

УДК 004.9: 005.8

Кравченко С.М.

Житомирський державний технологічний університет

Марчук Г.В.

Житомирський державний технологічний університет

ВИКОРИСТАННЯ МЕТОДОЛОГІЇ DEVOPS В УПРАВЛІННІ ІТ ПРОЕКТАМИ

У статті досліджується вплив методології DevOps на оптимізацію процесу розробки нових продуктів/сервісів. Виконується порівняння сучасних методологій розробки програмного забезпечення. Здійснюється процес моніторингу при застосуванні методології DevOps.

Ключові слова: проектне середовище, методології проектного управління, ІТ-проекти, розробка програмного забезпечення, моніторинг.

Постановка проблеми. Існує багато різних методологій і стандартів проектного управління. Щоб максимально ефективно вибудувати процеси управління, необхідно вибрати оптимальний набір практик управління й розробки програмних систем. Тому необхідно проаналізувати сучасні методології розробки програмного забезпечення, які необхідні для максимальної взаємозалежності розробки, експлуатації та супроводження програмних продуктів і сервісів, з метою їх більш швидкого створення й оновлення.

Аналіз останніх досліджень і публікацій. Протягом останніх десяти років стався значний зсув у підході, який кардинально змінив увесь ІТ-ландшафт. Розробники, втомлені від постійного очікування зворотного зв'язку, почали писати програмне забезпечення, яке автоматизує операційні завдання. Водночас фахівці ІТ-підрозділів почали делегувати свої глибокі знання й багатий досвід у програмне забезпечення, яке написано розробниками.

Сьогодні, коли дуже чіткі межі між розробкою та експлуатацією почали зникати, це призвело до прискорення всього життєвого циклу програмного забезпечення, більш коротких сесій забезпечення якості (Quality Assurance) та багатьох інших змін. З'являються нові процеси, такі як управління безперервною інтеграцією й безперервним розгортанням. Багато великих програмних рішень, які розроблені за допомогою методології DevOps, розгортаються по кілька разів на день, а не кілька разів на рік, як було раніше.

Ця тенденція в автоматизації робочих процесів прогресувала поступово протягом останніх 10 років, поки програмне забезпечення та процеси не досягли своєї зрілості. І вплив, який DevOps уже робить і буде чинити в майбутньому на

інформаційні технології підприємств, необхідно ретельно проаналізувати з погляду всіх залучених учасників цього процесу.

Натепер здійснення моніторингу є однією з найбільш недооцінених галузей під час прийняття методології DevOps. Часта зміна програмного коду змінює підхід до роботи служби моніторингу, роблячи щоденною нормою здійснення таких завдань, як контроль передачі потоків у режимі реального часу, відтворення історії подій, просунута візуалізація тощо. Здійснення постійного та якісного моніторингу стає критично важливим компонентом усіх додатків [5].

Рух DevOps зародився у 2009 році, але його витоки сягають ще глибше. Так, фахівці як приклад інструментарію-прабатька називають популярний кросплатформний клієнт-серверний додаток Puppet, перша версія якого з'явилася ще у 2005 році.

Пізніше на ринку з'явився аналогічний інструментарій, включаючи Chef, Ansible, SaltStack і багато інших. Крім того, навколо цих інструментів утворилися спільноти; розробники й системні адміністратори акумулювали свої знання у вигляді «рецептів», що дають змогу легко і швидко налаштувати програмне забезпечення незалежно від основного дистрибутива Unix.

Постановка завдання. За допомогою методології DevOps оптимізувати процес розробки нових продуктів/сервісів, вибрати оптимальний набір практик управління та розробки вимог до інформаційних систем. За допомогою DevOps здійснити процес постійного і якісного моніторингу. Метою роботи є дослідження методології управління проектами для підвищення ефективності її практичного застосування.

Виклад основного матеріалу дослідження. Під час дослідження основних методологій розробки програмного забезпечення проаналізовано ітеративний підхід до створення та експлуатації програмних систем і сервісів.

Що призвело до появи DevOps? Потрібно спочатку розглянути розробку програмного забезпечення традиційним способом з використанням каскадної моделі (Waterfall).

Сутність каскадної моделі (рис. 1) – поетапне виконання проекту, причому робота над кожним наступним етапом починається тільки після закінчення попереднього.

Але каскадний підхід не працював: команди провалювали терміни й не вкладалися в бюджет. Метод водоспаду не брав до уваги проблеми, що виникають, затримки і збої, мінливі вимоги замовника та навколишнього середовища. Потрібно було шукати альтернативу й міняти процес роботи: регулярно оглядатися назад, аналізувати виконану роботу і тут же усувати перешкоди та вносити зміни. Тому з'явилися гнучкі методології Agile та її похідні.

Методологія Agile передбачає лише процес розробки програмного забезпечення. Agile може реалізуватись різними способами, серед яких – eXtreme Programming, Scrum, Kanban, Lean Software Development тощо.

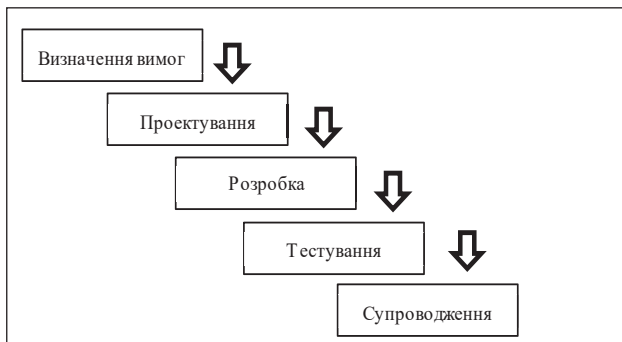


Рис. 1. Процес розробки продукту за методологією Waterfall

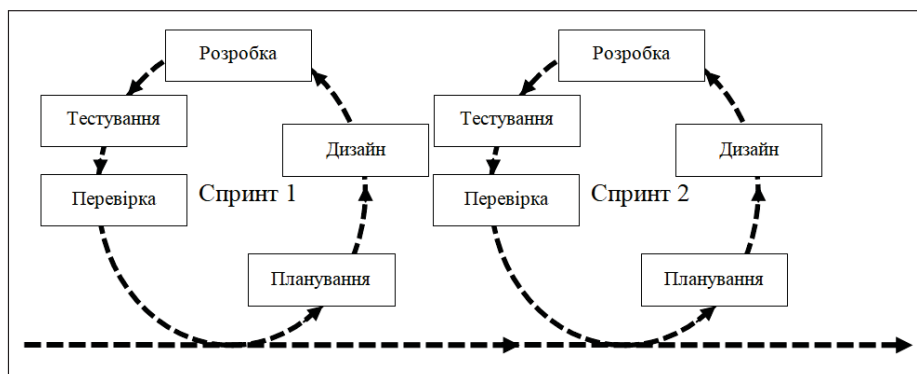


Рис. 2. Процес розробки продукту за гнучкою методологією Agile

Agile – ітеративна модель розробки, в якій програмне забезпечення створюється інкрементно із самого початку проекту, на відміну від каскадних моделей, де код доставляється в кінці робочого циклу (рис. 2).

Основа гнучкої методології – розбиття проектів на маленькі робочі частинки, які називаються користувачькими історіями. Згідно з пріоритетністю, завдання вирішують у рамках коротких двотижневих циклів (ітерацій).

Особливість Agile в тому, що розробник повністю вільний у своїх діях. Це дає команді низку переваг:

- спрощується комунікація в команді;
- покращується взаєморозуміння між її членами;
- прискорюється загальний процес розробки.

Дейв Томас, один із авторів початкового «Маніфесту гнучкої методології розробки програмного забезпечення», вважає, що цінності гнучкої розробки, закладені в Маніфест, так і не були втілені, а саме позначення Agile перетворилося в маркетинговий термін, позбавлений реального змісту. Так, Маніфест свого часу зламав кайдани на розумі розробників, розвінчавши принципи розробки 80-х і 90-х, але далі процес пішов у непередбаченому напрямі, перетворивши Agile в набір жорстких вимог, яким розробники намагаються сліпо відповідати, незалежно від їх застосування в кожному конкретному випадку.

У подальшому методологія, яка буде активно обговорюватись, стане DevOps, можливості якого багато хто поки не усвідомлює повною мірою. Не можна сказати, що DevOps заперечує Agile: його, скоріше, можна розглядати як розвиток не цілком реалізованих принципів гнучкої розробки. Ключова особливість підходу DevOps – активна взаємодія розробників (Development), тестувальників (QA) і служб експлуатації та підтримки (Operations).

Методологія DevOps у буквальному (акронім DevOps походить від поєднання двох англійських слів – development та operations) та образному сенсах є злиттям двох раніше роз'єднаних процесів: розробки й експлуатації програмного продукту. Протягом багатьох років ці дві групи були роз'єднані між собою

кордонами, зумовленими як специфічними відмінностями, так і наборами оперованих знань, що особливо було помітно у великих ІТ-організаціях корпоративного масштабу.

Розділення було пов'язане з тим, що увага розробників не поширювалась далі за їхній програмний код, а системні адміністратори хоча й працювали з кодом, наданим їм першою групою, але їхнім завданням було взяти цей код і переконатися в його працездатності для конкретних умов. Повний розрив між цими двома групами зазвичай є основною причиною тривалих циклів забезпечення якості (QA) й мінімальною кількістю виробничих розгортань через побоювання простою або страху просто що-небудь «зіпсувати».

DevOps сфокусована на гранично активній взаємодії й інтеграції в одній команді програмістів, тестувальників та адміністраторів, які синхронізовано обслуговують загальний для них сервіс/продукт. Головна мета цього – створення єдиного циклу взаємозалежності розробки, експлуатації та розгортання програмного забезпечення, щоб у кінці допомагати організаціям (сервісам, стартапам) швидше й безболісніше створювати та оновлювати їхні програмні продукти й сервіси, які експлуатуються в режимі реального часу або «в продакшені».

Орієнтуючись на активну взаємодію та інтеграцію інженерів-програмістів, тестувальників і системних адміністраторів, методологія DevOps базується на ідеї про тісну взаємозалежність розробки та експлуатації програмних продуктів і сервісів, з метою їх швидшого створення й оновлення. Але, щоб такий підхід став успішним, життєво необхідне здійснення всебічного моніторингу в режимі реального часу (рис. 3).

Фахівці нагадують, що для DevOps на першому місці серед пріоритетів стоїть не швидке розгортання, а забезпечення його максимальної надійності без порушень у функціях. До того ж ця методологія, на відміну від Agile, багато

в чому залежна від автоматизації процесів, тому підбираються відповідні інструменти, які будуть уважатися складниками методології DevOps.

Методологія передбачає три головні тези:

1. Оцінюється продуктивність системи загалом, а не відділів або конкретних розробників, адміністраторів, фахівців з якості. Завдання керівника, що впроваджує DevOps, – змусити всіх учасників проекту працювати єдиною командою на досягнення глобальної мети: збільшення цінності продукту для користувачів. Обов'язкова умова – розвивати крос-функціональність, робити співробітників більш універсальними солдатами.

2. Безперервна зворотний зв'язок «справа наліво». Тобто в продукт постійно впроваджуються зміни, відстежується реакція користувачів, готується матеріал для нових завдань, вони йдуть у роботу. Такий «конвеєр» вимагає високого рівня автоматизації процесів тестування й розгортання, створювати його за рахунок нарощування персоналу не правильно.

3. Створення внутрішньокорпоративної культури, що заохочує експерименти. Не можна зрозуміти, як буде оптимально для продукту, без тестування на його кінцевих споживачах. Відповідно, і кожен член команди, і клієнт повинні мислити так: сміливо впроваджувати нове, щоб отримувати найбільш цінний аналітичний матеріал для майбутніх етапів розробки [6].

Автоматизація DevOps. Інструментарій DevOps забезпечив високий ступінь стандартизації, пропонуючи ефективні способи розгортання, налаштування й запуску багатьох серверів за допомогою всього лише кількох автоматизованих інструментів, а не покладаючись на втручання операторів. Усе частіше роль команди за технологічними операціями переорієнтується на розгортання й обслуговування автоматизованих додатків-сервісів, що надаються на вимогу, наприклад, через модель PaaS або кластер контейнерів Linux. Розробники розгортають і масштабують індивідуальні програми

в мережі пристроїв, залишаючи команді системних адміністраторів запуск і масштабування цих мереж. Наприклад, щоб отримати безперервну подачу змін на сервер, рекомендують автоматизувати процеси розгортання, складання, тестування, управління версіями тощо [5].

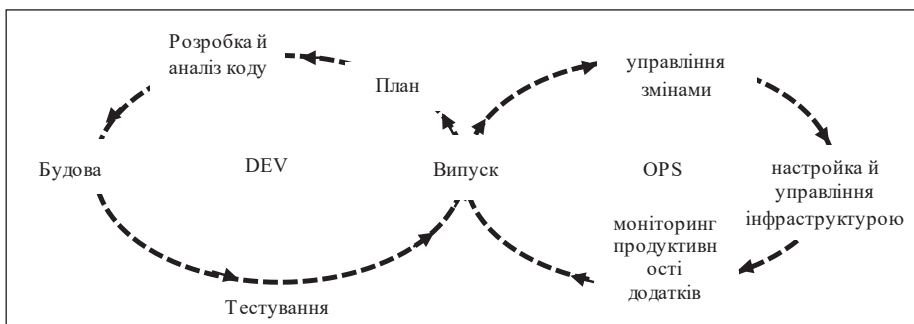


Рис. 3. Процес розробки продукту методологією DevOps

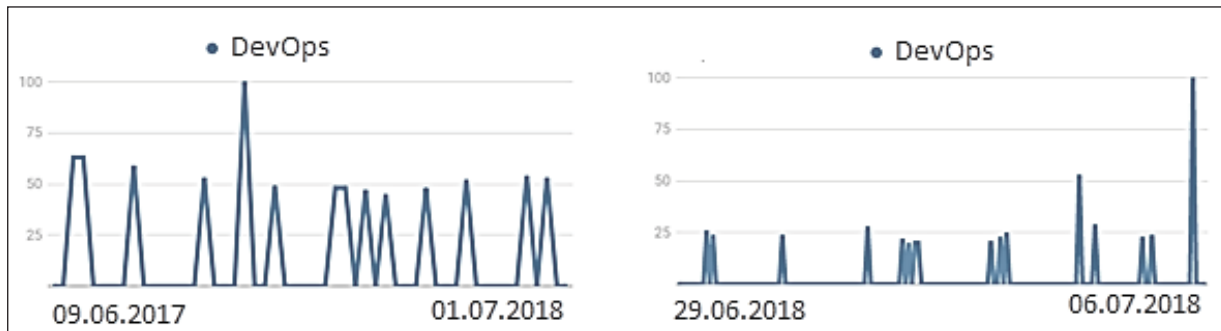


Рис. 4. Динаміка популярності DevOps

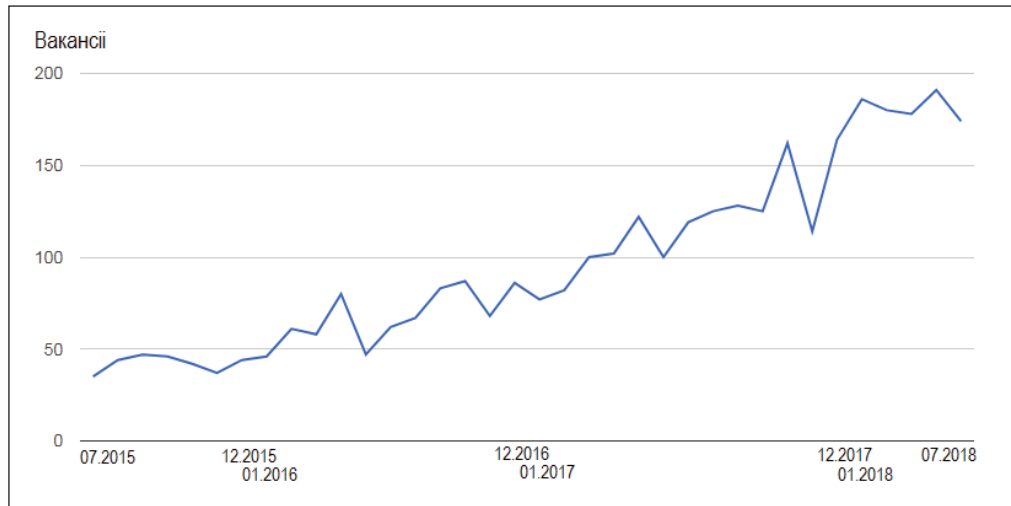


Рис. 5. Тренди jobs.dou.ua: DevOps

Для автоматизованого процесу необхідна підтримка інструментальних засобів (таблиця 1).

Таблиця 1

Інструментальні засоби для автоматизації процесу

Тип інструмента	Назва
Автоматизація інфраструктури	Vcfg2, CFEngine, Chef, CloudFormation, Puppet
Автоматизація розгортання	Capistrano, ControlTier, Func, Glu, RunDeck
Інфраструктура як послуга	Amazon Web Services, CloudStack, IBM SmartCloud, OpenStack, Rackspace
Автоматизація збірки	Ant, Maven, Rake, Gradle
Автоматизація тестування	JUnit, Selenium, Cucumber, easyb

Сучасний моніторинг. У сучасних DevOps-архітектурах комплексного програмного забезпечення існує безліч даних, які необхідно відстежувати. Більше недостатньо відстежувати тільки найпростіші статистичні дані, такі як RAM, CPU й дисковий I/O. Тепер рішення для моніторингу має

працювати з API та передавати дані безпосередньо із самих додатків. Щоб зрозуміти всю цю інформацію, одним із критеріїв пошуку потрібної сучасної системи моніторингу повинні стати можливості аналізу в реальному часі потокових даних, відтворення історії змін і ще кращі засоби візуалізації.

Наступним найбільш важливим способом оцінити хмарну систему моніторингу є кількість і якість підтримуваних модульних інтеграцій. Роботу зі скількома мовами програмування підтримує система моніторингу? Багато старих систем моніторингу працюють лише з кількома, з особливим фокусом на Java і .Net, незважаючи на те що високорівневі мови програмування стають усе більш важливими для розробки корпоративних додатків. Тому потрібно буде знайти нову систему моніторингу, яку можна буде зав'язати на популярні сценарні мови, такі як Python, Ruby, PHP і Go.

DevOps став причиною ретельного документування всіх операційних знань у відкритих стандартах, що дають змогу записати та відстежити всі процеси й метрики протягом часу. Ці знання можна запрограмувати за допомогою логіки, використовуюваної для побудови сітки програм-

ного забезпечення на платформах, таких як PaaS і кластери контейнерів Linux, і їх можна повторювати та поєднувати з іншими.

Динаміка популярності. Сьогодні головним трендом у розробці програмного забезпечення поки ще залишається Agile, але DevOps стає одним із найактивніших трендів починаючи з 2011 року. На рис. 4 представлена динаміка популярності DevOps за останні 12 місяців і 7 днів по Україні за результатами Google Trends.

Згідно з Gartner, у 2016 році 25% із найбільших світових ІТ компаній упроваджували DevOps. Експерти пов'язують таке зростання популярності з масовим упровадженням безперервного розгортання.

На порталі jobs.dou.ua можна побачити динаміку зростання вакансій DevOps Engineer (рис. 5).

Висновки. Методологія DevOps продовжить розвиватися. Завжди знайдеться місце для нових інструментів, різних структур і крутих трендів, але основа, яка буде їх разом пов'язувати, полягає у виборі й правильному використанні програмного забезпечення для автоматизації. Інструменти для здійснення моніторингу повинні розвиватися паралельно з технічними тенденціями й у підсумку зуміти задовольнити потреби як розробників, так і системних адміністраторів, надавши необхідну видимість для прийняття нових технологій, таких як Docker, у виробничому середовищі.

Список літератури:

1. Кеннет С. Рубин. Основы Scrum. Практическое руководство по гибкой разработке ПО. Санкт-Петербург: Вильямс, 2016. 544 с.
2. Стеллман Э., Грин Д. Постигага Agile. Ценности, принципы, методологии. Москва: Манн, Иванов и Фербер, 2018. 448 с.
3. Дэвис Д., Дэниелс К. Философия DevOps. Искусство управления ИТ. Питер, 2017.
4. Хамбл Д., Фарли Д. Непрерывное развертывание ПО. Автоматизация процессов сборки, тестирования и внедрения новых версий программ. Санкт-Петербург: Вильямс, 2016. 432 с.
5. Методология DevOps и её влияние на облачные системы мониторинга. URL: <https://networkguru.ru/metodologiya-devops> (дата звернення: 20.05.2018).
6. DevOps – обзор методологии. URL: <https://blog.sibirix.ru/2014/11/19/devops> (дата звернення: 22.05.2018).

ИСПОЛЬЗОВАНИЕ МЕТОДОЛОГИИ DEVOPS В УПРАВЛЕНИИ ПРОЕКТАМИ

В статье исследуется влияние методологии DevOps на оптимизацию процесса разработки новых продуктов/сервисов. Выполняется сравнение современных методологий разработки программного обеспечения. Осуществляется процесс мониторинга при применении методологии DevOps.

Ключевые слова: проектная среда, методология проектного управления, ИТ-проекты, разработка программного обеспечения, мониторинг.

USE OF DEVOPS METHODOLOGY IN PROJECT MANAGEMENT

The article explores the impact of DevOps methodology on optimizing the process of developing new products/services. Comparison of modern software development methodologies is carried out. The monitoring process is implemented using the DevOps methodology.

Key words: project environment, project management methodology, IT projects, software development, monitoring.