

ІНТЕГРАЦІЯ ТА СУПРОВІД ПРОЕКТНИМИ КОМАНДАМИ ІНФОРМАЦІЙНО- ТЕХНОЛОГІЧНИХ ПРОДУКТІВ ВПРОДОВЖ ЖИТТЄВОГО ЦИКЛУ

О. М. Верес, В. В. Пасічник, Н. Е. Кунанець

Національний університет “Львівська політехніка”, кафедра інформаційних систем та мереж
Україна, м. Львів, вул. С. Бандери, 12

E-mail: oleh.m.veres@lpnu.ua, ORCID: 0000-0001-9149-4752; E-mail: vpasichnyk@gmail.com,
ORCID: 0000-0003-3007-2462; E-mail: nek.lviv@gmail.com, ORCID: 0000-0003-3007-2462

© Верес О. М., Пасічник В. В., Кунанець Н. Е., 2019

Підтверджено, що інтелектуальні системи розробляти із застосуванням єдиних науково-методичних принципів та низки вимог, формуванню яких передують ґрунтовний та всебічний системний аналіз суб'єктів діяльності. Проаналізовано історію формування автоматизованих систем управління вищим навчальним закладом у Національному університеті “Львівська політехніка”. Подано концепт DevOps як інноваційної парадигми формування та функціонування команд ІТ-проектів. Парадигму DevOps подають автори як комплекс інформаційних технологій, які в системному поєднанні забезпечують високоефективне функціонування інтегрованої команди ІТ-проекту – системних аналітиків, розробників, інтеграторів та експлуатаційників, що реалізує та підтримує його на всіх етапах життєвого циклу.

DevOps – це набір комунікаційних практик, які сприяють швидкому і надійному створенню, тестуванню та впровадженню передових інформаційно-технологічних продуктів у промислову експлуатацію. Концепція DevOps основана на побудові ефективної культури комунікування, яка забезпечує продуктивну співпрацю між усіма членами команди проекту. Культура комунікування є одним з найважливіших факторів успіху втілення концепції DevOps. Формування та реалізація концепції спільної відповідальності, прозорості та швидшого зворотного зв'язку є надійним базисом функціонування високофахової команди DevOps.

Команди, які працюють над інформаційно-технологічним проектом (ІТ-проектом), доволі часто не дотримуються принципів внутрішньокомандного “системного мислення”. За внутрішньою природою комплекс інформаційних технологій DevOps передбачає використання саме такої методології “системного мислення”, що загалом позитивно впливає на загальну успішність роботи команди, яка реалізовує проект.

Концепція DevOps усуває бар'єри між традиційно ізольованими групами розробників та фахівців, що експлуатують систему, об'єднуючи їх в єдину комплексну команду. Згідно з концепцією DevOps члени команди працюють з метою якнайповнішої оптимізації та підвищення продуктивності праці розробників, а також підвищення надійності продукту під час його промислової експлуатації.

Ключові слова: інтелектуальні системи, інформаційна сумісність систем, інформаційні технології DevOps, культура комунікування, інтеграція та супровід інформаційно- технологічних продуктів, проектні команди.

INTEGRATION AND SHIPPING BY PROJECT TEAMS OF INFORMATION TECHNOLOGICAL PRODUCTS FOR THE PERIOD OF THE LIFE CYCLE

O. M. Veres, V. V. Pasichnyk, N. E. Kunanets

National University "Lviv Polytechnic", Department of Information Systems and Networks

12, vul. S. Bandery, Lviv, Ukraine

E-mail: oleh.m.veres@lpnu.ua, ORCID: 0000-0001-9149-4752; E-mail: vpasichnyk@gmail.com,

ORCID:0000-0003-3007-2462; E-mail: nek.lviv@gmail.com, ORCID: 0000-0003-3007-2462

© Veres O. M., Pasichnik V. V., Kunanets N. E., 2019

The authors noted that the development of intellectual systems was carried out on the basis of the application of uniform scientific and methodological principles and a number of requirements, the formation of which is preceded by a thorough and comprehensive system analysis of actors. The history of the formation of automated control systems for higher educational institutions at the National University "Lviv Polytechnic" is analyzed. The concept of DevOps is presented as an innovative paradigm for the formation and operation of IT project teams. The DevOps paradigm is presented by the authors as an information technology complex, which in a system combination provides a highly effective operation of the integrated team of the IT project – system analysts, developers, integrators and operators, which implements and supports it at all stages of the life cycle.

DevOps is a set of communication practices that contribute to the rapid and reliable creation, testing and implementation of advanced information technology products for industrial exploitation. The DevOps concept is based on building an effective communication culture that provides productive collaboration between all members of the project team. The culture of communication is one of the most important factors in the success of DevOps concept implementation. Formation and implementation of the concept of joint responsibility, transparency and faster feedback is a reliable basis for the operation of the high-performance team DevOps. Teams that work on an information technology project (IT project), often do not adhere to the principles of intra-system "thinking". By the internal nature of the DevOps information technology complex, the use of such a methodology of "system thinking", which generally has a positive effect on the overall success of the team that implements the project.

The DevOps concept eliminates the barriers between traditionally isolated groups of developers and those who operate the system, integrating them into a single integrated team. According to the DevOps concept, team members work to maximize optimization and increase productivity of developers, as well as increase the reliability of the product during its industrial exploitation.

Key words: intellectual systems, information interoperability systems, DevOps information technologies, communication culture, integration and support of information technology products, project teams.

Постановка проблеми

Численні галузі економіки потребують надійності, видимості та простежуваності програмних продуктів, що використовуються для забезпечення безпеки та надійності їх використання. Їх представники вимагають від виробників супроводу експлуатації програмних продуктів, підтвердження відповідності розроблених інтелектуальних систем встановленим практикам. Розподіл функціональних обов'язків між розробниками та експлуатаційниками часто є перешкодою для ефективної комунікації при створенні та впровадженні програмних продуктів.

Аналіз останніх досліджень

Напрямок досліджень з DevOps недостатньо представлений в наукових публікаціях, хоча слід зазначити, що дослідники не ігнорували роль атрибутів DevOps у діяльності підприємства, але

детально не досліджували цих ознак. У результаті ідентифікація ключових незалежних атрибутів, які впливають на реалізацію DevOps, залишається не дослідженою. У деяких дослідженнях в області DevOps розглядали концепції, принципи, практики, переваги та проблеми, пов'язані з реалізацією концепції DevOps.

А. Дук, Р. Пеннерс, Г. Ліхтер відзначають відсутність єдиного визначення DevOps [1]. Достатньо детальний огляд літератури в області DevOps здійснили F. Erix та ін. [2]. М. Сакс [3] проаналізував основні поняття, цілі, принципи, компоненти та переваги DevOps, порівнюючи його з технологією гнучкого розроблення програмного забезпечення. У низці публікацій [4–7] DevOps порівнюють із іншими методологіями проектної діяльності. Л. Лвактер та ін. [8] проаналізували основні компоненти DevOps, визначивши його концептуальні засади. С. Амбер [9] та Д. ДеГрадів [10] наполягали на необхідності впровадження на підприємствах гнучкої методології, що ґрунтується на співпраці між членами команди, проілюстрували переваги її використання, зокрема збільшення доходів, скорочення витрат. А. Патварвган та ін. [11] підкреслює переваги використання гнучкої методології під час впровадження DevOps на підприємстві, що допоможе досягти високих результатів.

У наукових розвідках представлено результати дослідження окремих компонентів DevOps, серед яких безперервна інтеграція, безперервне тестування, безперервне розгортання та безперервний моніторинг. К. Гсієг та ін. [12] проаналізували особливості безперервної інтеграції, вважаючи її одним з ключових аспектів DevOps, відзначаючи наявність десяти моделей безперервної інтеграції, які можуть бути використані в процесі реалізації цієї технології на підприємстві. М. Артач та ін. [13] у своєму дослідженні акцентували увагу на інструментах для вирішення питань, пов'язаних з постійним удосконаленням програмних продуктів. Д. Брунео [14] та Д. Цукер [15] подали результати дослідження щодо реалізації технології DevOps з використанням хмарних технологій, що сприяло отриманню якіснішої інформації за менший проміжок часу. К. Коінс [16] сформулював вимоги до ефективної комунікації між різними групами команди і подав модель розробленої інформаційної системи для вирішення проблеми комунікації. Ю. Перец та співавтори [17] запровадили інструментарій, який забезпечує ефективне вирішення завдань, покладених на команду. Інструментарій ґрунтується на використанні стохастичних моделей продуктивності, що допомагало рефакторингу програмного забезпечення. С. Банг та співавтори [18] провели експеримент з реалізації трьох проектів веб-розробки з використанням якісного аналізу та визначили, що знання, навички та здібності як під час розроблення програмного забезпечення, так і під час комунікації команди забезпечують співпрацю, командну культуру.

С.Мохамед та співавтори [19] розробили кількісний інструмент, що вимірює рівень зрілості впровадження DevOps в установі.

В. Гупта, П. К. Капур, Д. Лумар [20] запропонували методи вимірювання продуктивності різних атрибутів роботи великих підприємств за впровадження DevOps. Разом з тим, слід зазначити, що особливості концепції DevOps потребують подальших досліджень та використання напрацьованого досвіду.

Мета статті – проаналізувати процеси інтеграції та супроводу проектними командами інформаційно-технологічних продуктів впродовж життєвого циклу.

Проект “Інформаційна система управління діяльністю ЗВО”

Інформаційну систему управління діяльністю ЗВО (ІСУ ЗВО) загалом розробляли на основі застосування єдиних науково-методичних принципів, що містять такі вимоги:

- **використання єдиної інформаційної бази, що включає уніфіковані системи документів;**
- **необхідність структурно-функціональної сумісності систем, підсистем і задач ІСУ ЗВО різного рівня.**

Виконання зазначених вимог передбачає практичну реалізацію принципу інформаційної сумісності систем, підсистем і задач різного рівня в ІСУ ЗВО. Поняття “інформаційна сумісність” передбачає використання на всіх рівнях ІСУ ЗВО єдиної системи показників і єдиної уніфікованої системи класифікації та ідентифікації показників і об'єктів управління. ІСУ ЗВО повинна

враховувати можливість спільної роботи з опрацювання інформаційних ресурсів з інформаційними системами вищих рівнів (Міністерство освіти і науки України, обласні державні адміністрації тощо) для зменшення затрат з адаптації та непередбачених втрат даних.

Основою побудови ІСУ ЗВО повинен стати ґрунтовний та всебічний системний аналіз суб'єктів діяльності. Як елементи схеми реалізації вимог інтегрованого опрацювання даних при управлінні ЗВО можна виділити лінійно-функціональні організаційні комплекси, що безпосередньо орієнтуються на виконання стратегічних цілей ЗВО, і програмно-цільові комплекси, які забезпечують інтеграційні процеси при підготовці спеціалістів і проведення наукових досліджень.

Лінійно-функціональні комплекси ЗВО сформовано для таких видів діяльності, як навчально-методична, науково-дослідна, виробничо-господарська і управлінська, які складаються з відносно постійних і регулярних процесів ЗВО.

Програмно-цільовий блок ЗВО сформовано для таких видів діяльності, що вирізняються кінцевими цілями, які залежать від багатьох результатів і забезпечуються підрозділами різних служб. До таких видів діяльності можна зарахувати підготовку спеціалістів і проведення наукових досліджень у певній предметній області.

У складі обчислювального центру Львівської політехніки до 1998 року функціонував сектор “Автоматизовані системи управління вищим учбовим закладом” (АСУ ВУЗ). Цей підрозділ за сферою діяльності в мережах ЗВО підпорядковувався Першому проректору. Сектор АСУ ВУЗ містив два підрозділи: бюро розробок і бюро впровадження та експлуатації.

Основними функціональними задачами бюро впровадження та експлуатації були такі:

- **супровід промислової експлуатації підсистем АСУ ВУЗ;**
- **адаптація та впровадження рекомендованих Міністерством освіти підсистем АСУ ВУЗ.**

Кадровий потенціал бюро становили фахівці з кваліфікацією інженера-програміста різних категорій, прикладного математика, інженера-системотехніка. Бюро супроводжувало такі підсистеми та задачі АСУ ВУЗ: “Заробітна плата”, “Успішність”, “Кадри”, “Облік матеріальних цінностей”, “Облік виконання завдань”, “Абітурієнт” та “Викладач очима студента”.

Основні функціональні задачі бюро розробок такі:

- **розроблення нових та модифікація наявних підсистем і задач АСУ ВУЗ;**
- **розроблення супровідної документації;**
- **тестування прототипів підсистем і задач.**

Кадровий потенціал бюро становили фахівці з кваліфікацією інженера-програміста різних категорій, прикладного математика, інженера-системотехніка. Співробітниками бюро було розроблено підсистеми для автоматизації таких видів діяльності як навчально-методична, виробничо-господарська і управлінська.

Для проектування інформаційних систем (ІС) застосовували класичні підходи до проектування. Проект ІС проходив всі основні проектні стадії.

Підставою для діяльності та запуску робіт (стадія “запуск”) є наказ ректора на розроблення відповідної підсистеми АСУ ЗВО.

Проектування нової підсистеми (ІС) розпочинається з формування колективу розробників (зі складу співробітників бюро розробки) та експертів (фахівців певного виду діяльності ЗВО). Дана група проводила передпроектне обстеження (стадія “обстеження”), загальний аналіз проблемної ситуації у закладі вищої освіти, розробляла загальне обґрунтування доцільності створення нової підсистеми.

У рамках системного аналізу предметної області (стадія “концепція, ТЗ”) досліджували вимоги різних типів користувачів, формували рекомендації з розроблення підсистеми. Проектна група готувала для керівника АСУ ВУЗ (ректор, перший проректор) технічне завдання (ТЗ) на проектування підсистеми чи задачі. Технічне завдання розробляли з врахуваннями вимог до системи та наявних ресурсів, а також обговорювали з провідними фахівцями з цієї проблематики, що працювали у Львівській політехніці. Затверджене технічне завдання ставало підґрунтям розроблення комплексу засобів функціонування підсистеми чи задачі АСУ ВУЗ (рис. 1).

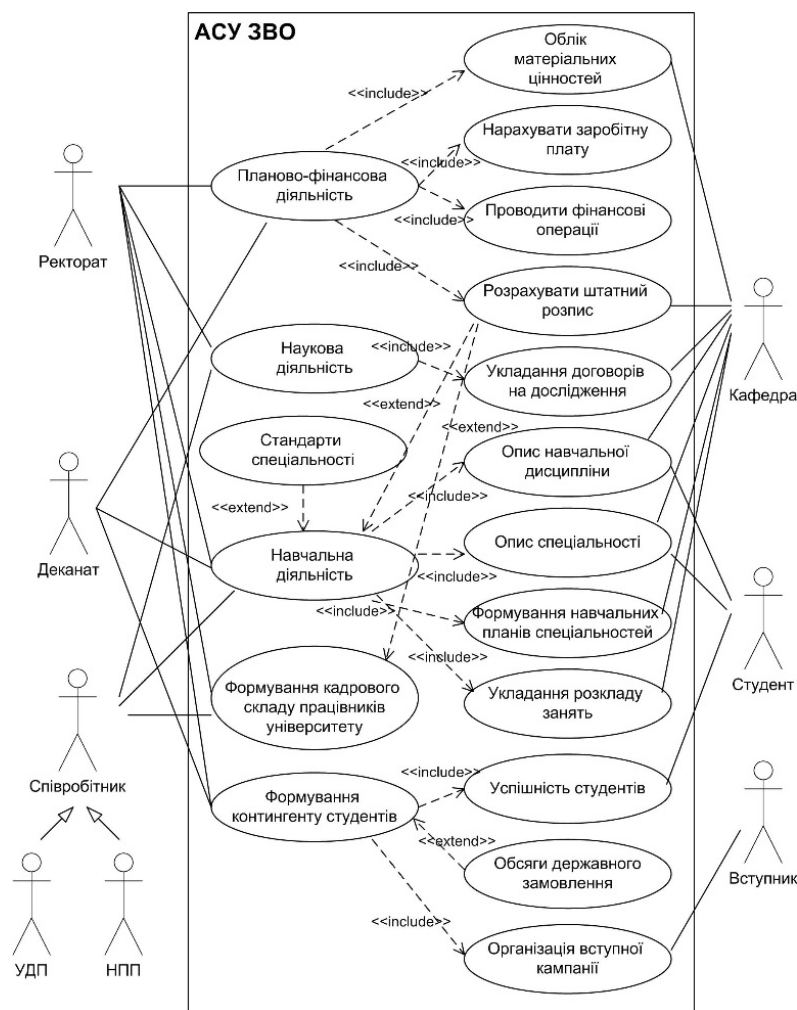


Рис. 1. Діаграма варіантів використання АСУ ВУЗ

У межах ескізного проекту розробляли архітектуру майбутньої підсистеми, що враховувала загальні вимоги та принципи побудови ІС АСУ ВУЗ. Потім проводили програмування (кодування), компонування та тестування розробленого пілотного проекту підсистеми. Результати тестувань стали підґрунтям розроблення виправлень та доповнень до ТЗ. Для підготовки до впровадження нової підсистеми чи задачі, розроблялися остаточні варіанти робочої документації проекту.

Група, що визначає вимоги користувачів та зовнішні специфікації системи (способи та набір правил взаємодії зі зовнішнім середовищем), працює постійно протягом всього життєвого циклу інформаційної системи, виконуючи корегувальні та контрольні функції.

У Львівській політехніці було розроблено та затверджено Техно-робочий проект першої черги підсистем АСУ ВУЗ, який відповідав вимогам ГОСТ 24.xxx “Единая система стандартов автоматизированных систем управления”.

До першої черги належать такі підсистеми і задачі, що розроблені в бюро: “Кадри”, “Облік матеріальних цінностей”, “Облік виконання завдань”, “Абітурієнт”, “Викладач очима студента”, “Контингент студентів”.

Підсистеми “Кадри”, “Облік матеріальних цінностей” та “Абітурієнт” було успішно переведено та адаптовано на нові програмно-апаратні планформи.

Розроблено задачу “Вступні іспити” підсистеми “Абітурієнт” – автоматизація проведення фахового вступного випробування (рис. 2). Особливістю нової методики проведення іспиту є надання кожному абітурієнту індивідуального завдання, яке сформовано системою залежно від змісту випробування. Автоматизація процесу проведення та опрацювання результатів вступних фахових випробувань дає можливість провести глибокий комплексний аналіз результатів складання іспитів, що дасть змогу оцінити ефективність профорієнтаційної роботи у всіх її аспектах.

DevOps – це набір комунікаційних практик, які сприяють швидкому і надійному створенню, тестуванню та впровадженню передових інформаційно-технологічних продуктів у промислову експлуатацію. Концепція DevOps основана на побудові ефективної культури комунікування, яка забезпечує продуктивну співпрацю між усіма членами команди проекту. Культура комунікування є одним з найважливіших факторів успіху втілення концепції DevOps. Формування та забезпечення реалізації концепції спільної відповідальності, прозорості та швидшого зворотного зв'язку є надійним базисом функціонування високофаксової команди DevOps.

Команди, які працюють над інформаційно-технологічним проектом (ІТ-проектом), доволі часто не дотримуються принципів внутрікомандного “системного мислення”. За внутрішньою природою комплекс інформаційних технологій DevOps передбачає використання саме такої методології “системного мислення”, що загалом позитивно впливає на загальну успішність роботи команди, яка реалізовує проект.

Суть концепції DevOps полягає в зміні ідеології процесів мислення членів команди і змушує розглядати всі процеси реалізації проекту цілісно, руйнуючи при цьому бар'єри між етапом розроблення – Development і операційного супроводу – Operations. Втілення принципів концепції DevOps полягає в ефективній реалізації тези щодо членів команди проекту: працюйте злагоженіше, швидше, розумніше та ефективніше.

Швидкості досягають завдяки налагодженню комунікації у команді з суттєвим скороченням часу реалізації фази зворотного зв'язку. DevOps – це поєднання комунікаційних принципів, підходів і засобів, які підвищують здатність компаній створювати інформаційні системи, застосунки і сервіси з високою оперативністю, надійністю та якістю. Із втіленням принципів DevOps розроблення і оптимізація інформаційно-технологічних продуктів відбувається швидше, ніж при використанні традиційних процесів їх розроблення та усталеного традиційного управління ІТ-інфраструктурою. Завдяки підвищенню оперативності дій та швидкості реалізації ідей компанії можуть підвищувати якість обслуговування клієнтів і ефективніше конкурувати на ринку.

Повна прозорість комунікації дає змогу команді DevOps мінімізувати втрати часу та прискорити вирішення проблем, забезпечує вищі якість і стабільність результатів під час виконання проекту. Завдяки налагодженню відкритої комунікації команд Dev і Ops та чіткому визначенню пріоритетів вдається ефективніше вирішувати проблеми, оперативніше виправляти помилки та ефективніше керувати виконанням позапланових робіт, які зазвичай вагомо впливають на загальну продуктивність команди.

У моделі DevOps практично стираються кордони між командами розробників та замовників (експлуатаційників). Зазвичай ці дві команди об'єднуються в одну, фахівці ІТ-галузі спільно працюють над усіма фазами життєвого циклу інформаційно-технологічного продукту: від розроблення та тестування до розгортання та експлуатації, що потребує від членів команди здобуття низки специфічних навичок, не обмежуючись виключно компетентностями в полі вузької спеціалізації. У команді DevOps групи контролю якості та безпеки активно комунікують та взаємодіють з групами розроблення та експлуатації впродовж всього життєвого циклу інформаційно-технологічного продукту: інформаційних систем, технологій, застосунків та сервісів, різнотипових програмних продуктів.

Команди DevOps використовують спеціальні методи, набір інформаційних технологій та інструменти, які допомагають працювати із застосунками та розвивати і вдосконалювати їх зі збереженням високої оперативності та надійності кінцевого продукту. За допомогою цих інструментів фахівці DevOps можуть самостійно вирішувати такі завдання, для яких зазвичай потрібна допомога фахівців-вузьких спеціалістів, що загалом суттєво підвищує швидкість виконання спільних робіт.

Переваги запровадження концепції DevOps як інноваційної парадигми формування та супроводу інформаційно-технологічних проектів

Швидкість

Члени команди DevOps працюють із високою швидкістю, щоб якнайшвидше запроваджувати нові можливості застосунків, технологій та системи для клієнтів, краще адаптуватися до мінливих

ринків і ефективніше досягати намічених цілей проекту. Модель DevOps допомагає групам розробників та експлуатаційників, об'єднаних в одну команду, ефективніше (якісніше та швидше) досягти цілей проекту. Наприклад, використання мікросервісів і реалізація принципів неперервного допрацювання програмних продуктів та інформаційно-технологічних сервісів дають змогу команді швидше опанувати над ними контроль, а надалі – оперативно їх адаптувати та оновлювати.

Швидке допрацювання

Підвищується частота і швидкість реалізації внутрішньокорпоративних процесів з метою якнайшвидшого оновлення і покращення якості відповідних продуктів. Що швидше розробляють нові функціональні можливості і вносять виправлення, то оперативніше вдається реагувати на нагальні потреби клієнтів і створювати продукти із суттєвими конкурентними перевагами. Безперервність процесів інтеграції і допрацювання допомагає удосконалити практично всі фази та етапи випуску інформаційно-технологічних продуктів: від специфікації функцій, збирання компонентів до повномасштабного їх розгортання.

Надійність

Щоб надійно і швидко розробляти інформаційно-технологічні продукти і зберігати лояльність кінцевих користувачів, команда DevOps постійно контролює якість оновлень застосунків та зміни в інфраструктурі. Реалізація методів безперервної інтеграції та допрацювання допомагає протестувати функціональність і безпеку кожної з внесених змін, а постійний моніторинг дає змогу відстежувати параметри продуктивності продукту в режимі реального часу.

Масштабування

Керування процесами розроблення інформаційно-технологічних продуктів: інформаційних систем, технологій, застосунків та сервісів та підтримання і вдосконалення інформаційно-технологічної інфраструктури забезпечує стабільне їх функціонування за умови довільного масштабування. Дотримання логічної послідовності реалізації процесів сприяє ефективнішому управлінню складними та швидкозмінними системами, що загалом суттєво зменшує ризики. Наприклад, правильно сформована інфраструктура сприяє ефективнішому управлінню середовищами розроблення, тестування і робочим середовищем, а також забезпечує їх відтворюваність.

Оптимізована спільна робота

Формування високоефективних команд у межах комунікаційної моделі DevOps природно прищеплює членам команди такі цінності, як співпричетність і відповідальність. Групи розроблення та експлуатації тісно взаємодіють між собою в одній команді, розподіляють функціональні обов'язки і об'єднують певні робочі процеси. Це загалом призводить до зменшення нераціональних дій та економить час (наприклад, зменшує час передавання справ від розробників до інженерів з експлуатації та усуває необхідність написання коду з урахуванням середовища, в якому він буде експлуатуватися).

Безпека

Команди DevOps розвиваються швидко, зберігаючи повний контроль над усіма процесами і дотримуючись при цьому усіх технологічних вимог та стандартів. Модель DevOps впроваджують із дотриманням принципів безпеки використовуючи при цьому точні налаштування, а також втілюючи ефективні методи управління конфігураціями.

Висновки

Концепція DevOps усуває бар'єри між традиційно ізольованими групами розробників та фахівців, що експлуатують систему, об'єднуючи їх в єдину комплексну команду. Згідно з концепцією DevOps члени команди працюють з метою якнайповнішої оптимізації та підвищення продуктивності праці розробників, а також підвищення надійності продукту під час його промислової експлуатації. Це забезпечується зокрема добре організованою комунікацією, а також високою якістю послуг, що надаються клієнтам. Фахівці з DevOps беруть на себе повну відповідальність за надані сервіси та створені програмні продукти, тісно співпрацюючи з групою експлуатації. У компаніях, які використовують модель DevOps, незалежно від їх організаційної

структури, співробітники супроводжують весь життєвий цикл інформаційно-технологічного продукту, від фази створення до фази його промислової експлуатації.

Список літератури

1. A. Dyck, R. Penners, H. Lichter, Towards definitions for release engineering and devops, Third International Workshop on Release Engineering, IEEE Press, New Jersey, USA. (2015) 3–3.
2. F. Erich, C. Amrit, M. Daneva, A mapping study on cooperation between information system development and operations, International Conference on Product-Focused Software Process Improvement, Springer International Publishing. (2014) 277–280.
3. M. Sacks, Devops principles for successful web sites, Pro Website Development and Operations, Apress, (2012) 1–14.
4. A. Patwardhan, J. Kidd, T. Urena, A. Rajgopalan, Embracing Agile methodology during DevOps Developer Internship Program, arXiv preprint arXiv:1607.01893. (2016).
5. C. Y Hsieh, C. T. Chen, Patterns for Continuous Integration Builds in Cross-Platform Agile Software Development, Journal of Information Science and Engineering, (2015) 31(3) 897–924
6. S. N. Mullaguru, Changing Scenario of Testing Paradigms using DevOps–A Comparative Study with Classical Models, Global Journal of Computer Science and Technology. (2015) 15(2).
7. S. W. Ambler, Disciplined agile delivery and collaborative DevOps, Cutter IT Journal. (2011) 24(12) 18.
8. L. E. Lwakatare, P. Kuvaja, M. Oivo, Dimensions of DevOps, International Conference on Agile Software Development, Springer International Publishing, Helsinki, Finland. (2015) 212–217
9. S. W. Ambler, Disciplined agile delivery and collaborative DevOps, Cutter IT Journal. (2011) 24(12) 18.
10. D. DeGrandis, Devops: So you say you want a revolution?, Cutter IT Journal, (2011), 24(8), 34–39
11. A. Patwardhan, J. Kidd, T. Urena, A. Rajgopalan, Embracing Agile methodology during DevOps Developer Internship Program, arXiv preprint arXiv:1607.01893. (2016).
12. C.Y Hsieh, C. T. Chen, Patterns for Continuous Integration Builds in Cross-Platform Agile Software Development, Journal of Information Science and Engineering, (2015) 31(3) 897–924
13. M. Artač, T. Borovšak, E. Di Nitto, M. Guerriero, D.A. Tamburri, Model-driven continuous deployment for quality DevOps, 2nd International Workshop on Quality-Aware DevOps. ACM, New York, USA. (2016) 40–41.
14. D. Bruneo, T. Fritz, S. Keidar-Barner, P. Leitner, F. Longo, C. Marquezan, A. Roth, CloudWave: Where adaptive cloud management meets DevOps, 2014 IEEE Symposium on Computers and Communications (ISCC), IEEE, Funchal. (2014) 1–6.
15. D. Cukier, DevOps patterns to scale web applications using cloud services, 2013 companion publication for conference on Systems, programming, & applications: software for humanity, ACM, New York, USA. (2013) 143–152.
16. C. A. Cois, J. Yankel, A. Connell, Modern DevOps: Optimizing software development through effective system interactions, IEEE International Professional Communication Conference. IEEE, Pittsburg, PA. (2014) 1–7.
17. J. F. Perez, W. Wang, G. Casale, Towards a devops approach for software quality engineering, Workshop on Challenges in Performance Methods for Software Development, ACM, New York, USA. (2015) 5–10.
18. S. K. Bang, S. Chung, Y. Choh, M. Dupuis (2013) A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps, ACM, New York, USA. 2nd annual conference on Research in information technology, (2013). 6162.
19. Mohamed S. I. (2016). DevOps Maturity Calculator DOMC – Value oriented approach, International Journal of Engineering Research & Science, 2(2) 25–35
20. Viral Gupta, P. K. Kapur, Deepak Kumar (2017) Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling, Information and Software Technology . Information and Software Technology. Volume 92, December 2017, Pages 75–91