напряму співпрацюють із бізнесом.

На діаграмі Венна на рис. 1 перетини множин 1-2, 1-3 та 2-3 класів сутностей D , S та P утворюють інтегровані міждисциплінарні ресурси (сутності), виділені за ознакою наявності функціональних відношень між ресурсами (сутностями) різних типів, зокрема Set D, Set S, Set P.

Integrated Interdisciplinary Entities 1-2 (ІЕ 1-2) – це наявні сутності розробки та підтримки, що дають можливість збільшувати або зменшувати потужності проєкту за потреби.

Integrated Interdisciplinary Entities 2-3 (ІЕ 2-3) – це наявні сутності, що дають можливість оптимізувати необхідний рівень підтримки продуктів, товарів та послуг на всіх стадіях їх життєвого циклу, постійно випускати нові версії проєкту, неперервно його тестувати та проводити моніторинг.

Integrated Interdisciplinary Entities 1-3 (ІЕ 1-3) – наявні сутності, що дають можливість оптимізувати процеси продукування продуктів, товарів та послуг на всіх стадіях їх життєвого циклу.

Означення 1. Тут, і надалі у цій статті, інтегровані міждисциплінарні ресурси (сутності) ІЕ 1-2, ІЕ 2-3 та ІЕ 1-3 – це підмножини, що формуються на основі наявності властивостей інтеграції/інтероперабельності між елементами множин D , S та P . А інтеграція/інтероперабельність

елементів множин D , S та P полягає у здатності реалізувати колаборації пар сутностей множин $D i S$, $S i P$, $D i P$.

Тоді для трьох множин – складових DevOps: формалізованих ресурсів (сутностей) інженерії – D ; сутностей інженерії і підтримки життєдіяльності продуктів, товарів та послуг для всіх стадій їх життєвого циклу – S ; сутностей управління інженерією та підтримкою продуктів, товарів та послуг для всіх стадій їх життєвого циклу – P , формалізуємо означення DevOps на основі оновленої діаграми Венна (рис. 1) та наявності властивостей інтеграції/інтероперабельності між елементами цих множин.

Означення 2. Концепт DevOps може бути формалізований як множина трійок елементів міждисциплінарних ресурсів (сутностей) із трьох множин ресурсів (сутностей): сутностей інженерії продуктів, товарів та послуг для всіх стадій їх життєвого циклу – D , сутностей підтримки продуктів, товарів та послуг для всіх стадій їх життєвого циклу – S та сутностей управління інженерією та підтримкою продуктів, товарів та послуг для всіх стадій їх життєвого циклу – P таких, що мають спільну властивість/ознаку, яка забезпечує інтеграцію/інтероперабельність між елементами $d \in D$, $s \in S$, $p \in P$ і встановлює відповідності колаборації C_{dspd} і C_{dpsd} між елементами цих множин.

Тобто

$$\begin{aligned} \text{DevOps} = \{ & d, s, p, d \mid \forall d \in D, \forall s \in S, \\ & \forall p \in P, (d, s, p, d) \in C_{dspd} \} \cup \\ & \cup \{ d, p, s, d \mid \forall d \in D, \forall p \in P, \\ & \forall s \in S, (d, p, s, d) \in C_{dpsd} \}, \end{aligned} \quad (1)$$

або

$$\begin{aligned} \text{DevOps} = & C_{dspd} \cup C_{dpsd} = \\ = \{ & \text{кортеж}(D, S, P, D, G_{dspd}) \cup \\ & \cup \text{кортеж}(D, P, S, D, G_{dpsd}) \}, \end{aligned} \quad (2)$$

де: G_{dspd} і G_{dpsd} – графіки/діаграми/матриці відповідностей колаборації C_{dspd} і C_{dpsd} .

Таким способом ми обґрунтуємо структурне представлення DevOps оновленої діаграми Венна за допомогою наявності властивості/ознаки відповідності, що забезпечує інтеграцію/інтероперабельність між елементами $d \in D$, $s \in S$, $p \in P$ і встановлює відповідності колаборації C_{dspd} і C_{dpsd} між елементами множин ресурсів D , S і P оновленої діаграми Венна (рис. 1).

Така декомпозиція класів сутностей D , S і P та узагальнене системне представлення кола-

борації елементів множин цих сутностей представлення концепту DevOps (1, 2) дає змогу систематизувати існуючі означення та формалізувати дефініцію концепту «DevOps» як колаборацію сутностей інженерії, сутностей підтримки та сутностей управління інженерією та підтримкою продуктів, товарів та послуг на всіх стадіях їх життєвого циклу.

Така декомпозиція і систематизація дає нам глибше уявлення того, що таке концепт DevOps, які завдання він вирішує і, в той же час, показує, наскільки широкою є область знань, що застосовується для організації виробництва продуктів, товарів та послуг протягом їх життєвого циклу.

Це дає нам можливість узагальнити поняття «концепт DevOps» та сформулювати означення «DevOps-система».

Означення 3. Концепт DevOps – міждисциплінарна методологія управління колаборацією сутностей інженерії та підтримки життєдіяльності продуктів, товарів та послуг на всіх стадіях їх життєвого циклу.

Виходячи з усього викладеного вище, можна формалізувати поняття DevOps як системи та перейти до поширення цього поняття не тільки на розробку як процес, а і на активний бізнес-процес на кожній зі стадій життєвого циклу продуктів, товарів та послуг.

Означення 4. DevOps-система, або система DevOps, – це множина сутностей і відношень між ними, необхідна і достатня для забезпечення колаборації сутностей інженерії та підтримки життєдіяльності продуктів, товарів та послуг на всіх стадіях їх життєвого циклу за заданих умов і обмежень.

Тоді на підставі результатів такого системного аналізу можна формалізувати структурне та динамічне представлення уніфікованої моделі системи DevOps.

Зокрема, структурне представлення уніфікованої моделі DevOps-системи у вигляді діаграми компонентів у нотації UML показано на рис. 2.

Опишемо кожен з інтерфейсів уніфікованої моделі DevOps-системи у вигляді діаграми компонентів:

Dec-to-Dev – інтерфейс, який слугує для того, щоб передати інформацію про цілі бізнесу, бажаний продукт, вимоги до нього та поставлені задачі розробникам.

Dec-to-Ops – інтерфейс передавання інформації про цілі бізнесу, бізнес-вимоги, бажаний отриманий результат, важливі сфери бізнесу та поставлені задачі команді операцій.

DevOps Система

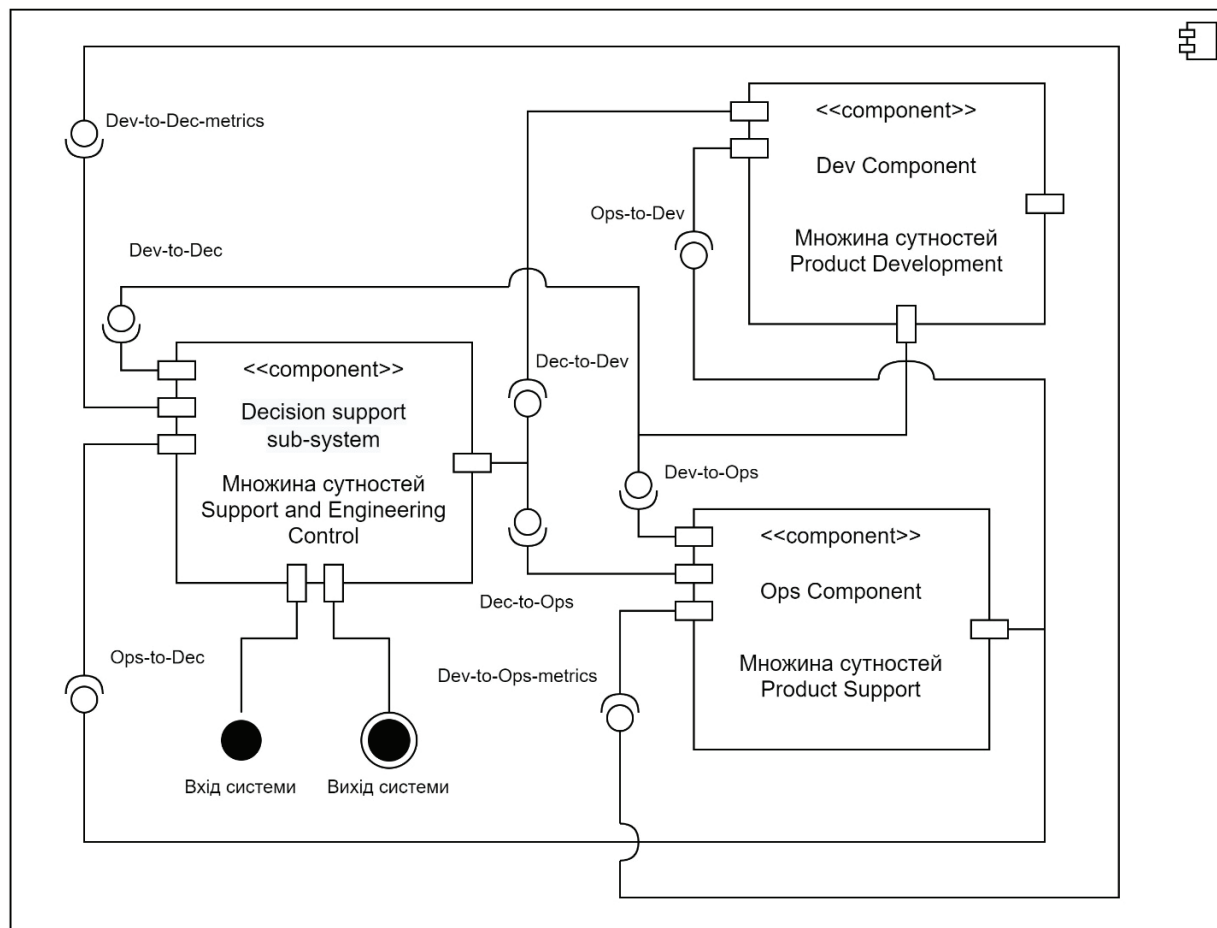


Рис. 2. Уніфікована модель DevOps-системи. Діаграма компонентів у нотатції UML

Dev-to-Dec – інтерфейс передавання інформації від команди розробників до системи управління, а саме: прогрес, отримані результати, затребувані ресурси тощо.

Dev-to-Ops – інтерфейс передавання інформації від команди розробників до команди операцій, а саме: отримані артефакти, програмний код для валідації, документація, зворотній зв'язок тощо.

Ops-to-Dev – інтерфейс передавання зворотного зв'язку та нових задач від команди операцій до команди розробників.

Ops-to-Dec – інтерфейс передавання інформації про прогрес на проекті, затребувані ресурси, уточнення бізнес-вимог системі управління.

Dev-to-Dec-metrics – інтерфейс передавання інформації про впровадження продуктів, їх потужність, зовнішній вигляд продуктів, відповідність продукту реаліям ринку.

Ops-to-Dec-metrics – інтерфейс передавання інформації про впровадження продуктів, відсоток покриття бізнес-вимог, відсоток покриття вимог щодо функціональності продукту, основні показники ефективності (Key Performance Indicators, KPI) продукту.

А динамічне представлення уніфікованої моделі DevOps-системи у вигляді діаграми діяльності в нотатції UML має вигляд, показаний на рис. 3. Діаграма діяльності формалізує колаборацію компонентів уніфікованої моделі DevOps-системи.

Уніфікована модель DevOps-системи у вигляді діаграми компонентів і формалізованих специфікацій інтерфейсів між цими компонентами та уніфікована модель DevOps-системи у вигляді діаграми діяльності цих компонентів дають можливість перейти до розробки обґрунтованих методів інженерії DevOps-систем.

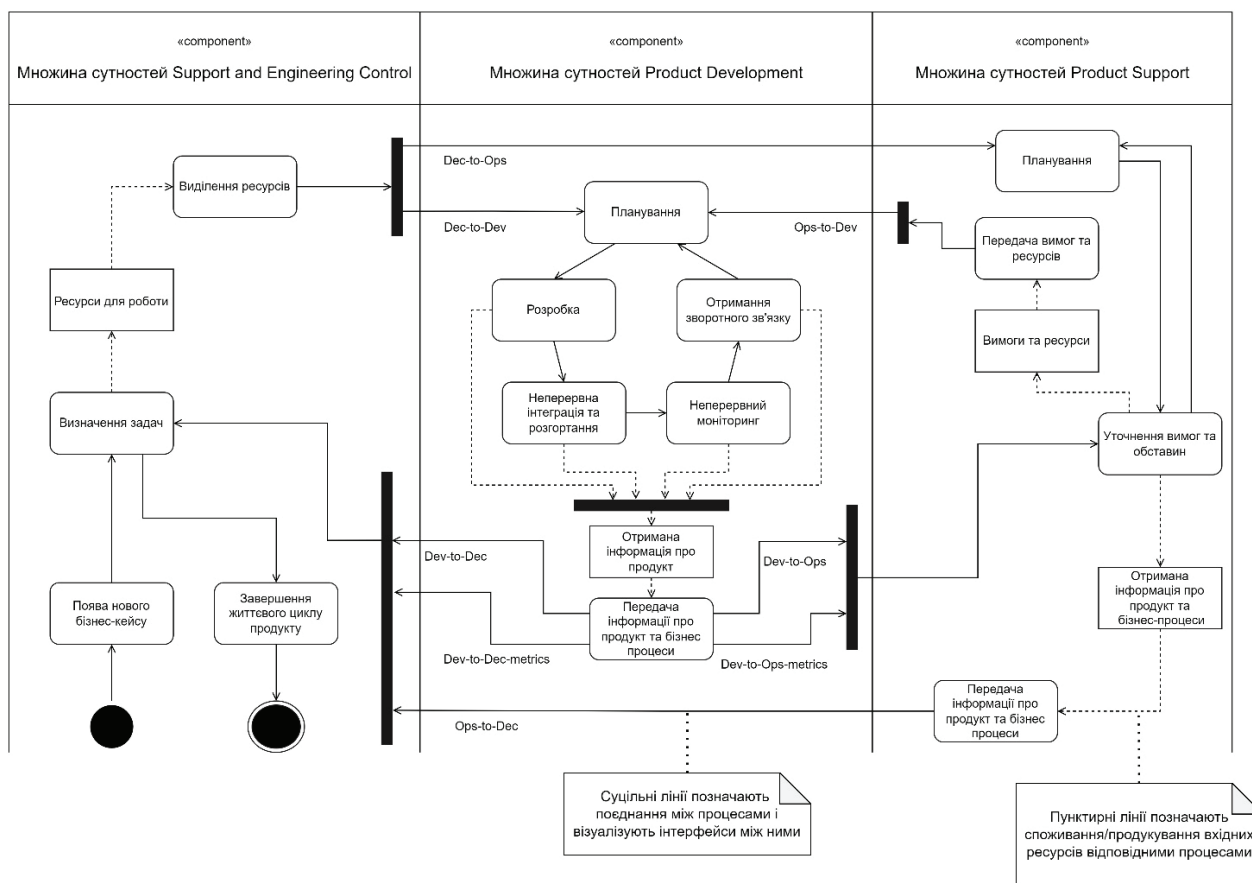


Рис. 3. Уніфікована модель DevOps-системи. Діаграма діяльності у нотатції UML

Висновки

1. У результаті аналізу наукової літератури та кращих інженерних практик встановлено, що незважаючи на практичну цінність і результативність впровадження DevOps, у наявних публікаціях відсутні теоретичні дослідження та обґрунтування методів інженерії систем DevOps, окрім прикладів окремих патернів та окремих архітектурних рішень DevOps для ІТ-індустрії.

Більшість авторів підкреслюють, що DevOps є по суті системою автоматизації бізнес-процесів колаборації команд (організаційних одиниць) Орг.С, але при цьому не розглядають DevOps як систему.

2. У статті обґрунтовано і запропоновано структурне представлення концепту DevOps у формі діаграми Венна як системи сутностей продукування, підтримки, процесів управління продукуванням і підтримки та відношень між ними. Ці результати дають можливість вибрати теоретичні та прикладні інструменти для формалізації уніфікованої моделі DevOps-систем.

3. Запропонована уніфікована модель DevOps-системи може бути застосована для розробки обґрунтованих методів інженерії DevOps-систем, призначених для управління колаборацією сутностей інженерії та підтримки життєдіяльності продуктів, товарів та послуг на всіх стадіях їх життєвого циклу.

References

- [1] M. Senapathi, J. Buchan, H. Osman, "DevOps capabilities, practices, and challenges: Insights from a case study", in *Proc. 22nd Int. Conf. Eval. Assess. Software Eng.*, 2018.
- [2] P. Jha, R. Khan, "A review paper on DevOps: Beginning and more to know", *Int. J. Comp. Applic.*, vol. 180, no. 48, pp. 16–20, 2018. doi: 10.5120/ijca2018917253

- [3] DevOps 101 with Atlassian [Online]. Available: <https://www.atlassian.com/blog/wp-content/uploads/devops-101-atlassian.pdf>
- [4] “Competitive Advantage through DevOps”, Harvard Business Review Analytic Service, Res. Rep., 2019.
- [5] “2021 Puppet State of DevOps”, CircleCI, Report, 2021.
- [6] P. Haindl, R. Plusch, “Focus areas, themes, and objectives of non-functional requirements in DevOps: A systematic mapping study”, in *46th Euromicro Conf. Software Eng. Advanc. Applic.*, Portoroz, Slovenia, pp. 394–403, 2020. doi: 10.1109/SEAA51224.2020.00071
- [7] C. Scoular, D. Duckworth, J. Heard, D. Ramalingam, “Collaboration: Definition and structure”, The Australian Council for Educational Research, report, 2020.
- [8] M.L. Pedra, M.F. Silva, L.G. Azevedo, “DevOps adoption: Eight emergent perspectives”, ArXiv abs/2109.09601, 2021.
- [9] D. Lypez-fernández, J. Díaz, J.G. Martín, J.E. Pérez, B. González-Prieto, “DevOps team structures: Characterization and implications”, ArXiv abs/2101.02361, 2021.
- [10] M. Gokarna, R. Singh, “DevOps: A historical review and future works”, in *Proc. 2021 Int. Conf. Comp. Commun. Intell. Syst.*, Greater Noida, India, pp. 366–371, 2021. doi: 10.1109/ICCCISS1004.2021.9397235
- [11] K. Maroukian, S.R. Gulliver, “Leading DevOps practice and principle adoption”, ArXiv abs/2008.10515, 2020.
- [12] N. Bhuiyan, “A framework for successful new product development”, *J. Indust. Eng. Manag.*, vol. 4, no. 4, 2011. doi: 10.3926/jiem.334.
- [13] R.G. Oliveira, “Lean and performance measuring – Developing a new performance measurement framework to fit lean”, M.S. thesis, Universidade de Porto, 2018.
- [14] M. Sallin, M. Kropp, C. Anslow, J.W. Quilty, A. Meier, “Measuring software delivery performance using the four key metrics of DevOps”, in *Agile Processes in Software Engineering and Extreme Programming. XP 2021. Lecture Notes in Business Information Processing*, vol 419, P. Gregory, C. Lassenius, X. Wang, P. Kruchten, eds. Cham: Springer, 2021. doi: 10.1007/978-3-030-78098-2_7
- [15] R. Pietrantuono, A. Bertolino, G.D. Angelis, B. Miranda, S. Russo, “Towards continuous software reliability testing in DevOps”, in *2019 IEEE/ACM 14th Int. Workshop on Automation of Software Test*, Montreal, QC, Canada, 2019. doi: 10.1109/AST.2019.00009
- [16] “What is DevOps?” [Online]. Available: <https://www.atlassian.com/devops>
- [17] A. Kosyakov, S.M. Beamer, S.J. Seymour, W.N. Sweet, *Systems Engineering. Principles and Practice*. Moscow, Russia: DMK Press, 2014, 624 p.
- [18] D.K. Hitchins, *Systems Engineering. A 21st Century Systems Methodology*. Wiley, 2007, 528 p.

P.P. Maslianko, I.V. Savchuk

DEVOPS – CONCEPT AND STRUCTURAL REPRESENTATION

Problems. Today, along with a huge number of methods, technologies and means of improving business efficiency, the concept of DevOps (Development & Operations) is also gaining rapid development.

In particular, the application of the DevOps concept in business leads practitioners to the question of the feasibility of application, formalization of this process, and usually the development of their own DevOps system focused on achieving the specified values of the metrics of the organizational structure.

Research goal and objectives. The purpose of this article is to study the essence, concept and definition of the DevOps concept, define the entities of the DevOps concept and formalize the unified DevOps model.

Methods of research. Methodology of comparative analysis of scientific results and engineering practices of DevOps implementation, systematization of entity classes of the DevOps concept in the form of a Venn diagram, graphic modeling of a unified DevOps model.

Research results. The practical value and effectiveness of DevOps implementation, the lack of theoretical research and justification of DevOps systems engineering methods, apart from examples of individual patterns and individual architectural solutions of DevOps for the IT industry, have been established.

The article substantiates and proposes a structural representation of the DevOps concept in the form of a Venn diagram as a system of entities of production, support, management of production and support processes, and relationships between them. These results make it possible to choose theoretical and applied tools for the formalization of a unified model of DevOps systems.

Conclusions. The unified model of the DevOps system in the form of a diagram of components and formalized specifications of the interfaces between these components and the unified model of the DevOps system in the form of a diagram of the activities of these components make it possible to proceed to the development of well-founded engineering methods of DevOps systems.

Keywords: DevOps, DevOps model, DevOps system engineering, DevOps concept, DevOps metrics, DevOps processes, software product management, collaboration, competitive advantage.

Рекомендована Радою
факультету прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
22 жовтня 2021 року

Прийнята до публікації
14 лютого 2022 року