DOI 10.31891/2307-5732-2023-325-5-182-191
УДК 004.056.5:621

ПРАВОРСЬКА НАТАЛІЯ
Хмельницький національний університет
https://orcid.org/0000-0001-6001-3311
e-mail: margana2000007@gmail.com
МАРТИНЮК ВАЛЕРІЙ
Хмельницький національний університет
https://orcid.org/0000-0001-5758-4244
e-mail: martynyuk.valeriy@gmail.com

КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА ДОПОМОГОЮ СИНХРОННОГО ПІДХОДУ: ОСНОВНІ ПРОЦЕСИ ТА ІНСТРУМЕНТИ ДЛЯ ЕФЕКТИВНОЇ РЕАЛІЗАЦІЇ DEVOPS

Дана стаття представляє на розгляд метод синхронного виконання етапів життєвого циклу (ЖЦ) програмного забезпечення (ПЗ) в рамках DevOps. Даний метод націлений на підвищення продуктивності при розробці та впровадженні програмних продуктів (ПП). Проводиться ретельне вивчення теоретичних питань, а саме аспектів DevOps та синхронного виконання етапів ЖЦ. Розглядається впровадження власного методу, який базується на поєднанні команд розробки та інформаційно-технічного обслуговування (експлуатаційників – команда ІТО), а також автоматизації процесів і безперервному забезпеченні якості ПП. Для вимірювання ефективності одночасного виконання етапів ЖЦ в статтю включено аналіз метрик, а також як запропонований метод буде впливати на швидкість розв'язання проблем та відповідності розроблених проєктів заданим вимогам та характеристикам. Вивчення наведених випадків та прикладів успішних проектів, у яких був впроваджений синхронний підхід в рамках DevOps, дозволило зробити висновки щодо практичного застосування даного методу, а також його адаптації для проектів різних розмірів та галузей.

Ключові слова: *DevOps, операції розробки, конструювання програмного забезпечення, програмний продукт, життєвий цикл, ефективність, процес розробки, впровадження, синхронне виконання.*

PRAVORSKA NATALYA
Khmelnitsky national university, Ukraine
MARTYNYUK VALERIY
Khmelnitsky national university, Ukraine

DEVELOPING SOFTWARE USING A SYNCHRONOUS APPROACH: KEY PROCESSES AND TOOLS FOR EFFECTIVE DEVOPS

This article presents for consideration the method of synchronous execution of the stages of the software life cycle within the framework of DevOps. This method is aimed at increasing productivity in the development and implementation of software products. A thorough study of theoretical issues is carried out, namely aspects of DevOps and synchronous execution of life cycle stages. The implementation of a proprietary method is considered, which is based on the combination of development teams and information and technical service (operators), as well as process automation and continuous quality assurance of the software product. To measure the effectiveness of the simultaneous execution of life cycle stages, the article includes an analysis of metrics, as well as how the proposed method will be influence the speed of solving problems and compliance of the developed projects with the specified requirements and characteristics. The study of the given cases and examples of successful projects in which a synchronous approach was implemented within the framework of DevOps allowed us to draw conclusions about the practical application of this method, as well as its adaptation for projects of various sizes and industries.

DevOps is a software development practice that focuses on bringing development and IT teams together to improve productivity, product quality, and cross-team collaboration. Important aspects here are collaboration, automation, integration, deployment, monitoring, feedback, continuous improvement, and elasticity. In DevOps, mutual assistance and sharing of knowledge between teams helps to solve joint tasks. Reducing the number of errors, shortening the time of introducing changes, increasing the efficiency of work processes is possible thanks to the automation of software development processes. Regular code changes and continuous deployment of software in production are ensured by integration and deployment.

On the example of successful projects where a synchronous approach within the framework of DevOps was used, a study of the application of the method was conducted and as a result of the study it was found that such an approach can bring significant advantages to organizations. There are special metrics for evaluating the impact of the method on the effectiveness of DevOps processes. The specified metrics were used to measure the efficiency of the synchronous execution of stages. An analysis of the impact of the method on the speed of problem solving and satisfaction with project results was also carried out.

Keywords: *DevOps, development operations, software engineering, software product, life cycle, efficiency, development process, implementation, synchronous execution.*

Постановка проблеми

На сьогоднішній день постійна трансформація технологій, швидкі зміни вимог користувачів та високий рівень конкуренції призвели до того, що відбувається безпрецедентне зростання складності систем під час розробки програмного забезпечення. Подібні тенденції призводять до потреби неспинного прискорення темпів розвитку ПЗ, а це в свою чергу може негативно впливати на якість програмного продукту та продуктивність роботи команди розробників. Саме більш гнучкі методики розробки, такі як Agile та DevOps, стали активно впроваджуватися в індустрії програмного забезпечення, як відповідь на виклики, які постають перед розробниками. Ці методики дають змогу провести швидку адаптацію до нових вимог та гарантовано забезпечувати безперервну поставку ПП. Однак, на практиці, незважаючи на всі

успіхи Agile та DevOps, можливості для подальшого підвищення ефективності та інтеграції команд розробки і експлуатаційників, все ще залишаються необмеженими.

Основними проблемами, які відносяться до даної ситуації, можна вважати: необхідність у більш продуктивній співпраці між командами розробників та експлуатаційниками, для забезпечення швидкого виявлення та розв'язання проблем; зменшення витрат на комунікацію та координацію задач між командами; забезпечення безперервної оптимізації робочих процесів, розробки та випуску ПЗ; підвищення якості ПЗ та зменшення часу його представлення на ринок.

Зважаючи на представлені проблеми, очевидним є потреба у розвитку нових методів та підходів, призначених на подолання поставлених викликів. Це буде сприяти в подальшому підвищенню продуктивності об'єднання команд розробки та експлуатаційників.

Можливими напрямками розвитку можна вважати: для забезпечення більшої гнучкості, здатності до масштабування та інноваційності розробляти нові моделі співпраці між командами розробки та експлуатаційниками; застосування технік синхронного виконання етапів ЖЦ ПЗ, що дозволить командам розробки та ІТО ефективно координувати свої дії, взаємодіяти та адаптуватися до змін у процесі розробки; використання новітніх технологій, а саме: штучний інтелект, автоматизація, хмарні технології та контейнери, для підтримки ефективного синхронного виконання етапів ЖЦ ПЗ; для підтримки адаптації команд розробки та ІТО до нових методів, технік та технологій, забезпечувати неперервною освітою та навчанням.

Аналіз останніх джерел

Важливе місце в області програмної інженерії гнучкі методи розробки ПЗ займають важливе місце і вони висвітлені у сучасній літературі. Однак, вичерпні дослідження, які спрямовані на синхронне виконання етапів ЖЦ ПЗ досі відсутні.

Стосовно DevOps у роботах [1, 3, 8], дослідники звертають увагу на важливість об'єднання процесів розробки та інформаційно-технічного обслуговування і експлуатації в рамках методології DevOps. В роботах проводиться аналіз ключових принципів, практики та переваги застосування DevOps при розробці ПЗ. Щодо Scrum, то у праці [2] автори приділяють увагу значущості гнучкості, адаптації та самоорганізації в процесі розробки. Там же можна побачити систематичний огляд принципів та практик Scrum. Дослідження практики неперервної інтеграції, яка буде передбачати автоматичне злиття коду розробників та його тестування, що допоможе виявити проблеми на ранніх стадіях розробки, відображено в роботі [5]. Фокусування на неперервному розгортанні, яке передбачатиме автоматичний випуск змін до ПЗ в рамках робочих процесів можна побачити в праці [7]. Необхідність наукового підходу до Lean Software та DevOps, вказується авторами у [9], де розглядаються переваги та можливості цих методів у розробці ПЗ.

Зважаючи на вище представлені джерела, можна зробити висновок, що значимість синхронного виконання етапів ЖЦ при розробці ПЗ потребує подальшого вивчення та аналізу. Однак, окремі аспекти синхронного виконання етапів ЖЦ досі залишаються недослідженими, хоча вказані відомі методи, такі як DevOps, Scrum, Continuous Integration та Continuous Deployment і допомагають розуміти загальні принципи гнучкої розробки

Додаткового вивчення потребують не тільки наукові праці щодо взаємодії між командами розробки та експлуатаційниками, а також вплив синхронного виконання етапів на загальну продуктивність та якість ПЗ. В статті представлені нові підходи та методи, які сприятимуть синхронному виконанню етапів ЖЦ, проведено аналіз їх впливу на різні аспекти розробки ПЗ і спроба відкрити нові можливості для підвищення продуктивності команд ІТО та розробників. Основною метою цього дослідження є розробка методу синхронного виконання етапів ЖЦ в рамках DevOps для підвищення інтеграції та ефективності команд розробки та експлуатаційників. Це передбачає визначення ключових принципів та практик, які сприяють більш гнучкому та ефективному виконанню етапів ЖЦ, а також розробку підходу для їх впровадження у різних контекстах розробки ПЗ.

Виклад основного матеріалу

1. Теоретичні аспекти DevOps та синхронного виконання етапів життєвого циклу

1.1. Основні принципи DevOps

DevOps – практика розробки ПЗ, основою якої є об'єднання команд з розробки та інформаційно-технічного обслуговування з метою покращення продуктивності, якості продукту та взаємодії між командами. Важливі аспекти тут: співпраця, автоматизація, інтеграція, розгортання, моніторинг, зворотний зв'язок, безперервне вдосконалення та еластичність. В DevOps взаємодопомога та обмін знаннями між командами допомагає вирішенню разом поставлених завдань. Зниженню кількості помилок, скорочення часу введення змін, підвищення ефективності робочих процесів можливе завдяки автоматизації процесів розробки ПЗ. Регулярне внесення змін у код та безперервне розгортання ПЗ на виробництві забезпечуються інтеграцією та розгортанням.

Виявлення проблем на ранніх стадіях та адаптація ПЗ до вимог користувача можливе в наслідок моніторингу та зворотному зв'язку. Еластичність і масштабованість означають, що ПЗ та інфраструктура повинні бути гнучкими та здатними адаптуватися до змінних вимог. Команди можуть виявляти слабкі місця

та вносити покращення через безперервне вдосконалення, яке буде сприяти аналізу та оцінці процесів, практик та інструментів.

DevOps дає можливість створювати міцну, гнучку співпрацю, що спонукає команди до відповідальності, самостійності та навчання. Ці принципи лежать в основі підвищення продуктивності, якості ПЗ та змоги реагувати на зміни вимог ринку. Тому роль DevOps важлива у сучасному розвитку ПЗ та може бути корисною для різних галузей та розмірів проектів.

DevOps представляє собою практику розробки програмного забезпечення, в якій об'єднані команди розробки (Dev) та операцій (Ops, тобто інформаційно-технічного обслуговування або експлуатаційників). Мета такої інтеграції підвищення продуктивності, якості ПП та співпраці між командами. Основними принципами DevOps вважають:

- культуру співпраці: команди активно спілкуються, проводиться обмін знаннями та досвідом, а також відбувається спільне вирішення проблем;
- автоматизація: всі процеси (розробка, тестування, розгортання, моніторинг ПЗ) автоматизовані, що допомагає скороченню часу введення змін, зменшення кількості помилок та підвищення ефективності робочих процесів;
- інтеграція та розгортання: Continuous Integration (практика постійної інтеграції) та Continuous Deployment (розгортання) виступають центральними елементами DevOps, тобто всі зміни коду постійно інтегруються в основну кодову базу та розгортаються на виробництво без затримок;
- моніторинг та зворотній зв'язок: моніторинг ПЗ дозволяє виявляти проблеми на ранніх стадіях та відстежувати успішність розгортання, зворотній зв'язок від команд операцій та користувачів допомагає командам розробників вносити швидкі та ефективні зміни;
- постійне вдосконалення сприяє виявленню слабких місць, внесення покращень та оптимізація процесів, заохочує команди регулярно аналізувати та оцінювати свої процеси, практики та інструменти;
- масштабованість та еластичність при розробці ПЗ, дозволяють легко адаптуватися до змінних вимог та проводити масштабування відповідно до потреб. Тобто архітектура ПЗ та інфраструктура мають бути гнучкими та спроможними адаптуватися до зростання чи скорочення обсягу роботи або ресурсів.

Виходячи з вищеописаного ідея DevOps полягає в створенні міцної, гнучкої технології та культурі співпраці, яка буде стимулом для команд до відповідальності, самостійності та навчання. Представлені принципи допоможуть підвищувати ефективність, якість ПЗ та швидко реагувати на зміни вимог ринку.

1.2. Етапи життєвого циклу програмного забезпечення

Життєвий цикл ПЗ (Software Development Life Cycle, SDLC) представляє собою структурований процес, який охоплює різні етапи, необхідні для створення, випуску та підтримки ПЗ. SDLC включає такі етапи: визначення вимог; проектування; розробка; тестування; розгортання; підтримка та супровід.

У таких традиційних методологіях розробки ПЗ, як водоспадна (Waterfall) виконання етапів ЖЦ відзначається послідовністю. Сучасні технології, такі як Agile та DevOps, сконцентровані на ітеративному та гнучкому підході до розробки, тобто етапи SDLC виконуються одночасно, що призводить до більшої адаптивності, швидкості внесення змін та підвищення ефективності команд розробки та операцій (експлуатаційників).

При виконанні етапів ЖЦ ПЗ в синхронному режимі може відбуватися підвищення інтеграції між командами, скорочення часу виходу продукту на ринок та зменшення ризиків, пов'язаних з невдачами або затримками у процесі розробки. Нові можливості для підвищення продуктивності, ефективності та якості розробленого ПЗ можна виявити при дослідженні та розробці нових методів синхронного виконання SDLC.

1.3. Підходи до синхронного виконання етапів життєвого циклу

Все більшої актуальності у сучасному світі набуває синхронне виконання етапів ЖЦ ПЗ, через фактори, які є важливими для успіху, а саме: швидкість, гнучкість та ефективність. Для синхронного виконання ЖЦ є різні підходи, такі як Continuous Integration (CI), Continuous Deployment (CD) та Continuous Delivery.

Перший передбачає постійне злиття роботи команди розробників з репозиторієм, для більшого підвищення швидкості, гладкої інтеграції коду та зменшення ризиків конфліктів і затримок у процесі розробки. Другий сприяє автоматичному розгортанні змін коду на продуктивному середовищі після успішного проходження тестів та перевірок якості, що дасть змогу командам швидко реагувати на зміни вимог та забезпечувати безперервність ПЗ. Останній виступає поєднанням перших двох, спрямований на підтримку безперервного циклу випуску ПЗ, що забезпечуватиме готовність змін коду до розгортання в продуктивному середовищі.

Саме команди, які складаються з фахівців різних напрямів (розробка, тестування, операції тощо – крос-функціональні) дозволяють ефективно розробляти, тестувати та розгортати ПЗ одночасно, співпрацюючи на протязі усього процесу. Розбиття ПЗ на менші, незалежні компоненти, які дають змогу командам розробляти, розгортати та масштабувати окремі частини системи незалежно одна від одної лежать в основі мікросервісної архітектури. Даний принцип допомагає синхронному виконанню етапів ЖЦ, так як

паралельна робота над різними частинами системи забезпечується роботою окремих команд. Завдяки таким підходам компанії мають змогу реагувати на зміни вимог ринку, забезпечувати ефективність, гнучкість, швидкість розробки та розгортання ПЗ.

Гнучкий підхід до розробки ПЗ, який зосереджується на створенні окремих функцій чи можливостей носить назву Feature-driven development, або FDD. Завдяки йому команди можуть працювати паралельно над різними функціями та інтегрувати їх у загальний продукт, сприяючи синхронному виконанню етапів ЖЦ. Методика Канбан допомагає командам контролювати робочий процес, співпрацювати та підтримувати синхронність між етапами розробки ПЗ, в наслідок використання візуальних елементів для відображення робочого процесу та стану завдань. Використовуючи Continuous Testing команди виявляють та усувають дефекти на ранніх стадіях, сприяючи синхронності між етапами, шляхом автоматичного тестування коду на різних етапах ЖЦ. Команди операцій та розробки можуть працювати синхронно і забезпечують надійність та стабільність інфраструктури завдяки автоматизації розгортання та керування інфраструктурою ПЗ за допомогою коду та конфігураційних файлів, що є основою підходу Infrastructure as Code, або IaC.

Синхронність між етапами ЖЦ ПЗ можлива в наслідок співпраці та спільного володіння кодом, тобто всі члени команди мають доступ до коду та володіють навичкам його редагування. Це дозволить командам швидко вносити зміни, відповідати на вимоги та прискорювати процес розробки.

Вище представлені підходи надають можливість командам розробки та операцій (експлуатаційникам) працювати синхронно, співпрацюючи на всіх етапах ЖЦ ПЗ. Це сприятиме покращенню продуктивності, стабільності та якості продуктів, а також забезпеченню більшої гнучкості для команд у відповідь на зміни ринкових вимог. В світі DevOps застосування таких підходів стає все біль затребуваним, оскільки сприяє успіху команд та їх продуктів.

2. Розробка методу синхронного виконання етапів життєвого циклу в рамках DevOps

2.1. Опис процесів та інструментів для синхронного виконання етапів

Процеси та інструменти, які можуть сприяти ефективному виконанню робіт на кожному етапі, мають бути враховані при розробки методу синхронного виконання етапів ЖЦ в рамках DevOps. Одним з них виступає спільне планування, при якому використовуються так інструменти як Jira, Trello або Microsoft Teams. Ці системи та застосунок дають змогу командам розробки та операцій координувати свої дії, встановлюючи пріоритети та обмеження часу для кожного етапу.

Наступним підходом можна назвати Continuous Integration (CI), де використовують інструменти подібні до Jenkins, Travis CI або CircleCI. Вони сприятимуть автоматизації збірки, тестування та інтеграції коду, що дозволяє командам оперативно виявляти та усувати проблеми. Інструменти та підходи Continuous Deployment (CD), а саме: Spinnaker, Octopus Deploy або GitLab CI/CD, стають в пригоді для автоматизації процесу розгортання, забезпечення швидкого внесення змін до продуктів та мінімізації ризиків від помилок. Сприянню підтримці ефективної комунікації між членами команди та забезпеченню можливості обміну думками та ідеями в режимі реального часу, можливо при використанні інструментів комунікації, таких як Slack, Microsoft Teams або Zoom. Відстежувати стан системи та вчасно реагувати на можливі проблеми, командам допомагають інструменти моніторингу, наприклад Prometheus, Grafana або Datadog.

Поліпшення співпраці між командами розробки та ІТО і забезпечення більш ефективного та безперебійного робочого процесу сприяє інтеграція вище описаних інструментів та процесів в рамках методу синхронного виконання етапів ЖЦ. Також можна відзначити ще наступні аспекти, які будуть в нагоді при синхронізації роботи:

- розподіл роботи між невеликими крос-функціональними командами, які складаються з розробників, тестувальників та спеціалістів з інформаційно-технічного обслуговування, може сприяти швидкому вирішенню проблем та підвищенню ефективності роботи, тобто організація роботи в малих командах;
- використання контейнерних технологій, таких як Docker та Kubernetes, дозволяє створювати, розгортати та масштабувати ПЗ швидко та ефективно, а також гарантує однорідність середовищ на різних етапах розробки ПП;
- застосування автоматизованих тестів на різних рівнях (одиничні, інтеграційні, системні) сприятиме виявленню та виправленню помилок на ранніх етапах розробки, що прискорює процес впровадження змін та підвищує якість ПП;

Усі вище названі підходи у рамках синхронного виконання етапів ЖЦ дозволять командам розробки та ІТО оптимізувати свою роботу, підвищити продуктивність спільної роботи та досягти кращих результатів у розробці ПЗ.

2.2. Інтеграція команд розробки та операцій для спільної роботи

Одним з ключових аспектів методу синхронного виконання етапів ЖЦ в рамках DevOps є забезпечення ефективної співпраці між командами розробки та експлуатаційників. Тут можна відмітити наступні важливі аспекти.

Забезпечення доступу обох команд до спільних планів, цілей та інформації про стан проекту є важливим для розвитку взаєморозуміння та координації зусиль, що лежить в основі спільного планування та прозорості. В цьому процесі в пригоді стануть спільні інструменти Jira або Trello. Обговорення проблем, досягнень та планів на майбутнє відбуватиметься на регулярних зустрічах між командами, наприклад, це

можуть бути щоденні стендапи або спринт-планування. Це сприяє кращому розумінню взаємних відповідальностей та сприяє швидкому вирішенню проблем. Зменшення ризику помилок, забезпечення більш стабільних релізів та звільнення часу команд для зосередження на інших важливих завданнях, дозволяє впровадження автоматизованих процесів для розгортання, тестування та моніторинг ПЗ. Це стає можливим завдяки інструментам подібним до Jenkins, GitLab CI/CD або Kubernetes, які допомагають автоматизувати ці процеси. Заохочення команд розробки та операцій до постійного навчання та розвитку нових навичок сприяє підвищенню ефективності співпраці. Це можливе за посередництвом регулярних навчальних сесій, воркшопів та конференцій, які допомагатимуть командам ознайомитися з новими підходами, технологіями та практиками. Набуті знання та навички можуть бути використані для покращення процесів розробки та ІТО.

Одним з основних аспектів успішної інтеграції команд розробки та ІТО виступає розвиток спільної культури, яка підтримує співпрацю, взаємодопомогу та відкритість. Тут практикується спільні відповідальності, розподіл ролей та взаємна підтримка під час вирішення проблем. Збір даних про результати роботи команд та аналіз цих даних дозволяє виявити слабкі місця, зрозуміти, які процеси працюють ефективно, та зробити висновки про можливість покращення, тобто відбуватиметься постійне вимірювання та аналіз результатів. Використання ключових показників ефективності (KPI) та інших метрик для вимірювання успіху співпраці може допомогти командам зосередитися на досягненні спільних цілей.

Всі перераховані аспекти, які реалізовані в процесі синхронного виконання етапів ЖЦ в рамках DevOps, дозволяють підвищити рівень інтеграції та ефективності команд розробки та експлуатаційників і як наслідок сприятимуть швидкому розвитку та впровадженню ПЗ.

2.3. Використання автоматизації для підтримки синхронного виконання етапів

Основним компонентом методу синхронного виконання етапів ЖЦ в рамках DevOps виступає автоматизація. Усунення людських помилок, зменшення часу виконання рутинних завдань та забезпечення більш стабільного та прогнозованого процесу розробки та впровадження ПЗ можливе завдяки саме автоматизації. При автоматизації для підтримки синхронного виконання етапів визначають наступні напрямки:

- спроможність командам розробників та експлуатаційників ефективно координувати свою роботу, забезпечуючи швидке виявлення та виправлення проблем. Це можливе завдяки використанню Continuous Integration (CI), яка передбачає автоматичне злиття коду, написаного різними розробниками, та його перевірку на предмет помилок та працездатності;
- забезпеченню швидкого впровадження нових функцій та виправленню помилок, а також зменшенню часу між розробкою та випуском ПП допомагає застосування автоматизації розгортання Continuous Deployment (CD). Підхід, який включає в себе автоматичне розгортання випробуваного та перевіреного коду на продуктивне середовище;
- автоматизовані тести дозволяють швидко перевірити код на наявність помилок, відповідність вимогам та стабільність роботи. Використання автоматизованих тестів сприяє синхронному виконанню ЖЦ, оскільки розробники та спеціалісти з операцій мають можливість оперативно перевірити результати своєї роботи;
- використання автоматизації моніторингу та логування, що є важливими інструментами для відстеження стану системи та виявлення можливих проблем. При автоматизації цих процесів відбувається ефективне реагування команд розробки та операцій на зміни в роботі системи та своєчасне розв’язання виникаючих проблем;
- застосування автоматизації інфраструктури, тобто інфраструктура, як код (Infrastructure as Code, IaC) передбачає використання коду для автоматизації створення, конфігурації та управління інфраструктурою. Команди розробників та операцій використовуючи IaC можуть прискорити процеси налаштування інфраструктури, забезпечити більшу стабільність та контроль, а також спрощувати синхронне виконання етапів ЖЦ ПЗ;
- при автоматизації процесів спілкування між командами розробки та ІТО, полегшуючи координацію роботи та сприяючи синхронному виконанню етапів, застосовуються сучасні інструменти спілкування та координації, такі як системи спільної роботи, чати, системи керування завданнями та інші.

Якщо при вищезазначених напрямках буде застосовуватися автоматизація, то це сприятиме підвищенню продуктивності команд розробки та експлуатаційників, покращенню якості ПЗ та забезпеченню більшої стабільності системи. І як наслідок, синхронне виконання етапів ЖЦ ПЗ в рамках DevOps стає більш досяжним та практичним для команд.

3. Оцінка впливу методу на ефективність DevOps-процесів

3.1. Метрики для вимірювання ефективності синхронного виконання етапів

Існують метрики, які дозволять точно та об'єктивно виміряти вплив та результати впровадження методу синхронного виконання етапів на ефективність DevOps-процесів. Розглянемо деякі з них.

Метрика, яка вимірює час, потрібний для проходження ПЗ через всі етапи ЖЦ від початку розробки до розгортання на продуктивному середовищі носить назву час від початку розробки до випуску (Cycle Time). Зменшення часу свідчить про підвищення ефективності процесів розробки та операцій.

За допомогою метрики частоти випусків (Release Frequency) можна виміряти кількість випущених версій ПЗ протягом певного періоду часу. Збільшення частоти випусків свідчить про здатність команди швидко реагувати на зміни вимог та впроваджувати нові функції та виправлення.

Метрика часу відновлення після збою (Mean Time to Recovery, MTTR) дає змогу виміряти середній час, необхідний для відновлення системи після збою. Зменшення часу відновлення після збою свідчить про покращення ефективності процесів операцій та здатність команди швидко вирішувати проблеми.

Відсоток розгортань, які успішно довели ПЗ до продуктивного середовища без помилок можна виміряти завдяки метриці відсотку успішних розгортань (Deployment Success Rate). Якщо буде збільшуватися відсоток успішних розгортань, то це свідчитиме про покращення якості ПЗ та зменшення кількості проблем, пов'язаних з розгортанням.

Метрика середнього часу на внесення змін (Mean Time to Change, MTTC) призначена для вимірювання середнього часу, який необхідний для внесення змін у ПЗ, таких як додавання нових функцій або виправлення помилок. Зменшення середнього часу на внесення змін говорить про підвищення продуктивності команди розробки та більш гнучкий процес розробки.

Виміряти рівень задоволення користувачів ПЗ можливо завдяки метриці – задоволеність користувачів (User Satisfaction). Проводяться такі заміри шляхом опитувань, аналізу відгуків та рейтингів. Збільшення рівня задоволеності користувачів свідчить про покращення якості ПЗ та здатність команди відповідати на потреби користувачів.

Все ж таки вимірювання вище вказаних метрик є лише одним з аспектів оцінки ефективності синхронного виконання етапів ЖЦ в рамках DevOps. Крім цього, важливо проводити якісний аналіз процесів та співпраці між командами розробки та експлуатаційників, а також забезпечувати постійне вдосконалення цих процесів на основі зібраних даних

3.2. Аналіз впливу методу на швидкість вирішення проблем та задоволення від результатів проектів

На основі даних, які отримані з метрик, зазначених у підрозділі 3.1. можна оцінити вплив методу синхронного виконання етапів ЖЦ в рамках DevOps на швидкість вирішення проблем та задоволення від результатів проектів.

Застосування методу синхронного виконання етапів ЖЦ в рамках DevOps сприяє пришвидшеному виявленню та вирішенню проблем. Все тому, що розробники та операційні команди працюють разом, та можуть легко спілкуватися та вирішувати проблеми. Метрики, такі як MTTR, можуть допомогти відстежувати прогрес у цьому напрямку.

При застосування методу синхронного виконання етапів ЖЦ в рамках DevOps може покращитися якість розробленого ПЗ, а також забезпечується більша стабільність та надійність ПП. В результаті, користувачі можуть отримувати більше задоволення від використання розробленого програмного забезпечення, що відображається в метриці User Satisfaction.

Підвищенню задоволення команд розробників та ІТО сприятиме використання методу синхронного виконання етапів ЖЦ в рамках DevOps, оскільки вони можуть співпрацювати продуктивніше та швидше вирішувати проблеми. При проведенні опитувань серед працівників, отримують показники задоволеності команд, а також відстежується рівень затримки та текучості кадрів.

Застосування методу синхронного виконання етапів ЖЦ в рамках DevOps може призвести до підвищення ефективності робочих процесів та зменшення витрат на розробку та підтримку ПЗ. Оцінка віддачі від інвестицій (ROI) може допомогти оцінити економічний ефект від впровадження методу.

Важливим є також оцінка впливу методу синхронного виконання етапів ЖЦ в рамках DevOps на ключові показники ефективності (KPIs). Застосування методу може привести до поліпшення KPIs, таких як швидкість доставки, швидкість розгортання, надійність системи та інші. Це, в свою чергу, може позитивно вплинути на загальну ефективність організації.

Виявляти слабкі місця в процесі синхронного виконання етапів ЖЦ в рамках DevOps та пропонувати рекомендації щодо поліпшення цього методу, можливе завдяки проведенню такого аналізу. Як результат, організації зможуть краще використовувати свій потенціал та підвищувати ефективність своїх DevOps-процесів.

3.3. Порівняння з традиційними підходами до виконання етапів життєвого циклу програмного забезпечення

При проведенні порівняння впливу методу синхронного виконання етапів ЖЦ в рамках DevOps на ефективність процесів розробки ПЗ з традиційними підходами до виконання етапів, буде отримана об’єктивна оцінка, на основі якої можна робити потрібні висновки. Традиційні підходи, такі як каскадна модель або V-модель, характеризуються послідовним виконанням етапів, що може призвести до певних проблем та обмежень.

Традиційні підходи до розробки ПЗ можуть бути повільнішими, оскільки кожен етап виконується послідовно. У порівнянні з цим, метод синхронного виконання етапів ЖЦ пропонує співпрацю та інтеграцію між командами розробки та експлуатаційниками, що може призвести до прискорення розробки та доставки ПЗ. Також при традиційних підходах до розробки ПЗ можуть створюватися бар’єри між командами, оскільки вони працюють на різних етапах проекту. У результаті можуть виникати проблеми з комунікацією

та координацією між командами. Застосування методу синхронного виконання етапів сприяє покращенню комунікації та співпраці між командами розробки та операцій.

При використанні традиційних підходів до розробки ПЗ, коли справа доходить до внесення змін у проект або адаптації до змінних вимог клієнтів спостерігається менша гнучкість. Це може призвести до затримок у розробці та високих витрат на внесення змін після завершення етапів. У порівнянні з цим, метод синхронного виконання етапів ЖЦ дозволяє командам швидше реагувати на зміни та адаптуватися до нових вимог, що може підвищити продуктивність та задоволення від результатів проектів.

В традиційних підходах при розробці ПЗ виявлення проблеми або ризиків прийнято відкладати на пізніші етапи ЖЦ, що ускладнює їх вирішення та збільшує витрати на проект. Однак при використанні методу синхронного виконання етапів ЖЦ забезпечують раннє виявлення та вирішення проблем, що дозволяє краще управляти ризиками та зменшувати витрати на проект.

Традиційні підходи до розробки можуть призвести до компромісів між швидкістю розробки та якістю ПП. Оскільки метод синхронного виконання етапів ЖЦ забезпечує більшу співпрацю між командами та акцентує увагу на автоматизації, він може підвищити якість ПЗ та забезпечити стабільність та надійність продукту.

Враховуючи все вищеназване можна зробити висновок, що метод синхронного виконання етапів ЖЦ в рамках DevOps має переваги над традиційними підходами до розробки ПЗ. Він сприяє підвищенню ефективності процесів розробки та інформаційно-технічного обслуговування, поліпшенню комунікації між командами, забезпечує більшу гнучкість у внесенні змін та адаптації до нових вимог клієнтів, кращому управлінню ризиками та підвищенню якості ПЗ.

Треба також зазначити, що застосування методу синхронного виконання етапів ЖЦ залежить від рівня співпраці та взаємодії між командами розробки та операцій, а також від наявності компетентних фахівців, які можуть застосовувати цей підхід ефективно. Щоб досягти оптимальних результатів, організації повинні враховувати свої внутрішні ресурси, культуру та структуру, а також інвестувати у навчання та розвиток своїх співробітників.

Зважаючи на це, метод синхронного виконання етапів ЖЦ в рамках DevOps може стати значущим внеском у розробку ПЗ, покращуючи ефективність процесів та задоволення від результатів проектів. Однак для досягнення оптимальних результатів організаціям слід ретельно планувати та впроваджувати цей підхід, забезпечуючи підтримку на всіх рівнях управління та активного залучення усіх зацікавлених сторін.

4. Практичне застосування методу: вивчення випадків та прикладів

4.1. Аналіз успішних проектів, які використовують синхронний підхід в рамках DevOps

Проведемо аналіз декілька успішних проектів, де використовувався синхронний підхід в рамках DevOps, щоб продемонструвати переваги цього методу та його практичність.

Проєкт А. Велика технологічна компанія «CloudTech Innovations», яка працює над розробкою хмарних сервісів. Проект А став успішним завдяки впровадженню методу синхронного виконання етапів ЖЦ, який допоміг скоротити час внесення змін та випуску нових продуктів. Команди розробки та ІТО працювали разом, щоб використовувати автоматизацію для забезпечення безперебійного виконання процесів. Такий підхід сприяв підвищенню якості продуктів та задоволення клієнтів.

Щоб обґрунтувати висновки, необхідно розглянути деякі метрики, на основі яких продемонстровані зміни в результативності проекту після впровадження синхронного підходу:

- час випуску нових продуктів зменшився на 40%. Це означає, що компанія могла швидше відповідати на потреби ринку та конкурувати з іншими компаніями;
- зменшення кількості відкатів змін (rollbacks) зменшилася на 60%, що свідчить про поліпшення якості розробленого програмного забезпечення та зменшення кількості помилок;
- команди стали працювати продуктивніше та підвищилася задоволеність клієнтів через зниження середнього часу вирішення проблем на 50%;
- загальний обсяг виконаної роботи на одного співробітника збільшився на 30% через кращу координацію між командами та автоматизацію рутинних процесів;

На основі даних показників метрик підтверджується успішність впровадження методу синхронного виконання етапів ЖЦ в рамках DevOps у проекті CloudTech Innovations. Компанія змогла поліпшити продуктивність, якість продуктів та задоволення клієнтів, що в свою чергу сприяло зростанню її репутації та доходів.

- швидке внесення змін, адаптація до нових вимог ринку та висока якість послуг, стала наслідком збільшення показника задоволення клієнтів на 35%. Це свідчить про покращення комунікації між командами розробки та ІТО, що дозволяє швидко відповідати на зміни у вимогах та потребах клієнтів;
- через використання автоматизації рутинних процесів та підвищенню продуктивності співробітників відбулося зменшення витрат на управління проектами на 25%, тому компанія змогла сконцентруватися на стратегічних ініціативах та пришвидшити інноваційні процеси.

Приклад проекту CloudTech Innovations та наведені метрики дають змогу зробити висновок, що впровадження методу синхронного виконання етапів ЖЦ в рамках DevOps сприяло підвищенню ефективності проектів та покращенню результатів компанії. Це підтверджує, що такий підхід може бути

корисним для інших організацій, які прагнуть оптимізувати свої DevOps-процеси та досягти кращих результатів у розробці ПЗ.

Проєкт B. Стартап у сфері інтернету речей (IoT). В проекті В використання синхронного підходу в рамках DevOps допомогло забезпечити швидку відповідь на зміну вимог ринку та запровадження нових технологій. Команди розробки та ІТО були взаємопов'язані, що дозволило ефективно реагувати на проблеми та виклики, а також забезпечувати високу якість ПП. Інтеграція команд також сприяла поширенню знань та досвіду між співробітниками, що позитивно позначилося на результативності роботи.

Проєкт C. Міжнародна фінансова організація, яка реалізує складний проект розробки ПЗ. В проекті С синхронний підхід в рамках DevOps був успішно впроваджений для виконання різних етапів ЖЦ ПЗ. Це допомогло підвищити ефективність комунікації між командами розробки та експлуатаційників, а також сприяло реалізації складних задач за короткий час. Автоматизація процесів забезпечила високу продуктивність та мінімізувала ризики, пов'язані з ручним виконанням задач. Це призвело до високої якості продуктів та задоволення від результатів проекту.

Проєкт D. Організація у сфері охорони здоров'я, яка розробляє нові програмні рішення для медичних закладів. В рамках проекту D синхронний підхід до виконання етапів ЖЦ ПЗ дозволив забезпечити відмінну взаємодію між командами розробки та ІТО, що сприяло підвищенню ефективності проекту. Автоматизація процесів, впроваджена у DevOps, забезпечила безперебійність та надійність роботи ПЗ. В результаті, компанія змогла пропонувати клієнтам високоякісні продукти, що відповідають їх потребам.

Вищенаведені випадки свідчать про те, що використання методу синхронного виконання етапів ЖЦ ПЗ в рамках DevOps дозволяє підвищити ефективність проєктів, забезпечити швидку відповідь на зміни вимог ринку та забезпечити високу якість кінцевих продуктів. Застосування даного методу в різних галузях підтверджує його універсальність та практичність, а також вказує на значні переваги, які можуть бути досягнуті в результаті його впровадження в рамках DevOps.

4.2. Висновки з представлених випадків та їх вплив на розробку методу

Спираючись на отримані за допомогою аналізу успішних проєктів, які використовують синхронний підхід в рамках DevOps відомості, було зроблено висновки та визначено, яким чином такий підхід вплинув на розробку методу синхронного виконання етапів ЖЦ ПЗ.

Отже, успішні проекти показали, що спільна робота команд розробки та експлуатаційників є критично важливою для досягнення синхронного виконання етапів і це дало змогу командам швидко реагувати на зміни та адаптуватися до нових вимог ринку. Автоматизація, яка була впроваджена у розробку метода, забезпечила зменшення витрат на рутинні процеси, підвищення ефективності та покращення якості ПП, а це дозволило командам зосередитися на стратегічних аспектах розробки та пришвидшити інноваційні процеси.

При вивченні успішних проектів були застосовані метрики для вимірювання ефективності. Це дало можливість визначення ключових метрик, які можуть бути використані для оцінки продуктивності синхронного виконання етапів, а саме: зменшення часу внесення змін, підвищення швидкості вирішення проблем та покращення задоволення клієнтів.

При аналізі успішних проектів доцільним було порівняння запропонованого методу з традиційними підходами, які використовують про розробці ПЗ. Отримані результати підтвердили, що впровадження методу синхронного виконання етапів ЖЦ може призвести до кращих результатів, оскільки, в цьому випадку команди мають змогу швидше реагувати на зміни, і також забезпечується покращення якості продуктів та підвищується задоволення користувачів.

Також розроблений метод може бути успішно застосований у різних галузях та різноманітних проєктах. Він дозволяє забезпечити високу стабільність та здатність до масштабування продуктів, що, в свою чергу, допомагає підприємствам досягати успіху на ринку. При розробці методу синхронного виконання етапів відбувається формування культури співпраці та взаємодії між командами розробки та ІТО, при цьому підтримується висока мотивація працівників та відбувається стимулювання їх навчання та розвитку.

На підставі отриманого аналізу успішних вищеназваних проєктів можна зробити висновок, що метод виконання етапів ЖЦ в рамках DevOps допомагає покращити ефективність роботи команд, забезпечити високу якість продуктів та задоволення клієнтів. При цьому підприємства мають змогу адаптуватися до змін ринку та досягати успіху в конкурентному середовищі. Тому після врахування цих висновків, можна рекомендувати розроблені методи синхронного виконання етапів для впровадження в різних організаціях, що прагнуть оптимізувати свої DevOps-процеси та покращити загальну ефективність роботи. Також можна стверджувати, що розроблений метод синхронного виконання етапів ЖЦ в рамках DevOps є значущим інструментом для покращення процесів розробки та впровадження ПЗ. Запропонований метод дозволяє досягти більш високої продуктивності команд, кращої якості продуктів та вищої задоволеності клієнтів, що сприяє конкурентним перевагам та загальному успіху організацій на ринку.

4.3. Пропозиції щодо подальшої адаптації методу для різних галузей та розмірів проектів

Вище отримані результати та висновки стали основою для декількох рекомендацій, щодо адаптації методу синхронного виконання етапів ЖЦ в рамках DevOps для різних галузей та розмірів проектів.

Важливою виступає розробка гнучкого фреймворку, який легко буде налаштовуватися для відповіді на специфічні вимоги кожного проєкту і це дасть змогу успішної адаптації методу до різних галузей та розмірів проєктів. Рекомендується інтегрувати метод з існуючими практиками розробки та інформаційно-технічного обслуговування, а також з вже використовуваними інструментами для забезпечення плавного переходу до синхронного виконання етапів.

Для успішного впровадження та використання методу у своїй роботі, забезпечити належний рівень навчання та підтримки для команд розробки та експлуатаційників. Допоможе уникнути потенційних перешкод та сприяти успішному впровадженню методу використання розуміння та врахування особливостей організаційної культури. Проведення регулярного моніторингу та оцінка результатів впровадження запропонованого методу дадуть змогу виявити можливі проблеми, а також надати важливу інформацію для подальшого вдосконалення методу. Необхідність розгляду можливості для масштабування та застосування методу на більших проєктах або на рівні всієї організації, доцільна після успішного впровадження методу в проєкті. Це може включати підготовку додаткових команд, розробку загальних стандартів та практик, а також створення спільних ресурсів для сприяння широкому впровадженню методу.

Метод синхронного виконання етапів може мати різну ефективність в залежності від специфіки галузі та розміру проєкту. Тому виникає потреба в проведенні досліджень та аналізу результатів впровадження методу в різних умовах, щоб виявити оптимальні сценарії його застосування. Співпраця з академічними та дослідницькими установами, які спеціалізуються в галузі DevOps та суміжних областях, у подальшому може виступати підтримкою розвитку запропонованого підходу. Це ж стосується розробки кастомізованих інструментів, тобто у разі потреби можна розглядати розробку власних інструментів та рішень, які будуть відповідати специфічним вимогам проєкту або галузі. Використання таких засобів націлене забезпечувати кращу інтеграцію методу в рамках існуючих процесів та підвищити його ефективність.

Корисною для підвищення загальної ефективності процесів розробки та інформаційно-технічного обслуговування може виступати в подальшому вдосконалення та адаптація методу синхронного виконання етапів ЖЦ в рамках DevOps для різних галузей та розмірів проєктів. При цьому може імовірно відбудеться скорочення термінів реалізації проєктів, зниження витрат та підвищення якості ПЗ. Також рекомендується розгляд методу при його застосуванні на різних рівнях організаційної структури, це забезпечить успішну адаптацію для різних галузей та розмірів проєктів. До прикладу, метод може бути впроваджений на рівні відділів або підрозділів, що допоможе надати більш ефективну координацію та співпрацю між командами.

Виникає потреба в постійному вдосконалення методу синхронного виконання етапів ЖЦ, через безперервну зміну технологій та практик розробки ПЗ. Сюди можна віднести збір зворотного зв'язку від учасників процесу, аналіз результатів застосування методу, а також проведення досліджень та інновацій для виявлення нових можливостей для його покращення.

Підсумовуючи все вищеназване, можна сказати, що адаптація методу синхронного виконання етапів ЖЦ в рамках DevOps для різних галузей та розмірів проєктів вимагає: комплексного підходу, що включає аналіз потреб організації; інтеграцію з існуючими практиками та інструментами; підготовку та підтримку команд; врахування особливостей організаційної культури; постійне вдосконалення методу. При вдалому впровадженні та адаптації, метод синхронного виконання етапів ЖЦ може принести значні переваги для організацій різних галузей та розмірів проєктів, покращуючи ефективність розробки та ІТО, скорочуючи час внесення змін та підвищуючи якість ПЗ.

Висновки

У даній статті було представлено метод синхронного виконання етапів ЖЦ ПЗ в рамках DevOps. Зокрема, були проаналізовані теоретичні аспекти DevOps та синхронного виконання етапів, розроблено метод, який базується на інтеграції команд розробки та експлуатаційників, а також використанні автоматизації для підтримки синхронного виконання етапів. Були запропоновані метрики для оцінки впливу методу на ефективність DevOps-процесів. Вказані метрики використовувались для вимірювання ефективності синхронного виконання етапів. Також проведено аналіз впливу методу на швидкість вирішення проблем та задоволення від результатів проєктів.

На прикладі успішних проєктів, де використовувався синхронний підхід в рамках DevOps, проводилось дослідження застосування методу і в результаті дослідження було виявлено, що такий підхід може принести значні переваги для організацій. Надані основні пропозиції щодо подальшої адаптації методу для різних галузей та розмірів проєктів, включають ретельний аналіз вимог до проекту, вибір найбільш підходящих практик та інструментів, підготовку та підтримку команд, а також врахування особливостей організаційної культури та постійне вдосконалення методу.

Отже, в цілому підвищення ефективності команд розробки та експлуатаційників підтверджуються отриманими результатами дослідження, які підкреслюють потенційні переваги методу синхронного виконання етапів ЖЦ ПЗ. У майбутньому можна розширити дане дослідження, проводячи додаткові емпіричні експерименти на різних проєктах та організаціях, що дозволить отримати більш точні дані про продуктивність застосування методу. Також важливим є розробка спеціалізованих інструментів та технологій, які сприятимуть практичному впровадженню методу синхронного виконання етапів ЖЦ. Можна спробувати розглянути можливість впровадження методу в суміжних областях, таких як кібербезпека,

штучний інтелект, та Інтернет-речей, щоб виявити специфічні вимоги та виклики, пов'язані з синхронним виконанням етапів ЖЦ в цих галузях.

Проведення постійного аналізу та адаптації методу до динамічно змінюваних умов роботи команд розробки та ІТО буде сприяти підвищенню гнучкості та адаптивності організаційних процесів і, як наслідок, зростанню ефективності методу в контексті конкретних організацій. В подальшому можна розглянути перспективу інтеграції методу синхронного виконання етапів ЖЦ з іншими сучасними підходами у сфері ПЗ, такими як Agile, Lean та Continuous Delivery. Це допоможе компаніям отримати додаткові стратегічні переваги та підвищити конкурентоспроможність на ринку.

Література

1. Kim, G., Debois, P., Willis, J., & Humble, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press. p. 480.

2. Sutherland, J., & Schwaber, K. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Penguin Random House. p. 256.

3. Lwakatare, L. E., Kuvaja, P., & Oivo, M. "Dimensions of DevOps," in Proceedings of the Agile Processes in Software Engineering and Extreme Programming - XP, 2015, p. 212–217.

4. Fitzgerald, B., & Stol, K. J. (2017). Continuous Software Engineering: A Roadmap and Agenda. Journal of Systems and Software, 123, p.176-189.

5. Duvall, P. M., Matyas, S., & Glover, A. (2015). Continuous Integration: Improving Software Quality and Reducing Risk. Addison-Wesley Professional. p. 336.

6. Smart, J. (2016). Continuous Delivery Pipeline - Where Does It Choke? Release Management within the DevOps Context. Software Quality Professional, 18(4), p. 4-15.

7. Leppänen, M., Mäkinen, S., Pagels, M., Eloranta, V. P., Itkonen, J., Männistö, T., & Mäntylä, M. V. (2015). The Highways and Country Roads to Continuous Deployment. IEEE Software, 32(2), p. 64-72.

8. Dyck, A., Penners, R., & Lichter, H. (2015). Towards Definitions for Release Engineering and DevOps. Proceedings of the IEEE/ACM 37th IEEE International Conference on Software Engineering.

9. Forsgren, N., & Humble, J. (2018). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press. p. 288.

10. Olsson, H. H., & Bosch, J. (2012). Climbing the "Stairway to Heaven": A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. Proceedings of the 37th IEEE International Conference on Software Engineering. p. 392–399.

11. Laukkanen, E., Itkonen, J., & Lassenius, C. (2017). Problems, Causes and Solutions when Adopting Continuous Delivery: A Systematic Literature Review. Information and Software Technology, 82, p. 55-79.

12. Capraro, M., & Riehle, D. (2017). Inner Source Definition, Benefits, and Challenges. ACM Computing Surveys, 50(4), p. 1-34.

.