

Орлов М.В.

Національний університет «Львівська політехніка»

Грибовський О.М.

Національний університет «Львівська політехніка»

Жовнір Ю.І.

Національний університет «Львівська політехніка»

Дуда О.М.

Тернопільський технічний університет імені Івана Пулюя

ВІД КОНЦЕПЦІЇ ДО РЕАЛЬНОСТІ: РОЛЬ МЕТОДОЛОГІЇ DEVOPS В ЕКОСИСТЕМАХ ІОТ

В статті розглядаються сучасні підходи до інтеграції методології DevOps у розроблення та підтримку рішень на базі інформаційних технологій Інтернету речей. У контексті швидкого розвитку Інтернету речей, де кількість підключених пристроїв стрімко зростає, ефективне управління життєвим циклом програмного забезпечення стає критично важливим завданням. Методологія DevOps пропонує принципи безперервної інтеграції та доставки (CI/CD), що дозволяють оптимізувати процеси розроблення, тестування, впровадження та обслуговування IoT-рішень. У статті проаналізовано основні етапи впровадження методології DevOps у різних елементах IoT-екосистем, включаючи хмарну інфраструктуру, мережеві компоненти та кінцеві пристрої. Автори акцентують увагу на проблемах масштабування IoT-рішень та підтримки стабільної роботи у середовищах з високим рівнем динаміки і складності. Дослідження підкреслює, що використання методології DevOps в системах на базі IoT дозволяє знизити час розроблення, забезпечити оперативне тестування оновлень і підвищити надійність систем завдяки моніторингу в реальному масштабі часу. Також розглянуто аспекти кібербезпеки, які є критично важливими для IoT-рішень, зокрема впровадження автоматизованих перевірок безпеки на кожному етапі розроблення. У статті проаналізовано застосування DevOps для управління великими IoT-мережами та підкреслено значення міждисциплінарної співпраці команд розробників, інженерів та операторів. Автори доходять висновку, що DevOps сприяє не лише покращенню якості та швидкості розробки, але й інтеграції нових функцій без порушення роботи системи.

Ключові слова: CI/CD (Безперервна інтеграція та доставка), автоматизація, хмарна інфраструктура, моніторинг у реальному часі, кібербезпека, масштабування, управління життєвим циклом програмного забезпечення, інтеграція апаратного та програмного забезпечення.

Постановка проблеми. У сучасному світі інформаційні технології Інтернету речей (Internet of Things – IoT) стають все більш важливою складовою повсякденного життя та виробничих процесів. Стрімке зростання кількості підключених пристроїв та висока складність IoT-систем вимагають формування нових системно-методологічних підходів до розроблення та управління відповідним програмним забезпеченням. Такого роду розлогі функціонально та територіально розподілені інформаційні системи генерують потребу переосмислення чотирьох етапів її життєвого циклу, який включає – системний аналіз, системне проектування, системну інтеграцію та системне

адміністрування. Використання сучасної методології DevOps сприяє об'єднання фази розроблення (Development) та оперування (Operations) для системного забезпечення процесів безперервної інтеграції та доставки програмного забезпечення. Запровадження методології DevOps в інформаційних системах, базованих на IoT є не просто модним трендом, це технологічна необхідність, яку треба обов'язково забезпечувати у випадках, коли метою є реалізація в повній мірі високого потенціалу технологій IoT.

Аналіз останніх досліджень і публікацій. Проведено аналіз наукових дерел та існуючих підходів дозволив здійснити огляд сучасних нау-

кових джерел щодо використання методології DevOps у сфері IoT. Автори творчо використали методику дослідження, яка була запропонована Кітгенгальмом та Чартером [1].

Проведений авторами аналіз наукових публікацій визначив поточний стан досліджень методології DevOps в контексті побудови систем з використанням технологій IoT, при цьому були виділені ключові тенденції та виклики, що виникають при реалізації таких інформаційно-технологічних рішень. Ряд авторів у своїх дослідженнях наголошують на перевагах методології DevOps у швидкій адаптації IoT-систем до змінних вимог та зростаючого навантаження. Методологія DevOps розглядається як високотехнологічний процес розробки та поширення програмного забезпечення. Він поєднує в собі культуру співпраці та консолідації підходів до управління, технологій кодування та системних технік інтеграції. Методологія DevOps подається як нова інноваційна парадигма та концепція, яка впроваджується в галузі IT і стосується програмного забезпечення та вбудованих систем в таких аплікаціях як робототехніка та інтелектуальні агенти. Екосистема IoT подається як набір фізичних пристроїв, таких як давачі та виконавчі пристрої, зінтегрованих в розлогих мережевих структурах, що містять набори серверів та шлюзів, які забезпечують надійне їх підключення та взаємодію. Зазначені пристрої можуть інстальоватись на трьох рівнях, що реалізують три обчислювальні концепції, а саме граничних обчислень, туманних обчислень та хмарних обчислень [2].

У ряді робіт аналізуються питання, як саме розподілені хмарні середовища та інфраструктура як код (IaC) полегшують масштабування великих IoT-мереж. Дослідники вважають безперервну інтеграцію [3] процесом розроблення, у якому розробники та інтегратори імплементують код у спільні інструменти. Це базова практика, яка підтримує безперервність роботи програмного забезпечення розумних агентів, які є основою систем, на базі технологій Інтернету речей. Базуючись на конкретних характеристиках, були розроблені та розгорнуті інформаційні системи з використанням ряду фреймворків для систем на базі IoT. Автори запропонували метамодель інформаційних технологій IoT, що спрямовано на стандартизацію існуючих архітектур систем на базі цієї технології, а також тих, які будуть запропоновані в майбутньому. У певній кількості публікацій розглядаються питання впровадження моніторингових інструментів у IoT-системах, що

працюють у масштабі реального часу з використанням хмарних технологій. Це дозволяє оперативно реагувати на зміни стану системи та вчасно виявляти аномалії [4]. Програми, які відстежують події та збирають дані з давачів, розгортаються на рівнях пограниччя, туману та хмар. У роботі подано загальну метамодель екосистем IoT, яка базується на мікросервісах і підтримується Docker як інструментом контейнеризації та Ansible як інструментом моніторингу. Запропонована метамодель є поданням 5 метамоделей, таких як IoT-мікросервіси – Ansible – Docker – Kubernetes [5]. Проаналізовані результати наукових досліджень підтвердили, що методологія DevOps є ключовою методологією підвищення ефективності та надійності IoT-екосистем. Використання методології DevOps у інформаційних системах, базованих на технологіях IoT, забезпечує швидку адаптацію до змін, оптимізує процеси розроблення та підтримки прийняття рішень, а також підвищує рівень безпеки систем загалом. Водночас відзначаються виклики, пов'язані з інтеграцією методології DevOps у різномірних IoT-середовищах, в яких поєднуються чисельні різнотипові апаратні та програмні компоненти.

Постановка завдання. Метою статті є проаналізувати роль методології DevOps в екосистемах IoT та обґрунтувати шляхи її впровадження для оптимізації процесів розроблення, тестування, розгортання та обслуговування IoT-рішень. Об'єкт дослідження – екосистеми Інтернету речей (IoT), які включають апаратні та програмні компоненти, хмарні сервіси, мережеві інфраструктури та засоби автоматизації. Предмет дослідження – методологія DevOps та її вплив на процеси управління життєвим циклом IoT-рішень, зокрема на розроблення, інтеграцію, тестування, розгортання, масштабування та моніторинг такого класу систем. Для досягнення зазначеної мети визначено такі основні завдання дослідження: провести аналіз концепцій та принципів методології DevOps у контексті їх застосування в IoT-екосистемах; вивчити аспекти масштабованості та надійності IoT-систем, які реалізуються завдяки методології DevOps; розробити рекомендації для впровадження методології DevOps у IoT-проектах з урахуванням їх специфіки та широкого спектру вимог.

Виклад основного матеріалу. Синергія методології DevOps та інформаційних технологій класу IoT (див. Рисунок 1) є критично важливою в прискоренні інновацій, підвищення надійності, живучості та безпеки розлогих системних IoT-рішень. Практики DevOps сприяють автоматиза-

ції багатьох аспектів розроблення та розгортання IoT-застосунків. Вони включають автоматизоване тестування, збірку та розгортання програмного забезпечення, що дозволяє швидше виявляти та виправляти помилки, зменшуючи ризики людських помилок і суттєво підвищуючи надійність реалізації широкої партитури процесів. Використання Безперервної Інтеграції (або Continuous Integration – в подальшому використовуватимемо термін «CI/CD») у розробці інформаційних систем на основі технологій IoT дозволяє частіше випускати оновлення та нові функції програмних продуктів. Це особливо важливо для інтелектуальних IoT-пристроїв, які потребують регулярних оновлень для підтримки належного рівня безпеки та розширення функціональності. Водночас використання CI/CD забезпечує швидке та безпечно впровадження змін, що підвищує рівень задоволеності користувачів такого роду систем. DevOps містить функцію безперервного моніторингу IoT-систем, що дозволяє вчасно виявляти аномалії та проблеми. Це допомагає запобігати збоєм

та забезпечувати стабільну роботу пристроїв. Окрім того, моніторинг дозволяє збирати дані для аналізу продуктивності та багатопараметричної оптимізації інформаційних систем.

Використання концептуального аспекту DevOps-інфраструктура як код спрощує процеси управління складною ІТ інфраструктурою в системах, базованих на технологіях IoT. Розробники можуть швидко відтворювати налаштування ІТ інфраструктури, що полегшує масштабування та забезпечує консистентність середовищ. Така функціональність є особливо важливою для інформаційних систем, що реалізовані на великих, розлогих IoT-мережах, коли необхідно керувати та оркеструвати велику кількість пристроїв.

Використання контейнерів та мікросервісів у архітектурі систем на основі технологій IoT підвищує гнучкість та масштабованість таких систем. Контейнери дозволяють ізолювати застосунки та зменшити рівень їх взаємозалежності, що спрощує процедури розгортання та управління. Мікросервіси дозволяють розробляти застосунки

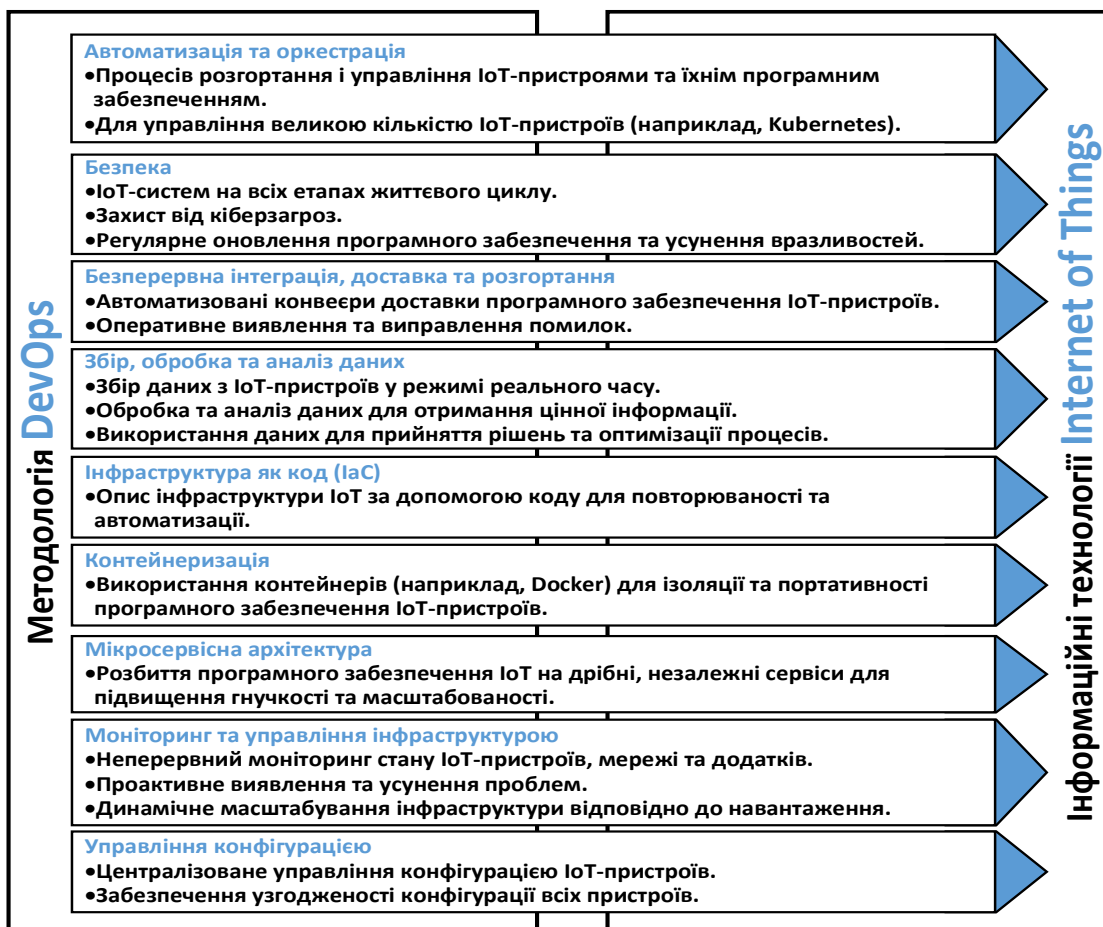


Рис. 1. Взаємодія методології DevOps та інформаційних технологій класу IoT / The interaction of the DevOps methodology and information technologies of the IoT class

як набори незалежних сервісів, що полегшує реалізацію процедур їх оновлення та масштабування.

Для визначення зв'язків методології DevOps та інформаційних систем, базованих на інформаційних технологіях класу IoT автори роботи [6] сформулювали три дослідницькі питання (RQ).

RQ1. Які концепції DevOps застосовуються в контексті програмних продуктів, що реалізують інформаційні системи на основі технологій IoT?

Було визначено п'ятнадцять концептуальних аспектів методології DevOps і співвіднесено їх з процесами автоматизації процедур розгортання [6]. Результати подані на рисунках 2 та 3.

RQ2. Які переваги від впровадження DevOps у програмних системах, базованих на технологіях IoT?

Було ідентифіковано 4 переваги, а результати запиту наведено нижче. Застосовуючи DevOps фреймворк для IoT ви надаєте структуру для ефективного управління IoT-системами, що підвищує продуктивність і зменшує час на інтеграцію з іншими компонентами вашого рішення. Використання хмарних обчислень для хостингу серверів (бекенду) дозволяє легко масштабувати і зберігати дані, забезпечуючи безпечний доступ з будь-якого місця (локації). Операбельність із

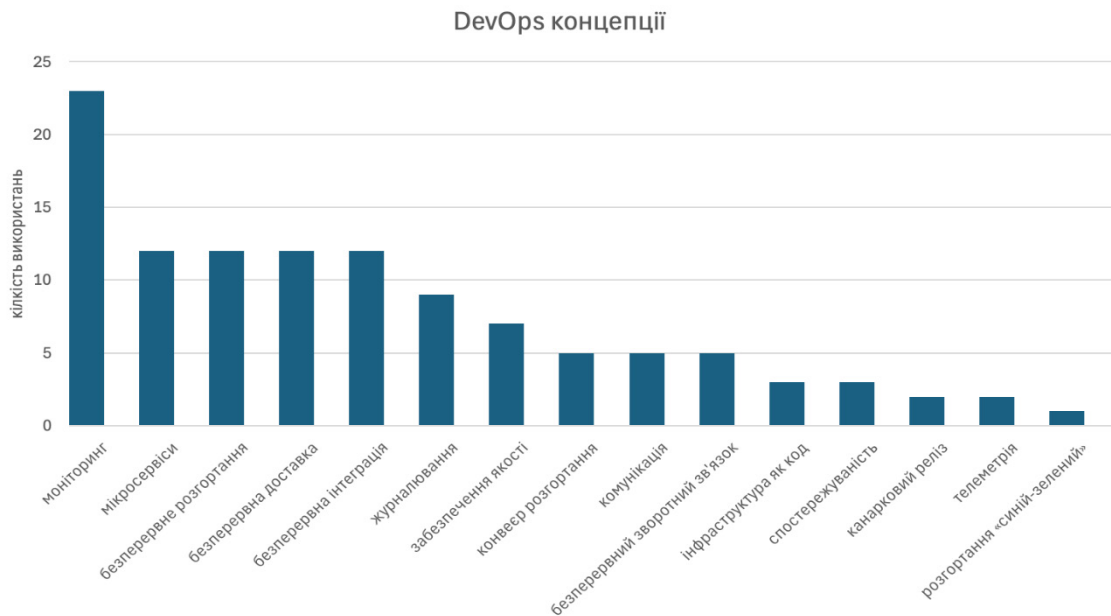


Рис. 2. Концептуальні аспекти DevOps, які застосовуються в програмних продуктах, що реалізують функціонал інформаційних систем з використанням технологій IoT / Conceptual aspects of DevOps, which are used in software products that implement the functionality of information systems using IoT technologies



Рис. 3. Розподіл кількості концептуальних аспектів методології DevOps на різних етапах розроблення програмних продуктів/ Distribution of the number of conceptual aspects of the DevOps methodology at different stages of the development of software products

використанням контейнерів спрощує підтримку й оновлення, що знижує витрати на обслуговування. Мікросервіси сприяють формуванню малих команд, що сприяє швидшій розробці, ізольованому розгортанню та стабільності кожної компоненти.

RQ3. Які виклики виникають при впровадженні DevOps у програмних системах, базованих на технологіях IoT?

Було ідентифіковано виклики. Результат запиту дозволив сформулювати таке твердження: *При впровадженні методології DevOps у програмних системах, базованих на технологіях IoT, виникає ряд викликів, пов'язаних з масштабуванням, управлінням різнорідними пристроями, забезпеченням належного рівня безпеки, підтримкою безперервного розгортання на обмежених ресурсах, а також складністю налаштування моніторингу та автоматизації в умовах нестабільного зв'язку.* Реалізуючи план нашого дослідження ми продовжили роботу по формуванню відповідей на ряд додаткових питань, які наведені нижче:

RQ4. Які інструменти та технології методології DevOps найчастіше використовуються в розробці IoT-застосунків?

Провівши аналітичне дослідження можемо стверджувати про виявлення шести груп інструментів (див. рис. 4).

Подані вище групи інформаційно-технологічних інструментів дозволяють забезпечити безперервну інтеграцію та доставку, автоматизацію процесів, моніторинг та управління конфігураціями, що є критично важливими для успішного розроблення IoT-застосунків [7–9].

RQ5. Методи забезпечення безпеки DevOps, які використовуються в IoT-застосунках.

Інтеграція факторів безпеки на кожному з етапів розроблення засобами методології DevSecOps (Розроблення. Безпека. Операційність) забезпечує проактивний підхід до незалежного рівня забезпечення безпеки. Це включає автоматизоване сканування вразливостей, тестування рівнів безпеки та впровадження кращих практик безпеки [10]. Дослідження належного рівня безпеки засобами методології DevOps в IoT-застосунках вимагає впровадження спеціальних методів та практик, оскільки IoT-застосунки часто містять унікальні безпекові ризики, пов'язані з фізичними пристроями, мережами і даними. Ось кілька ключових методів убезпечення DevOps процесів, які застосовуються в IoT-застосунках (див. рис. 5).



Рис. 4. Інструменти методології DevOps, що найчастіше використовуються в розробці IoT-застосунків/ DevOps tools most commonly used in IoT application development



Рис. 5. Ключові методи забезпечення DevOps процесів в IoT-застосунках / Key methods of ensuring DevOps processes in IoT applications

RQ6. Як методологія DevOps сприяє масштабованості IoT-систем?

Методологія DevOps відіграє ключову роль у забезпеченні масштабованості IoT-систем завдяки використанню методів автоматизації, постійної інтеграції, моніторингу та швидкого реагування на зміни (див. Рис. 6). Однією з ключових переваг методології DevOps є її моделюючі потужності і здатність створювати платформи для моделювання IoT-систем. Це дозволяє ефективно тестувати та впроваджувати складні IoT-рішення, зменшуючи ризики та підвищуючи надійність. Методологія DevOps загалом надає широкий спектр можливостей щодо імітації реальних умов експлуатації IoT-систем, що забезпечує точніше тестування та виявлення потенційних проблем до моменту їх фактичної появи в продуктивному середовищі.

Контейнеризація є важливою складовою методології DevOps, оскільки дозволяє легко створювати ізольовані середовища для моделювання різних компонентів IoT-системи. Наприклад, використання Docker дозволяє розробникам створювати контейнери, що точно відтворюють середовище, в якому будуть працювати IoT-пристрої. Це спрощує тестування взаємодії між різними частинами системи та дозволяє швидко вносити зміни. Використання оркестрування контейнерів за допомогою Kubernetes дозволяє моделювати масштабовані IoT-архітектури, тестувати їх стійкість та ефективність у різних умовах навантаження. Моніторинг та логування є важливими аспектами методології DevOps для IoT-систем. Використання рішень для централізованого моніторингу (наприклад, Prometheus та Grafana)



Рис. 6. Використання методології DevOps в процесах масштабування IoT-систем / Using the DevOps methodology in the processes of scaling IoT systems

дозволяє збирати метрики та аналізувати роботу кожного компонента системи в реальному масштабі часу. Це дає можливість виявляти потенційні проблеми ще до їх виникнення у продуктивному середовищі. Централізоване логування за допомогою ELK stack (Elasticsearch, Logstash, Kibana) забезпечує зручний спосіб зберігання та аналізу логів з усіх IoT-пристроїв, що допомагає швидко знаходити та усувати помилки. Методологія DevOps також забезпечує можливості для симуляції навантаження та стрес-тестування IoT-систем. За допомогою інструментів, таких як Apache JMeter або Gatling, можна створювати симуляції реального навантаження на систему, що дозволяє виявляти вузькі місця та покращувати продуктивність. Наприклад, можна моделювати одночасне підключення тисяч IoT-пристроїв та аналізувати, як система справляється з таким навантаженням. Проведення стрес-тестів дозволяє оцінити стійкість IoT-системи до високих навантажень та виявити можливі проблеми в умовах пікових навантажень.

Управління конфігураціями та інфраструктурою як кодом (IaC) автоматизує створення та налаштування інфраструктури для IoT-систем. Інструменти для управління конфігураціями, такі як Ansible, Chef або Puppet, дозволяють автоматично налаштовувати середовища розробки, тестування та продуктивного використання. IaC (наприклад, Terraform) забезпечує консистентність середовищ, що знижує ризик виникнення проблем через від-

мінності в конфігураціях. Впровадження та активне використання методології DevOps для моделювання IoT-систем забезпечує значні переваги, серед яких висока точність тестування, швидке виявлення та виправлення помилок, а також досягнення підвищеної надійності системи. Це робить DevOps незамінним інструментом для розробників та операторів IoT-рішень, сприяючи підвищенню якості та ефективності їх роботи.

Висновки. За результатами проведених досліджень можемо стверджувати, що кращими DevOps практиками для інформаційних систем, що реалізуються на основі технологій IoT визначено наступні: Безперервна інтеграція/безперервна доставка (CI/CD) що є основною

практикою DevOps вона відіграє важливу роль у розробці IoT. Вона передбачає автоматизацію процесів доставки програмного забезпечення та змін інфраструктури. Для систем на основі технологій IoT інтеграція CI/CD сприяє підтримці якості програмного забезпечення, підвищенню ефективності та прискоренню процесів доставки оновлень або нових функцій на пристрої. Водночас це забезпечує можливість легких відкатів попередніх змін, зменшуючи при цьому впливи потенційних індукованих проблем. Практичні втілення інформаційних систем, базованих на технологіях IoT проаналізовані в процесі цієї наукової розвідки, підтверджують корисність зазначених процедур.

Список літератури:

1. Kitchenham B., Charters S. Guidelines for Performing Systematic Literature Reviews in Software Engineering Technical Report EBSE-2007-01. – Keele University and University of Durham, UK, 2007. 65 p. – Available at: <https://www.dur.ac.uk/resources/ebse/EBSE2007-01.pdf>. Retrieved from https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf
2. K̄halyly B., Belangour A., Erraissi, A., Banane M. Meta-model approach of applied DevOps on Internet of Things ecosystem. *IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. 2020. P.1–6. <https://doi.org/10.1109/ICECOCS50124.2020.9314552>
3. Kim G., Willis J., Debois P., Humble J. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. Portland: IT Revolution Press, 2016. 480 p. Available at: <https://itrevolution.com/book/the-devops-handbook..>
4. Díaz J., Pérez-Martínez J., López-Peña, M., Mena, G., Yague, A. Self-service cybersecurity monitoring as enabler for DevSecOps IEEE Access. – 2019. – Vol. 7. – P. 100283–100295. – <https://doi.org/10.1109/ACCESS.2019.2930000>. Available at: <https://ieeexplore.ieee.org/document/8784768>.
5. Dave D. M., Bhanushali A. Continuous integration and continuous deployment (CI/CD) for AI-enabled IoT systems. 2023. Retrieved from https://www.researchgate.net/publication/380694991_Continuous_Integration_and_Continuous_Deployment_CICD_for_AI-Enabled_IoT_Systems
6. Botta A., De Donato W., Persico V., Pescapé A. Integration of cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*. 2016. Vol.56. P.684–700. <https://doi.org/10.1016/j.future.2015.09.021>
7. Pereira I. M., de Senna Carneiro, T. G., Figueiredo, E. Understanding the context of IoT software systems in DevOps. *arXiv*. 2021. 2103.12345. Available at: <https://arxiv.org/abs/2103.12345>. <https://doi.org/10.48550/arXiv.2103.12345>.
8. Maayan, G. D. A DevOps guide to IoT technology. Retrieved from <https://devops.com/a-devops-guide-to-iot-technology>. [Place of publication: if available]. [Publisher: if available], 2023.
9. Apprecode. (n.d.). DevOps in the creative industries: Streamlining content creation workflows. [Place of publication: if available]. [Publisher: if available], [n.d.] Retrieved from <https://apprecode.com/blog/devops-in-iot-accelerating-innovation-in-the-internet-of-things>.
10. Michalowski, M. Using DevOps practices to enhance IoT security. Retrieved from <https://www.ietfforall.com/using-devops-practices-to-enhance-iot-security>.

Orlov M.V., Hrybovskiy O.M., Zhovnir Yu.I., Duda O.M. FROM CONCEPT TO REALITY: THE ROLE OF DEVOPS METHODOLOGY IN IOT ECOSYSTEMS

The article explores contemporary methodologies for integrating DevOps practices into developing, deploying, and maintaining Internet of Things (IoT) solutions. Given the explosive growth of IoT and the increasing volume of interconnected devices worldwide, the efficient management of software lifecycles has become an essential task for businesses and technology developers. The DevOps methodology provides a framework centered on the principles of continuous integration and continuous delivery (CI/CD), aiming to streamline the processes of coding, testing, deployment, and ongoing support for IoT solutions. The paper delves into the intricacies of applying DevOps at different levels of the IoT ecosystem, encompassing cloud-based infrastructures, communication networks, and edge computing devices. The authors offer a detailed analysis of

how automation is pivotal in coordinating software and hardware interactions, ensuring seamless integration and minimizing the potential for errors. Emphasis is placed on the unique challenges associated with scaling IoT solutions, such as managing the vast data generated by devices and maintaining system stability in highly dynamic and complex operational environments. In the context of optimizing IoT deployments, the article highlights how the DevOps approach helps shorten development cycles and enhance the efficiency of testing updates. Organizations can rapidly address issues by implementing continuous monitoring and real-time data analysis, improving system reliability and robustness. The research further addresses cybersecurity concerns inherent in IoT applications, advocating for automated security assessments and integrating protective measures throughout the development lifecycle. These strategies ensure that vulnerabilities are identified and mitigated proactively, thus reinforcing the overall security posture of IoT systems. The article presents illustrated case studies that demonstrate successful implementations of DevOps strategies in managing extensive IoT networks. These examples showcase the benefits of interdisciplinary collaboration between software developers, network engineers, and operational teams. These examples underscore the significant impact of adopting a DevOps culture on IoT projects' efficiency and scalability while facilitating the seamless addition of new functionalities without disrupting existing services. The authors conclude by asserting that DevOps principles not only elevate the quality and speed of IoT software development but also enable a more adaptive and resilient technological infrastructure. The article offers practical recommendations for professionals, including developers, system architects, and project managers, emphasizing strategies to enhance productivity and responsiveness in IoT environments. These insights are precious for those seeking to implement DevOps methodologies to improve coordination, optimize resource utilization, and drive continuous innovation in the rapidly evolving IoT landscape.

Key words: *CI/CD (Continuous Integration and Delivery), automation, cloud infrastructure, real-time monitoring, cybersecurity, scaling, software lifecycle management, hardware-software integration.*