



М. В. Орлов, В. В. Пасічник

Національний університет "Львівська політехніка", м. Львів, Україна

СИСТЕМНЕ ОЦІНЮВАННЯ РИЗИКІВ І ВИКЛИКІВ ПІД ЧАС ВПРОВАДЖЕННЯ МЕТОДОЛОГІЇ DEVOPS У КОРПОРАТИВНИХ ІТ-ІНФРАСТРУКТУРАХ

Досліджено особливості застосування методології в ІТ-проектах провідних фірм України. Наведено методику системного оцінювання ризиків і викликів під час впровадження методології DevOps у корпоративних ІТ-інфраструктурах. Проведений аналіз дав змогу сформувати системний погляд на теоретичні результати цієї методології. Проаналізовано результати практичних розробок і впровадження методології DevOps в реальних корпоративних ІТ-середовищах. Встановлено, що наразі немає методологічно цілісних системних підходів до формування комплексних стратегій щодо ідентифікації, класифікації та повного розуміння ризиків і викликів, які генеруються під час впровадження DevOps концепцій в корпоративних ІТ-структурах. Сформульовано мету, об'єкт та предмет оригінального дослідження, а також завдання, виконання яких дало змогу забезпечити їх реалізацію. Сформовано та надано фахові трактування множини базових понять та означень, які слугують термінологічним базисом для досліджуваного фрагменту оригінальної проблемної області. Сформовано послідовність основних етапів впровадження методології DevOps в ІТ-інфраструктурах. Серед ключових практик зазначеної методології виділено постійну інтеграцію, постійне впровадження, постійне розгортання, мікросервісну архітектуру, а також тестування на етапі розроблення програмного продукту. Внаслідок проведеного аналізу сформульовано обґрунтовані висновки про те, що однією із ключових особливостей успішного втілення методології DevOps є створення та активне використання в цих процесах крос-функціональних команд, які об'єднують розробників, тестувальників, інженерів відділу експлуатації та інші ролі. З'ясовано переваги та недоліки DevOps, а також визначають основні ризики та виклики, які виникають при її впровадженні. Проаналізовано ситуації, які можуть генеруватись в процесах імплементації методології DevOps у випадках несумісності відповідних інструментів і відсутності стандартів. Виокремлено групи ризиків, пов'язаних з недостатнім рівнем автоматизації підтримки функцій безпеки ІТ-інфраструктури на належному рівні. Сформоване системне бачення щодо компонування окремої стратегії мінімізації ризиків і подолання викликів під час впровадження методології DevOps в конкретних ІТ-інфраструктурах. У роботі використано системний підхід, який дає змогу комплексно охопити наукові та технологічні особливості DevOps, врахувати взаємозв'язки між ними та оцінити їх вплив на ефективність та безпеку ІТ-проектів. Запропоновано рекомендації щодо оптимізації процесів DevOps, зменшення ризиків і подолання викликів, а також окреслюють перспективи подальшого дослідження в цьому напрямі.

Ключові слова: масштабування; неперервна інтеграція/доставка; опис кодом; керівник проекту; аналіз даних інцидентів, виявлення помилок, віртуальне середовище.

Вступ / Introduction

У сучасному бізнес-середовищі корпоративні ІТ-інфраструктури зазнають постійних і стрімких змін з метою підтримки високого темпу розвитку та конкурентоспроможності. Однією з ключових стратегій в цьому контексті є впровадження методології DevOps, що спрямована на покращення ефективності та швидкості розроблення та впровадження програмного забезпечення. Однак цей процес супроводжується значними ризиками та викликами, які можуть виникнути через комплексні зміни в організаційній культурі, процесах розроблення, архітектурі систем та керуванні змінами. Дослідження та системне оцінювання цих ризиків і викликів є надзвичайно важливими для успішної імплементації методології DevOps у корпоративних ІТ-інфраструктурах.

Відсутність цілеспрямованих стратегій та ретельного системного аналізу може призвести до непередбачених наслідків, таких як збої в роботі систем, погіршення якості продукту або фінансових втрат. Отже, є актуальним проведення наукового дослідження з метою ідентифікації, класифікації та розуміння ризиків і викликів, що виникають під час впровадження методології DevOps у корпоративних ІТ-інфраструктурах, а також розроблення рекомендацій щодо ефективного управління ними та їх мінімізації.

Об'єкт дослідження – оцінювання ризиків і викликів під час впровадження методології DevOps в корпоративних ІТ-інфраструктурах.

Предмет дослідження – методи і засоби оцінювання ризиків, які виникають під час впровадження методології DevOps у корпоративних ІТ-інфраструктурах.

Інформація про авторів:

Орлов Микола Вікторович, аспірант, кафедра інформаційних систем та мереж.

Email: orlov.nv.86@gmail.com; <https://orcid.org/0009-0007-9835-6177>

Пасічник Володимир Володимирович, д-р техн. наук, професор, кафедра інформаційних систем та мереж.

Email: vpasichnyk@gmail.com; <https://orcid.org/0000-0002-5231-6395>

Цитування за ДСТУ: Орлов М. В., Пасічник В. В. Системне оцінювання ризиків і викликів під час впровадження методології devops у корпоративних ІТ-інфраструктурах. Науковий вісник НЛТУ України. 2024, т. 34, № 3. С. 125–132.

Citation APA: Orlov, M. V., & Pasichnyk, V. V. (2024). Systemic assessment of risks and challenges in implementing the devops methodology in corporate IT infrastructures. *Scientific Bulletin of UNFU*, 34(3), 125–132. <https://doi.org/10.36930/40340316>

Мета роботи – проаналізувати ризики і виклики, пов'язані із впровадженням методології DevOps в корпоративних IT-інфраструктурах, що дасть змогу розробити ефективні стратегії мінімізації ризиків і подолання відповідних викликів.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

1. Розглянути базові принципи методології DevOps та оцінити їх значущість в умовах функціонування сучасних корпоративних IT-інфраструктур.
2. Виявити основні ризики, що виникають під час впровадження методології DevOps, зокрема технічні, організаційні та безпекові особливості.
3. Визначити виклики, які виникають під час впровадження DevOps методології, зокрема, в культурних, організаційних і технічних аспектах.
4. Подати стратегії мінімізації ризиків і подолання викликів під час впровадження DevOps, зокрема оцінку ризиків, розроблення плану впровадження та навчання персоналу.

Аналіз останніх досліджень та публікацій. Божич Велібор [3] зазначає, що DevOps (англ. *Development and Operations*) є методологією розроблення програмного забезпечення, яка сфокусована на співпраці, комунікації та інтеграції команд розробників програмного забезпечення та команд IT-операцій. Метою DevOps є спрощення процесу розроблення програмного забезпечення, скорочення часу виходу на ринок і підвищення надійності та якості чергових версій випусків програмного забезпечення. Практики та інструменти DevOps можуть допомогти організаціям скоротити час виходу на ринок, покращити якість програмного забезпечення, сприяти підвищенню їх ефективності та продуктивності. Методологію зосереджено на стандартизації середовищ розробки з метою забезпечення швидкої підготовки випусків. За найліпших умов розробники повинні у всякому оточенні мати доступ до систем автоматизації складання і випуску й мати змогу керувати оточенням, а інформаційно-технологічна інфраструктура бути якнайкраще зосереджена на застосунку.

У роботі [7] наголошено, що використання практик DevOps підвищує прибутковість, культуру та продуктивність праці, яких промислові та технологічні лідери впродовж десятиліть намагалися досягти, з метою пошуку балансу між гнучкістю, надійністю та безпекою.

Методичний посібник DevOps Adoption Playbook [13], який отримав нагороду "Книга року DevOps 2017", містить практичні рекомендації щодо впровадження методології DevOps у масштабі всього підприємства. Автор книги – Санджив Шарма, отримавши практичний досвід впровадження DevOps у компанії IBM; надає унікальні поради та висловлює міркування щодо використання цієї методології у великих організаціях. На погляд автора, методологія DevOps сприяє швидкому та гнучкому створенню високоцінних застосунків і систем, оскільки використовує необхідні практики, інструменти автоматизації, організаційні та культурні зміни, які ведуть до інновацій. Дослідники зазначають, що для впровадження інформаційних технологій провідні корпорації формують власні IT-інфраструктури або використовують IT-інфраструктури, які провайдери надають хмарних технологій.

У роботі [5] запропоновано один із шляхів підвищення якості IT-інфраструктури і наголошено, що підтримання певних її параметрів на високому рівні потре-

бує постійного контролю та пропонується використовувати нейронні мережі для оцінювання якості функціонування IT-інфраструктури.

Результати дослідження, подані у роботі [11], ґрунтуються на використанні декомпозиційно-компенсаційного методу управління рівнем обслуговування, що використовується в корпоративних IT-інфраструктурах, який полягає у декомпозиції завдань та компенсації негативного впливу різних чинників шляхом виділення додаткових ресурсів для критичних додатків. Запропонований підхід передбачає взаємодію процесів – узгодження рівня послуг, планування ресурсів і управління рівнем обслуговування.

У роботі [16] узагальнено моделі та методи управління, запропоновано підхід до розподілу обчислювальних ресурсів у формі частинок заданого розміру (віртуальних вузлів), проаналізовано особливості вирішення проблеми як у разі надлишку, так і за нестачі обчислювальних ресурсів.

Проведений аналіз наукових публікацій дав змогу сформувати цілісний системний погляд на раніше опубліковані наукові результати та практичні розробки, пов'язані з впровадженням методології DevOps у корпоративних IT-середовищах.

Матеріали та методи дослідження. У дослідженнях проблематики за профілем роботи можна використовувати низку методів дослідження з метою глибшого розуміння і отримання нових наукових результатів. Ось кілька з них: анкетування та опитування, зорієнтовані на збір відгуків і міркувань від зацікавлених сторін, якими є інженери з розроблення, виконавці DevOps, керівники проєктів тощо; аналіз інцидентів і помилок, які виникали безпосередньо під час впровадження методології DevOps, що може чіткіше окреслити ключові проблеми та ризики; симуляції та тестування, з метою створення сценаріїв для тестування стратегій DevOps у віртуальних середовищах або на окремих етапах розгортання може допомогти визначати можливі ризики та попередньо локалізувати їх; документальний аналіз з метою вивчення наукових досліджень та практичних розроблень, фахових публікацій та відгуків. Зазначені методи можуть сприяти отриманню повнішого уявлення про ризики та виклики, які виникають в процесі впровадження методології DevOps у корпоративних IT-інфраструктурах.

Результати дослідження та їх обговорення / Research results and their discussion

Аналіз концепції DevOps. Сучасні компанії та фірми постійно здійснюють способи підвищення ефективності своєї діяльності та швидкості впровадження інновацій, зокрема рухаючись шляхом вдосконалення своїх корпоративних IT-інфраструктур.

У цьому контексті концепція DevOps, що поєднує розроблення (Development) та експлуатацію (Operations) програмного забезпечення, набуває дедалі більшої популярності. Впровадження методології DevOps в корпоративній сфері може активувати значні покращення у розробленні та впровадженні програмного забезпечення, водночас привносячи певні ризики та виклики, які повинні бути ретельно оціненими та формувати системні контрзаходи з метою їх істотної мінімізації.

Корпоративна IT-інфраструктура – це інтегрована система комп'ютерних технічних засобів, програмного

забезпечення та комунікаційних мереж, яку використовує організація для забезпечення ефективної реалізації різноманітних бізнес-процесів і функцій. Зазвичай вона охоплює усі складові елементи, зокрема сервери, мережеве обладнання, системи зберігання даних, бази даних, програмне забезпечення для автоматизації бізнес-процесів, інструменти для аналізу даних, а також низку сервісних інформаційних технологій.

Така інфраструктура відіграє ключову роль у забезпеченні безперервності та якості ведення бізнесу, у підтримці операційних процесів, обмінів даними, убезпеченні та конфіденційності, а також в управлінні ресурсами та взаємодії з клієнтами та партнерами. Корпоративну IT-інфраструктуру розробляють та підтримують з урахуванням потреб та вимог конкретної організації, її галузі діяльності, масштабу та стратегічних цілей. Вона є важливою складовою частиною інформаційно-технологічної стратегії організації, істотно впливає на її за-

гальну продуктивність, конкурентоспроможність та здатність до інноваційного розвитку.

Загалом же DevOps є методологічним підходом до процесів розроблення та експлуатації програмного забезпечення, що базується на спільній відповідальності розробника та оператора продовж всього життєвого циклу програмного продукту, який містить розроблення, експлуатацію та сервісну підтримку. У фахових середовищах стверджують, що DevOps – це культурна та процесна практика, яка спрямована на зменшення часу між внесенням змін у програмне забезпечення та їхнім впровадженням у середовище використання продукту [1]. Вона поєднує процеси розроблення (Development) та експлуатацію (Operations), сприяючи їх автоматизації, вдосконаленню комунікації та спільній відповідальності за результат інформаційно-технологічних інновацій (рис. 1).

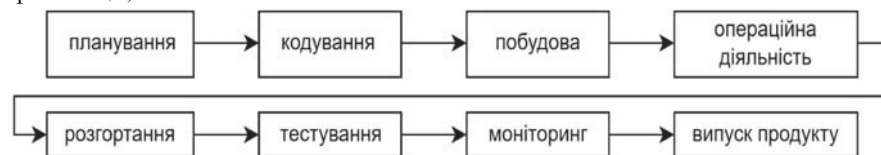


Рис. 1. Основні етапи реалізації методології DevOps у корпоративних IT-інфраструктурах / Main stages of implementing the DevOps methodology in corporate IT infrastructures

Основними принципами методології DevOps є автоматизація процесів розроблення, тестування та впровадження; контроль версій коду та конфігурацій; використання інструментів для моніторингу та збирання метрик; впровадження методологій Agile, управління проєктами; інтеграція процесів розроблення та експлуатації для забезпечення оперативності та стабільності реалізації етапів життєвого циклу програмних продуктів.

Серед ключових практик DevOps варто назвати Continuous Integration (постійну інтеграцію), Continuous Delivery (постійне впровадження), Infrastructure as Code (інфраструктура як код) [10], Microservices (мікросервісну архітектуру), тестування на етапі процесів розроблення та експлуатації, а також втілення концепту DevSecOps, який об'єднує в собі як DevOps, так і практики безпеки Sec.

Впровадження методології DevOps дає змогу організаціям значно прискорити темпи розроблення та впровадження програмного забезпечення, зменшити кількість помилок та ризиків, підвищити якість продукту, покращити комунікацію та співпрацю між розробниками та операторами, а також забезпечити більш гнучке реагування на зміни в бізнес-вимогах.

Основні класи ризиків при впровадженні DevOps. Впровадження методології DevOps супроводжується ризиками, які потрібно аналізувати та враховувати. Розглянемо основні класи ризиків, пов'язаних з впровадженням DevOps підходу в корпоративних IT-інфраструктурах. Впровадження методології DevOps зазвичай потребує значних змін у внутрішній структурі та базових культурних концептах організації. Цей підхід прагне до єднання процесів розроблення та експлуатації програмного забезпечення, що, водночас, генерує формування нових команд, запровадження нових процедур та процесів.

Перехід до методології DevOps передбачає відмову від традиційного відділення процесу розроблення від оперативного забезпечення та підвищення взаємодії між ними. Це, водночас, може потребувати не лише

технічних, але й певних культурних трансформацій та змін. У такій організації варто переглянути систему корпоративних цінностей, принципи та способи співпраці між відділами, підрозділами та командами. Впровадження DevOps передбачає активне використання командного підходу під час реалізації базових завдань, відкритості до змін та освоєння нових методологій.

Одним з ключових особливостей DevOps є створення крос-функціональних команд, які об'єднують розробників, тестувальників, інженерів відділу експлуатації та інші ролі. Це дає змогу істотно скоротити час на виправлення помилок, підвищити швидкість впровадження та знизити загальне число комунікаційних помилок. Ризик під час створення крос-функціональних команд може полягати у відсутності або недостатньому рівні співпраці та взаєморозуміння між різнопрофільними фахівцями в командах (до прикладу, мова може йти про нерівномірний розподіл обов'язків, конфлікти в команді, неналежне комунікування тощо).

Для успішного впровадження методології DevOps корпорація може змінити свою структуру, утворивши нові підрозділи або групи, які відповідатимуть за процеси розроблення та впровадження нових інструментів навчання персоналу, підтримку автоматизованих процесів тощо.

Розробники, інженери відділу експлуатації, тестувальники та інші члени команд повинні отримати необхідні навички та знання для ефективної роботи в новій культурі відповідно до нових методологій.

Зміна структури та культури організації – складний процес, який зазвичай критично впливає на успішне впровадження DevOps. Корпорації, які успішно адаптуються до цих змін, мають більше можливостей для інновацій та швидкого реагування на зміни ринку.

Впровадження методології DevOps потребує підвищеної уваги до технічних особливостей, оскільки пропонує підхід, який містить автоматизацію, інтеграцію та зміни наявних процесів розроблення та експлуатації. При цьому технічні ризики розподіляються на кілька

підгруп. До першої групи відносимо ризики, несумісні з наявною інфраструктурою.

Йдеться про неспроможність наявної інфраструктури відповідати вимогам, які накладаються методологією DevOps. Ситуація може подаватись як така, коли наявні сервери, бази даних або мережеві ресурси є недостатньо потужними, або не мають необхідного програмного забезпечення для ефективної роботи з інструментами автоматизації та інтеграції.

Ще одна група ризиків може зосереджуватись на невідповідностях обраних інструментів, що, водночас, може призводити до невдач впровадження методології DevOps. Недостатня функціональність, несумісність з іншими інструментами, або складність використання можуть сповільнити процеси впровадження та створити додаткові технічні проблеми.

Наступною групою ризиків є недостатня автоматизація, оскільки одним з ключових принципів DevOps є автоматизація процесів розроблення, тестування та впровадження. Недостатня автоматизація може призвести до окремих помилок, затримок у випусках нових релізів і нестабільності всіх рівнів системи. Ретельні пропрацьовані стратегії розроблення, реалізації процесів та впровадження автоматизації, зокрема неперервну інтеграцію CI (англ. *Continuous Integration*) та неперервне оновлення CD (*Continuous Delivery*) [4], тестування та моніторингу дадуть змогу уникнути зазначених ризиків. Неперервна інтеграція CI (англ. *Continuous Integration*) – це практика розроблення програмного забезпечення, яка полягає в автоматизованому процесі регулярного об'єднання (інтеграції) коду кількох розробників у спільний репозитарій. Основна мета CI – забезпе-

чити швидке об'єднання змін у коді, виявлення конфліктів та помилок під час розроблення, а також автоматизувати процеси тестування для забезпечення якості коду. Неперервне оновлення CD (англ. *Continuous Delivery*) – це практика, яка передбачає автоматизоване випуск готового програмного забезпечення після проходження усіх етапів розроблення, тестування та валідації. Головна ідея CD – забезпечити можливість випуску нової функціональності або виправлень у програмному забезпеченні в будь-який момент, шляхом автоматизації процесів випуску та збереження програмного забезпечення завжди готовим до випуску.

Проблема інтеграції нових інструментів і процесів у вже наявну інфраструктуру може виявитися складною та часом високо затратною. Неврахування сумісності, відсутність стандартів може призвести до збоїв у роботі системи та нестабільності відповідного ІТ-середовища.

Ризики можуть виникати і в контексті спроможності масштабування. На рис. 2 подано приклад ситуації, яка може бути згенерована на етапі автоматичного масштабування програмного продукту.

Врахування можливостей масштабування ІТ-інфраструктури та відповідних процесів може призвести до потреби швидкого розширення обчислювальних і технічних потужностей, водночас, неспроможність, або обмеження можливостей такого розширення зазвичай призводить до збільшення ризиків. Варто зазначити, що впровадження методології DevOps може згенерувати додаткові потенційні загрози для безпеки інформації та інфраструктури. Проаналізуємо деякі можливі групи ризиків безпеки, що пов'язані з впровадженням методології DevOps.

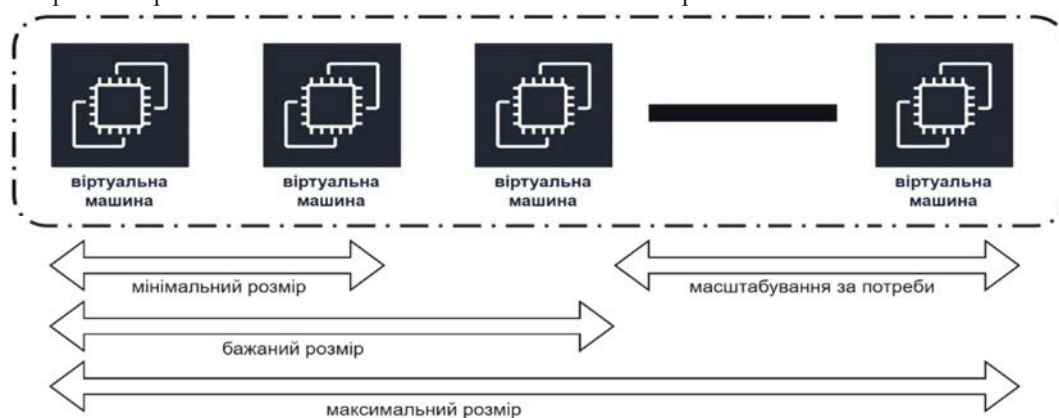


Рис. 2. Ситуативна схема процесу автоматичного масштабування програмного продукту / Situation diagram of the process of automatic scaling of the software product

Під час впровадження методології DevOps збільшується кількість ІТ-систем та сервісів, які піддаються кібератакам. Це створює більшу поверхню атаки для зломисників, які можуть намагатися скористатися слабкими місцями автоматизованих процесів, недоліками в коді або неадекватним контролем доступу.

Окремою групою ризиків є недостатня безпека інструментів DevOps, таких як інструменти автоматизації, системи контролю версій, а також інструменти моніторингу та логування. Такі недоопрацювання в інструментах можуть використати зломисники для кібератак на систему ІТ-інфраструктури.

Окреслимо ризики, генеровані недостатнім рівнем автоматизації функцій безпеки ІТ-інфраструктури. Недостатній контроль доступу, відсутність ефективної системи моніторингу безпеки та недопрацювання про-

цесів розгортання патчів є джерелами додаткових ризиків для корпорацій ІТ-інфраструктур.

Впровадження методології DevOps також може підвищити ризик людського чинника. Помилки конфігурації, необережне використання привілеїв групи адміністраторів і інші помилки, які можуть виникнути через недбалість або недосвідченість персоналу, можуть призвести до серйозних проблем безпеки.

З упровадженням методології DevOps може зрости складність процедур контролю доступу до різних класів ресурсів і систем, або інших проблем безпеки. Невірно налаштований доступ може призвести до витоку конфіденційної інформації, змін у важливих налаштуваннях систем або інших серйозних проблем безпеки.

Для зменшення впливу цих груп ризиків варто приділяти належну увагу впровадженню та підтримці на-

дійних практик безпеки, а також регулярно проводити аудит та оновлювати системи та процеси з метою виявлення і виправлення потенційних проблем.

Окремо зазначимо значущість впливу цих ризиків на якість програмного продукту, оскільки несправна або неякісна реалізація методології DevOps може вплинути на його якість. Несвоєчасний випуск, несистемна імплементація методології можуть призвести до появи багів і негативних ефектів у програмному забезпеченні.

Стратегії мінімізації ризиків і подолання викликів. Впровадження DevOps потребує не лише інноваційних підходів, але й належної стратегії для мінімізації ризиків і подолання викликів, що виникають під час цього процесу. Наведемо перелік ключових принципів успішного впровадження методології DevOps для підтримки та розвитку корпоративної IT-інфраструктури: розуміння DevOps методології, автоматизація та стандартизація процесів, забезпечення безпеки, навчання та розвиток персоналу, неперервну оптимізацію та вдосконалення, управління ризиками, співпрацю з експертами та консультантами.

Принцип розуміння DevOps методології. Варто фундаментально зафіксувати твердження про глибоке занурення в культуру методології DevOps, що передбачає активну адаптацію зазначених цінностей та практик до всіх особливостей роботи корпорацій. Це, водночас, потребує від працівників наявності у них низки таких прагнень.

По-перше, це – відкритість до змін. Культура методології DevOps ґрунтується на відкритості та постійному удосконаленні процесів. Організації варто формувати атмосферу, де працівники відчують впевненість у можливості пропонувати та реалізовувати ідеї щодо покращення процесів розроблення та експлуатації програмних продуктів. Як приклад наведено реалізацію ідеї впровадження методології DevOps в компанії Netflix, яка почала застосовувати цей підхід ще в 2009 р., коли було здійснено перехід від фізичних серверів до використання хмарної платформи AWS. Варто зазначити, що Netflix використовує методологію DevOps для підтримки своєї глобальної платформи стрімінгу, яка обслуговує понад 200 млн користувачів. Корпорація Netflix розробила власні інструменти автоматизації та моніторингу виробничих процесів, такі як Chaos Monkey, Spinnaker, Atlas та інші. Одним з викликів, який постає перед Netflix, була потреба забезпечення високої якості та безперебійності своїх послуг під час швидкого продукування нових функцій та оновлень. З цією метою корпорація використовує культуру хаос-інженерії, суть якої зводиться до експериментального введення штучних відмов у інформаційну систему для перевірки її стійкості та відновлюваності.

По-друге, командна співпраця. У методології DevOps надана перевага спільній роботі та взаємодії між різними відділами та командами. Корпорація повинна стимулювати співпрацю та командну роботу, забезпечуючи ефективний обмін інформацією та знаннями між різними членами найрізноманітніших команд, досягаючи так званого ефекту оркестрування.

По-третьє, зосередженість на досягненні результатів. У культурі методології DevOps підкреслено важливість досягнення конкретних результатів для кожного клієнта. Корпорація повинна формувати у своїх працівників орієнтацію на результат та фокусуватися на вирі-

шенні конкретних завдань та відповідності потребам користувачів.

По-четверте, культура відкритості та відповідальності. Методологія DevOps сприяє розвитку культури відкритості, де працівники відчують себе комфортно щодо вираження своїх думок та ідей. Корпорація створює середовище, де відкриті дискусії та зворотний зв'язок є нормою, а відповідальність кожного колективу за результати роботи є ключовою цінністю.

По-п'яте, підтримка процесів освіти та розвитку, оскільки DevOps є постійно зростаючим полем знань, яке потребує постійного навчання та розвитку. Корпорація повинна інвестувати у навчання свого персоналу, забезпечуючи доступ до ресурсів і підтримку для самонавчання та професійного зростання.

По-шосте, втілення принципу лідерства. Керівництво корпорації повинно виступати як приклад, демонструючи свою відданість культурі DevOps та особисту готовність до змін. Лідери активно підтримують цінності та практики DevOps, сприяючи їх впровадженню та поширенню в організації.

Запровадження культури DevOps у корпорації потребує системної зміни усіх особливостей роботи, від культури комунікації до процесів управління та прийняття рішень. Тому зазначену методологію варто трактувати як цілісну та загальносистемну з досягненням низки синергетичних ефектів, що можуть привнести значні системні покращення, підвищення якості та надійності розроблення, експлуатації програмних продуктів, які складають базу корпоративної IT-інфраструктури.

Принцип автоматизації та стандартизації процесів. Автоматизація та стандартизація процесів є ключовими особливостями успішного впровадження методології DevOps в корпоративній IT-інфраструктурі, оскільки дає змогу забезпечити швидкий та надійний цикл розроблення та супроводження програмного забезпечення. Проаналізуємо дещо детальніше більш розгорнуті стратегії в цьому напрямі.

Розроблення інфраструктури як коду (англ. *Infrastructure as Code, IaC*). IaC дає змогу автоматизувати процес розгортання та конфігурування IT-інфраструктури корпорації за допомогою коду. Це, водночас, уможливорює створення консистентної та повторюваної інфраструктури, що загалом сприяє швидкому впровадженню змін та забезпечує необхідний рівень безпеки та стабільності середовища.

Впровадження систем постійної інтеграції (англ. *Continuous Integration, CI*) **та постійного розгортання** (англ. *Continuous Deployment, CD*). Це дає змогу автоматизувати процеси розроблення, тестування та впровадження програмного забезпечення. Загалом це істотно сприяє швидкому виявленню та виправленню помилок, а також забезпечує оперативну автоматизовану доставку нового функціоналу зацікавленим користувачам (рис. 3).

Використання контейнеризації та оркестрування. Використання контейнеризації, зокрема такої як Docker, спрощує управління залежностями та розгортанням застосунків. Системи оркестрування, такі як Kubernetes (рис. 4), дають змогу автоматизувати процеси розподілу та масштабування контейнеризованих застосунків.

Стандартизація процесів розроблення та експлуатації. Встановлення стандартів для розроблення, тес-

тування, розгортання та моніторингу дає змогу забезпечити консистентність та передбачуваність у роботі ко-

манд. Це спрощує процеси співпраці команд та зменшує ризики появи помилок.

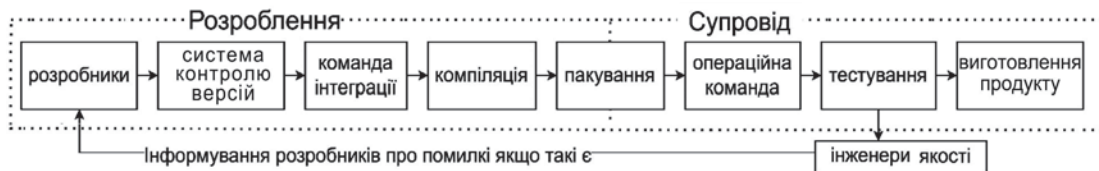


Рис. 3. Автоматизована лінія, що реалізує методологію DevOps / Automated line that implements the DevOps methodology

Впровадження систем автоматизованого тестування. Автоматизоване тестування дає змогу швидко та ефективно перевіряти якість програмного забезпечення. Це сприяє виявленню та виправленню помилок на ранніх етапах розроблення, що дає змогу істотно заощаджувати час та ресурси на втілення масштабованих інновацій.

Моніторинг та ведення журналів обліку. Розроблення систем моніторингу та ведення журналів дає змогу вчасно виявляти та реагувати на проблеми в IT-інфраструктурі та програмному забезпеченні. Це, водночас, забезпечує високу доступність та надійність системи. Як приклад використаємо досвід, напрацьований всесвітньо відомою корпорацією Amazon, яка є одним з найбільших хмарних провайдерів, і використовує методологію DevOps для підтримки своїх продуктів і сервісів, зокрема таких як Amazon Web Services (AWS),

Amazon Prime, Amazon Music, Amazon Video. Корпорація використовує системи моніторингу та журналювання, такі як CloudWatch, X-Ray, CloudTrail, Kinesis, Elasticsearch, для збирання, аналізу, візуалізації та сповіщення про метрики, логи, трасування, події та аномалії, пов'язані з роботою програмних застосунків і IT-інфраструктури. Це, водночас, дає змогу виявляти та усувати проблеми, покращувати продуктивність та якість, а також знижувати ризики втрати даних, порушення безпеки, незадовільної роботи сервісів. Логічним є висновок, у якому узагальнюється згадуване щодо автоматизації та стандартизації процесів, які дають змогу забезпечити ефективність, надійність та стабільність процесів розроблення та експлуатації програмного забезпечення. Це, водночас, зменшує ризики помилок, сприяє швидкому впровадженню змін та підвищує загальну продуктивність команди (рис. 4).

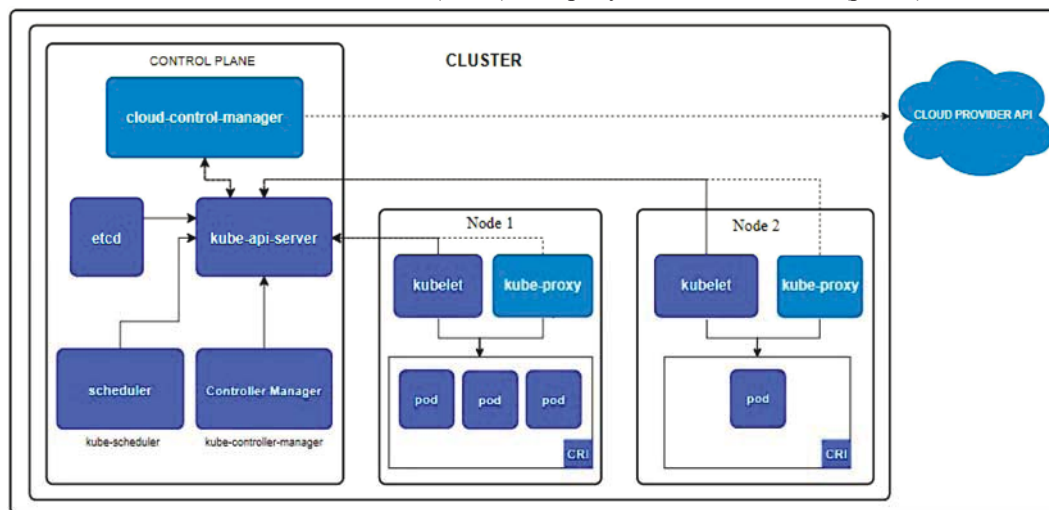


Рис. 4. Офіційна діаграма архітектури Кубернетіс Кластера / Official Kubernetes Cluster architecture diagram [8]

Забезпечення безпеки під час впровадження методології DevOps. Убезпечення є однією з найважливіших складових процесу впровадження методології DevOps, оскільки гарантує надійний захист від потенційних загроз та дій зловмисників, що можуть використовувати ті чи інші вразливості системи. Подамо більш розгорнуті характеристики стратегії забезпечення безпеки.

Захист інфраструктури. Важливо встановити механізми захисту для всієї корпоративної IT-інфраструктури, зокрема сервери, мережеві пристрої, бази даних тощо. Це може містити зокрема встановлення мережевих брандмауерів, застосування сильних паролів і шифрування даних.

Управління доступом. Налаштування прав доступу та привілеїв для користувачів і сервісів сприяє уникненню несанкціонованого доступу до системи. Це може бути досягнуто за допомогою ідентифікації та автентифікації користувачів, використання інструментарію ролей та груп, а також моніторингу та аудиту процедур доступу.

Захист даних. Важливо захищати конфіденційні дані від несанкціонованого доступу зловмисників і витоку через інсайдери. Це може бути досягнуто за допомогою шифрування даних, встановлення правил політик збереження даних і забезпеченням безпеки в момент трансляції даних у мережі.

Моніторинг та виявлення вторгнень. Системи моніторингу та виявлення вторгнень дають змогу виявляти та реагувати на підозрілу активність у реальному часі. Це може містити аналіз журналів подій, моніторинг мережевої активності та виявлення аномальної поведінки користувачів.

Постійне оновлення та патчінг. Важливо регулярно оновлювати програмне забезпечення та усувати виявлені вразливості за допомогою встановлення патчів і оновлень безпеки. Це допомагає зменшити ризики використання вразливостей зловмисниками.

Захист від DDoS-атак. DDoS-атаки можуть призвести до збоїв та перерв у роботі системи. Важливо володіти механізмами захисту, такими як CDN (англ. Con-

tent Delivery Network), встановлення мережевих фільтрів і обмежень, з метою запобігання атакам та забезпечувати доступність системи виключно для ідентифікованих легальних користувачів.

Забезпечення безпеки є невід'ємною частиною впровадження методології DevOps та потребує постійного супроводу та вдосконалення. Це допомагає забезпечити стабільність, надійність та безпеку системи, зменшуючи ризик потенційних загроз та атак.

Обговорення результатів дослідження. На погляд авторів роботи [2], запровадження методології DevOps передбачає низку тактик розроблення програмного забезпечення, спрямованих на скорочення терміну реалізації змін у дизайні програмного забезпечення. Однією з таких тактик є використання інфраструктури як коду, тобто написання плану, який містить специфікації розгортання, готові для оркестрування відповідних середовищ в хмарі. У роботі [2] обговорено необхідні для написання та підтримки цього плану елементи та абстракції, що безпосередньо групуються з ключовим промисловим стандартом OASIS "Специфікація топології та оркестрування для хмарних застосунків" (TOSCA), ухваленого більш ніж 60-ма великими промисловими лідерами з усього світу.

Автори роботи [14] зазначають, що методологія DevOps зорієнтована на покращення співпраці між розробниками та інформаційно-технологічними операційними групами з метою налагодження неперервного і ефективного процесу її розгортання. Проведені інсталяції, на погляд дослідників, засвідчують, що методологія DevOps базується на таких концептах, як культура співпраці, автоматизація та моніторинг.

Однак кількість досліджень, що стосуються зазначеної проблематики на даний час є критично малою, а тому нові результати такого роду наукових розвідок на думку авторів є цінними і допомогатимуть командам розробників ухвалювати вичерпні та коректні рішення.

У роботі [15] зазначено, що для оптимізації IT-проєкту постає потреба впровадження автоматизованих процесів, які сприяють підвищенню надійності кожного етапу його життєвого циклу. При цьому автоматизація процесів розроблення програмного забезпечення, на погляд дослідників, дає змогу використовувати сучасні підходи з неперервного розроблення та впровадження, подавши процес проєктування та розроблення автоматизованих засобів впровадження вебсервісу за DevOps методологією.

У роботі [9] дослідження подано суть концепту та дефініції поняття DevOps, визначення його сутності та формалізацію уніфікованої моделі DevOps як множини трійок елементів міждисциплінарних ресурсів (сутностей) із трьох множин ресурсів (сутностей). Сенапаті М., Бучан Дж. та Осман Г. [12] визначають DevOps як набір принципів і практик для покращення співпраці між розробленням та IT-операціями, проаналізувавши впровадження методології в організаціях з розроблення продуктів у Новій Зеландії, відзначаючи збільшення частоти розгортання з приблизно 30 до 120 випусків на місяць, а також покращення співпраці між IT-розробниками та оперативним персоналом.

У роботі [6] представлено DevOps як набір правил, інструментів і практик для створення більш ефективних способів вирішення майбутніх і існуючих проблем, пов'язаних із розробленням програмного забезпечення,

і підтримки його мети за допомогою операційних лідерів, вважаючи, це, по суті, модель, яка поєднує практики та інструменти, які допомагають підвищити основні цінності організації і є ідеальним архетипом об'єднання поточних розроблених процедур та інструментів.

Отже, за результатами виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – розроблено з використанням системного підходу процедури формування основних груп ризиків і викликів, а також множини тактик їх мінімізації та подолання під час впровадження методології DevOps у корпоративних IT-інфраструктурах.

Практична значущість результатів дослідження – сформовані в процесі дослідження прикладні методи зорієнтовані на їх практичну реалізацію та використання у IT-службах та підрозділах фірм, компаній та корпорацій, які запроваджують методологію DevOps у своїх корпоративних IT-інфраструктурах.

Висновки / Conclusion

Розглянуто важливі особливості впровадження методології DevOps у корпоративних IT-інфраструктурах, а також стратегії мінімізації ризиків і подолання викликів, які можуть виникнути під час реалізації цього процесу. За результатами дослідження можна зробити такі основні висновки.

Сформульоване в процесі проведеного дослідження твердження, що успішне впровадження методології DevOps потребує не лише технічних і технологічних знань і навичок, але й змін в корпоративній культурі та організаційних процесах. Важливим є створення сприятливого середовища для співпраці між розробниками та операторами, забезпечення ефективної високотехнологічної автоматизації процесів розроблення та експлуатації, а також постійного вдосконалення цих процесів з урахуванням змін вимог як ринку загалом, так і конкретних технологій.

Визначено ключові ризики, пов'язані з впровадженням методології DevOps: такі як організаційні, технічні, особливості безпеки та запропоновані відповідні стратегії їх мінімізації.

Зазначено, що, впровадження методології DevOps може згенерувати значні переваги для організацій, зокрема підвищення ефективності, зменшення часу на розроблення та випуск продукту, підвищення якості програмного забезпечення та збільшення гнучкості структур і адаптивності їх до змін. При цьому важливим для досягнення успіху є ретельно розроблена стратегія впровадження, врахування потенційних ризиків і вдосконалення відповідних процесів на постійній основі.

Подяка. Автори публікації висловлюють вдячність колегам за серію фахових дискусій та обговорень, які до певної міри підвищують якість сприйняття та активізації просування на ринку методології DevOps. Зазначена методологія вважається новим оригінальним системотехнічним поглядом на проблеми розроблення та супроводу широкого спектру інформаційних систем і технологій на базі розлогих корпоративних IT-інфраструктур. У спогадах представників старшого покоління українських айтишників проаналізований у роботі підхід залишається історичним відбитком періоду активного створення та організації процесів ефективного

функціонування на підприємствах, у виробничих, невинних об'єднаннях та організаціях комплексних відділів і служб автоматизованих систем управління (АСУ) різних класів, рівнів і профілів 1970-1980 років.

Матеріал, покладений в основу даної статті, практично напрацьовувався впродовж тривалого часу на підставі великої кількості реальних ІТ-проектів, що реалізовувались промисловим підрозділом DevOps однієї з крупних топових українських ІТ-компаній.

References

1. Armstrong, S., & Lowe, S. (2019). DevOps for Networking. Birmingham: Packt Publishing. URL: <https://subscription.packtpub.com/book/cloud-and-networking/9781786464859/copyrightpg>
2. Artac, M., Borovssak, T., Di, Nitto, E., Guerriero, M., & Tamburri, D. A. (2017). DevOps: Introducing Infrastructure-as-Code IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires, Argentina, 497–498. <https://doi.org/10.1109/ICSE-C.2017.162>
3. Božić, V. (2023). DevOps – software development methodology. <https://doi.org/10.13140/RG.2.2.11626.80327>
4. CI/CD (2024). Continuous Integration and Continuous Delivery/geeksforgeeks. URL: <https://www.geeksforgeeks.org/ci-cd-continuous-integration-and-continuous-delivery/>
5. Dorogyy, Y. Y., Telenik, S. F., Halushko, D. A., & Tsurkan, V. V. (2015). Structural Optimization of Neural Network for Qualitative Evaluation Method of IT-Infrastructure Functioning. Information and Telecommunication Sciences: *International Research Journal*, 6(2), 36–43. URL: <https://ela.kpi.ua/handle/123456789/18397>
6. Jha, Pratibha, & Khan, Rizwan. (2018). A Review Paper on DevOps: Beginning and More To Know. *International Journal of Computer Applications*, 180, 16–20. <https://doi.org/10.5120/ij-ca2018917253>
7. Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. Portland: Publisher IT revolution press. URL: <https://www.amazon.com/DevOps-Handbook-World-Class-Reliability-Organizations/dp/1942788002>
8. Kubernetes Official documentation. Kubernetes.io 2024. URL: <https://kubernetes.io/docs/concepts/architecture/>
9. Maslyanko, P. P., & Savchuk, I. V. (2021). DevOps: concept and structural representation. *KPI Scientific News*, 4, 39–51. <https://doi.org/10.20535/kpissn.2021.4.261938>
10. Morris, K. (2016). Infrastructure as Code: Managing Servers in the Cloud. Sebastopol: O'Reilly & Associates Inc. URL: <https://www.amazon.com/Infrastructure-Code-Managing-Servers-Cloud/dp/1491924357>
11. Rolik, O., Kolesnik, V., & Halushko, D. (2016). Decomposition-compensation method of service level management in corporate IT infrastructures with the use of adaptive genetic algorithm. In Proceedings of the International Conference Radio Electronics & Info Communications (UkrMiCo), 1–5. Kiev, Ukraine. URL: <https://ieeexplore.ieee.org/document/7739643>
12. Senapathi, M., Buchan, J., & Osman, H. (2019). DevOps capabilities, practices, and challenges: Insights from a case study. *CoRR*. <https://doi.org/10.1145/3210459.3210465>
13. Sharma, S. (2017). The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise. Hoboken: Wiley. URL: <https://www.amazon.com/DevOps-Adoption-Playbook-Multi-Speed-Enterprise/dp/1119308747>
14. Siebra, C., Lacerda, R., Cerqueira, I., Quintino, J., Florentin, F., Silva, F., & Santos, A. (2018). From Theory to Practice: The Challenges of a DevOps Infrastructure as Code Implementation. In Proceedings of the 13th International Conference on Software Technologies (ICSOFT 2018), 427–436. <https://doi.org/10.5220/0006826104270436>
15. Sokol, O. V., Tsvetkova, T. P., & Khariv, N. O. (2022). Design and development of automated web service implementation processes using the DevOps methodology. *Bulletin of the National University of Water Management and Nature Management*, 1(97), 231–243. URL: <http://ep3.nuwm.edu.ua/id/eprint/24831>
16. Telenyk, S., Rolik, O., Bukasov, M., & Halushko, D. (2013). Models and methods of resource management for VPS hosting. Technical Transaction. Automatic Control, 4-AC, 41–52. URL: <https://repozytorium.biblos.pk.edu.pl/resources/30710>

M. V. Orlov, V. V. Pasichnyk

Lviv Polytechnic National University, Lviv, Ukraine

SYSTEMIC ASSESSMENT OF RISKS AND CHALLENGES IN IMPLEMENTING THE DEVOPS METHODOLOGY IN CORPORATE IT INFRASTRUCTURES

A systematic assessment of risks and challenges in the implementation of the DevOps methodology in corporate IT infrastructures is given. The conducted analysis enabled forming a systematic view of the theoretical results of this methodology. Therefore, the article is devoted to the study of the peculiarities of the application of the methodology in the IT projects of the leading companies of Ukraine. Concurrently, the results of practical development and implementation of DevOps methodology in real corporate IT environments are analysed. In the course of research, it was established that there are currently no methodologically coherent systemic approaches to the formation of complex strategies for identification, classification and full understanding of risks and challenges generated by the implementation of DevOps concepts in corporate IT structures. The purpose, object and subject of the original research are formulated, as well as tasks, the fulfilment of which would enable ensuring their implementation. Professional interpretations of a set of basic concepts and definitions, which serve as the terminological basis for the fragment of the original problem area studied in the article, have been formed and provided. The sequence of the main stages of the implementation of the DevOps methodology in IT infrastructures has been formed. Among the key practices of the specified methodology there are as follows: continuous integration, continuous implementation, continuous deployment, microservice architecture, and testing at the stage of software product development. As a result of the analysis, reasonable conclusions were formulated, in particular, that one of the key features of the successful implementation of the DevOps methodology is the creation and active use in these processes of cross-functional teams that unite developers, testers, engineers of the operation department and other roles. The advantages and disadvantages of DevOps are clarified, as well as the main risks and challenges that arise during its implementation are determined. The situations that can be generated in the processes of implementing the DevOps methodology in cases of incompatibility of the relevant tools and lack of standards have been analysed. Groups of risks related to the insufficient level of automation of support of IT infrastructure security functions at the appropriate level are separately highlighted. A system vision was formed regarding the composition of a separate strategy for minimizing risks and overcoming challenges when implementing the DevOps methodology in specific IT infrastructures. The work uses a systematic approach that makes it possible to comprehensively cover the scientific and technological features of DevOps, considering the interrelationships between them and evaluate their impact on the efficiency and security of IT projects. Some recommendations for optimizing DevOps processes, reducing risks and overcoming challenges are offered, and prospects for further research in this direction are outlined as well.

Keywords: scaling; CI/CD; code description; project manager; incident data analysis; error detection; virtual environment.