

Чернівецький національний університет імені Юрія Федьковича

(повне найменування закладу вищої освіти)

Навчально-науковий інститут фізико-технічних і комп'ютерних наук

(назва інституту/факультету)

Кафедра комп'ютерних систем та мереж

(назва кафедри)

СИЛАБУС

навчальної дисципліни

Основи DevOps / DevNet

(вказіть назву навчальної дисципліни (іноземною, якщо дисципліна викладається іноземною мовою))

вибіркова

(обов'язкова чи вибіркова)

Освітньо-професійна програма – “Комп'ютерна інженерія”, “Програмування мобільних і вбудованих комп'ютерних систем та засобів Інтернету речей”

Спеціальність 123 – Комп'ютерна інженерія

(шифр і назва спеціальності)

Галузь знань 12 – Інформаційні технології

(шифр і назва галузі знань)

Рівень вищої освіти – перший (бакалаврський)

(вказати: перший (бакалаврський)/другий (магістерський)/третій (освітньо-науковий))

Навчально-науковий інститут фізико-технічних і комп'ютерних наук

(назва факультету / інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання – українська

(мова, на якій читається дисципліна)

Кількість кредитів: 3,0

Форми навчальної діяльності: лекції, лабораторні роботи, самостійна робота

Форма підсумкового контролю: залік

Розробники: Лісовенко Ірина Дмитрівна, асистент кафедри КСМ

(вказати авторів (викладач (ів)), їхні посади, наукові ступені, вчені звання)

Профайл викладача (-ів) <https://csn.chnu.edu.ua/>,
<https://csn.chnu.edu.ua/employees/lisovenko-iryna-dmytrivna/>

Контактний тел. + (38) 0372 50 94 32 (кафедра КСМ) – Лісовенко І.Д.

E-mail: i.lisovenko@chnu.edu.ua

Сторінка курсу в Moodle <https://lms.netacad.com/course/view.php?id=1247415>

Консультації згідно з розкладом (1 раз у 2 тижні)

1. Анотація дисципліни

Курс «Основи DevOps / DevNet» призначений для формування компетентностей випускників спеціальності 123 - Комп'ютерна інженерія в галузі прикладного застосування основ операційної розробки (DevOps), вивченню процесів керування розробкою програмного забезпечення, операційними процесами та інфраструктурою сервісів і продуктів. Його введення в навчальний план дозволяє розширити цикл дисциплін з програмування для підготовки бакалаврів, а також надати їм додаткові знання і практичні навички в розгортанні середовищ засобами DevNet при виконанні випускних кваліфікаційних робіт, та в майбутній професійній діяльності.

Перевага даного курсу в тому, що він відповідає сертифікату професійної розробки Cisco DevNet Associate Certificate.

2. Мета навчальної дисципліни: формування необхідного рівня теоретичної і практичної підготовки студентів для грамотного використання ними знань основ операційної розробки при освоєнні суміжних дисциплін та у майбутній професійній діяльності, а також формування знань для розгортання середовищ розробки засобами DevNet. Формування у студентів навичок налагодження інфраструктури забезпечення професійної розробки спеціалізованих програмних продуктів, використання технологій автоматизації виробничих процесів пов'язаних з розробкою прикладних програм та їх розгортання.

Завдання: вивчення основних тенденцій розвитку технологій, формування знань про основи спеціалізованого програмування та комп'ютерних мереж, також надати студентам базових знань і навичок, які вони зможуть використовувати у практичній діяльності для реалізації безпечного інтегрування різних сервісів; формування у студентів спроможності подальшого самостійного засвоєння знань та вмій, розвиток пізнавального хисту, освоєння раціональних прийомів оволодіння знаннями.

3. Пререквізити. Для коректного розуміння і засвоєння матеріалу даного курсу слухачі повинні попередньо оволодіти знаннями з дисциплін «Програмування. Ч1. Основи алгоритмізації і програмування на Python і JavaScript», «Програмування. Ч3. Основи об'єктно-орієнтованого програмування», «Архітектура комп'ютерів» та «Комп'ютерні системи». Результати навчання за цим курсом потрібні при вивченні дисципліни другого рівня освіти «Технології IoT & IoE для Big Data Analysis», при виконанні випускних кваліфікаційних робіт та в майбутній професійній діяльності.

4. Результати навчання

У результаті вивчення навчальної дисципліни студент повинен

4.1. Знати: ключові поняття і принципи організації середовища розробки засобами ресурсів DevNet; основні тенденції розвитку інструментів DevNet; методологію програмування в Python та особливості роботи в ОС Linux і з платформою GitHub; технології проектування спеціалізованого програмного забезпечення та середовище розробки програмних продуктів і сервісів засобами ресурсів DevOps/DevNet.

4.2. Вміти: використовувати для практичних задач мову програмування Python та навички роботи з ОС Linux, платформою GitHub; розгортати середовище розробки засобами ресурсів DevNet; втілювати передовий досвід проектування і розроблення

програмного забезпечення; створювати запити REST API через HTTPS для безпечного поєднання сервісів; обґрунтувати вибір процесів і пристроїв, необхідних для забезпечення мережного з'єднання; використовувати сучасні технології для розгортання та захисту прикладних програм і даних у хмарному середовищі; порівнювати методи тестування і впровадження програмного забезпечення у середовищах автоматизації та моделювання; порівняти платформи, які використовуються для колективної роботи, керування інфраструктурою та її автоматизації.

4.3. Набути компетентностей:

ЗК – загальних

ЗК 1. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК 3. Здатність застосовувати знання у практичних ситуаціях.

ЗК 7. Вміння виявляти, ставити та вирішувати проблеми.

ЗК 8. Здатність працювати в команді.

ФК - фахових

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

ФК13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

ПРН - програмні результати навчання

ПРН2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН4. Знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.

ПРН14. Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення

ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

5. Опис навчальної дисципліни

5.1. Загальна інформація

Назва навчальної дисципліни <i>Основи DevOps / DevNet</i>												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	4	8	3,0	90	2	12			24	54		залік
Заочна	4	8	3,0	90	2	4			6	80		залік

Примітка. Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 0,67 ((12+24)/54);

для заочної форми навчання – 0,13 ((4+6)/80).

5.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						заочна форма						
	усьо го	у тому числі					усьо го	у тому числі					
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	
Змістовий модуль 1. Основні технології DevOps													
Тема 1. Вступ до курсу. Основи мови програмування Python і ОС Linux	15,0	2,0		4,0		9,0	14,7	0,7		1,0			13,0
Тема 2. Проектування і розроблення ПЗ	15,0	2,0		4,0		9,0	14,7	0,7		1,0			13,0
Тема 3. Розуміння та використання API	15,0	2,0		4,0		9,0	15,6	0,6		1,0			14,0
Разом за ЗМ1	45,0	6,0		12,0		27,0	45,0	2,0		3,0			40,0

Змістовий модуль 2. Застосування інструментів DevOps											
Тема 4. Розгортання та безпека застосунків	15,0	2,0		4,0		9,0	14,7	0,7		1,0	13,0
Тема 5. Інфраструктура та автоматизація	15,0	2,0		4,0		9,0	14,7	0,7		1,0	13,0
Тема 6. Платформи керування і автоматизації інфраструктури	15,0	2,0		4,0		9,0	15,6	0,6		1,0	14,0
Разом за ЗМ2	45,0	6,0		12,0		27,0	45,0	2,0		3,0	40,0
Усього годин	90,0	12,0		24,0		54,0	90,0	4,0		6,0	80,0

5.2. Перелік тем і розподіл максимально можливої кількості балів, які отримують студенти за виконання всіх видів навчальної діяльності

Змістовий модуль 1. Основні технології DevOps

T1. Вступ до курсу. Основи мови програмування Python і ОС. (Тест №1 – 2,5 бали, ЛР № 1 – 4 бали)

T2. Проектування і розроблення ПЗ. (Тест №2 – 2,5 бали, ЛР № 2 – 4 бали)

T3. Розуміння та використання API. (Тест №3 – 2,5 бали, ЛР № 3 – 4 бали)

M1. Модульна контрольна робота №1 – 7,5 балів.

Змістовий модуль 2. Застосування інструментів DevOps

T4. Розгортання та безпека застосунків. (Тест №4 – 2,5 бали, ЛР № 4 – 4 бали)

T5. Інфраструктура та автоматизація. (Тест №5 – 2,5 бали, ЛР № 5 – 4 бали)

T6. Платформи керування і автоматизації інфраструктури. (Тест №6 – 2,5 бали, ЛР № 6 – 4 бали)

M2. Модульна контрольна робота №2 – 7,5 балів.

5.3. Теми лабораторних занять

№	Назва теми	Кількість
		годин
1.	Встановлення віртуального лабораторного середовища	4
2.	Огляд програмування на Python	4
3.	Керування версіями програмного забезпечення за допомогою Git	4
4.	Дослідження REST API з API симулятором та Postman	4
5.	Створення зразка веб-застосунку в Docker-контейнері	4
6.	Створення CI/CD конвеєра з використанням Jenkins	4
		24

Примітка. Методичні рекомендації та завдання до лабораторних робіт доступні на інтернет-ресурсі:
<https://lms.netacad.com/course/view.php?id=1247415>

Програмне забезпечення для виконання лабораторних робіт: Virtual Machine Lab Environment з пакетами Webex Teams, Packet Tracer, DPI Scanning. Програмне забезпечення надається безоплатно академією Cisco.

5.4. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Установка ОС Linux. Підсистема довідки. Базові команди. Робота з файлами і каталогами. Текстовий редактор vim.	6,0
2	Управління користувачами і групами. РАМ-модулі. Типи файлів Unix. Жорсткі диски. Монтування розділів. RAID і LVM	6,0
3	Встановлення програмного забезпечення. Управління процесами. Утиліти ps, top, htop, atop . Журналювання. Резервне копіювання. Налаштування мережевих параметрів.	6,0
4	Мережні технології. Моделі OSI і TCP / IP. Технології локальних мереж. Адресація канального рівня. Введення у маршрутизацію. IP-адресація. Протоколи IP, ICMP. Технологія NAT. Протоколи транспортного рівня: UDP і TCP. Протоколи прикладного рівня: протокол DNS.	6,0
5	Мова Go 13 Коротка історія мови. Підготовка коду, компіляція та запуск Використання пакетів Оператори мови Go. Функції Реалізація простого HTTP-серверу.	6,0
6	Використання Jenkins, Ansible, Zabbix	6,0
7	Контейнеризація, віртуалізація та Docker	6,0
8	Інструменти моніторингу	6,0
9	Технологія Kubernetes. Історія створення та призначення Завантаження/перезавантаження контейнерів. Pod-оболонки Планувальник Kubernetes.	6,0
		56

6. Форми і методи навчання

Форми навчання – це проблемні й оглядові лекції, лабораторні заняття, заняття із застосуванням комп'ютерної та телекомунікаційної техніки, інтерактивні заняття з навчанням одних студентів іншими, інтегровані заняття, проблемні заняття, відеолекції, відеозаняття і відеоконференції засобами Google Meet, Zoom, Cisco Webex, заняття з використанням системи електронного навчання Netacad.

Методи: проблемний виклад матеріалу, частково-пошукові та дослідницькі лабораторні практикуми, презентації, кейс-стаді, консультації і дискусії, робота в інтернет-класі: електронні лекції, лабораторні роботи, дистанційні консультації та ін., спрямовані на активізацію і стимулювання навчально-пізнавальної діяльності студентів.

Підходи до навчання: використовуються студентоцентрований, проблемно-орієнтований, діяльнісний, комунікативний, професійно-орієнтований, міждисциплінарний підходи.

Для викладання матеріалів з навчальної дисципліни «Основи DevOps / DevNet» використовуються наступні методи навчання.

6.1. Словесні методи навчання. Навчальна лекція

За допомогою даного методу забезпечується усне викладення матеріалу великими ємністю й складністю логічних побудов, доказів і узагальнень. В ході лекції використовуються прийоми усного викладення інформації, підтримання уваги протягом тривалого часу, активізації мислення студентів, прийоми забезпечення логічного запам'ятовування, переконання, аргументації, доказів, класифікації, систематизації і узагальнення. В залежності від специфіки лекційного матеріалу іноді використовується лекція-діалог.

6.2. Індуктивний метод навчання

Даний метод навчання використовується в рамках лекційних занять, коли матеріал носить, здебільшого, фактичний характер. В рамках лабораторних занять метод застосовується при виконанні технічних задач, коли студенти використовують раніше здобуті теоретичні знання при роботі з конкретними пристроями (комп'ютерами) та програмними продуктами.

6.3. Репродуктивний метод навчання

Даний метод навчання використовується в рамках лекційних і лабораторних занять, а також під час самостійної роботи студентів. Метод передбачає роботу студентів за визначеним алгоритмом. Згідно з методом для виконання завдань студентам надаються методичні вказівки, правила і навчальні приклади.

6.4. Проблемно-пошукові методи навчання

Проблемно-пошукові методи застосовуються в ході проблемного навчання, а саме в процесі виконання лабораторних робіт та індивідуальних науково-дослідних завдань, де під проблемною ситуацією треба вважати невідповідність між тим, що вивчається і вже вивченим. При використанні проблемно-пошукових методів навчання викладач використовує такі прийоми: створює проблемну ситуацію (ставить питання, пропонує задачу, експериментальне завдання), організує колективне обговорення можливих підходів до рішення проблемної ситуації, стимулює висування гіпотез, тощо. Студенти роблять припущення про шляхи вирішення проблемної ситуації, узагальнюють раніше набуті знання, виявляють причини явищ, пояснюють їхнє походження, вибирають найбільш раціональний варіант вирішення проблемної ситуації. Викладач обов'язково керує цим процесом на всіх етапах, а також за допомогою запитань-підказок. Також даний метод використовується при опрацюванні матеріалів в системі дистанційної освіти «Netacad».

6.5. Наочний метод навчання

Наочний метод достатньо важливий для студентів, оскільки забезпечує візуальне подання навчального матеріалу, зокрема, з використанням інформаційно-комунікаційних технологій. При викладанні дисципліни наочний метод навчання поєднується зі словесними методами для представлення інформації у вигляді таблиць, рисунків, схем та діаграм.

7. Система контролю та оцінювання

Засобами оцінювання та демонстрування результатів навчання є

- контрольні роботи;
- стандартизовані тести;
- презентації результатів виконаних завдань та досліджень;
- завдання на лабораторному обладнанні.

Формами поточного контролю рівня знань є усна та письмова відповідь студента при захисті виконаних лабораторних робіт, кількість отриманих балів при виконанні тестового завдання, а також письмова відповідь при написанні модульних контрольних робіт. Формами підсумкового контролю рівня знань є усна та письмова відповідь студента при здачі заліку.

7.1. Критерії оцінювання результатів навчання з навчальної дисципліни

Критерієм успішного проходження здобувачем освіти підсумкового оцінювання є досягнення ним мінімальних порогових рівнів оцінок за кожним запланованим результатом навчання навчальної дисципліни.

У залежності від характеру відповіді студента кількість балів за кожний вид діяльності може бути визначена за наступними критеріями:

К-ть балів	Критерії оцінки
Мах	Студент дає вичерпну відповідь на поставлене запитання
$0,8 \cdot \text{Мах}$	Студент при відповіді на поставлене запитання припустився незначних неточностей, які не впливають на суть відповіді
$0,6 \cdot \text{Мах}$	Студент при відповіді на поставлене запитання припустився помилок, які виправляє за допомогою викладача; в середньому може дати правильні відповіді на 50% питань теми
$0,4 \cdot \text{Мах}$	Студент при відповіді на поставлене запитання припустився суттєвих помилок, які все ж таки виправляє за допомогою викладача; дає правильні відповіді на 30% питань теми
$0,2 \cdot \text{Мах}$	Студент за допомогою викладача фрагментарно відповідає на запитання, проте не в повній мірі володіє мінімальним рівнем знань з даного питання
0	Характер відповідей дає підставу стверджувати, що студент неправильно зрозумів суть питання чи не знав правильної відповіді, а тому відповідав, припускаючись грубих помилок.

Примітка: за Мах прийнято максимальну оцінку для даного виду діяльності; заокруглення проводиться до одиниць балу.

Шкала та критерії оцінювання: національна та ЄКТС (Європейська кредитна трансферно-накопичувальна система, ECTS)

Оцінка за шкалою ЄКТС	Пояснення	Оцінка за 100-бальною шкалою	Оцінка за національною шкалою
A	відмінно	90 – 100	відмінно
B	дуже добре	80-89	добре
C	добре	70-79	
D	задовільно	60-69	задовільно
E	достатньо	50-59	
FX	(незадовільно) з можливістю повторного складання	35-49	незадовільно
F	(незадовільно) з обов'язковим повторним курсом	1-34	

Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)								Підсумк. контроль (залік)	Сумарна кількість балів
Змістовий модуль 1				Змістовий модуль 2					
T1	T2	T3	M1	T4	T5	T6	M2		
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	40	100

7.2. Умови зарахування результатів неформальної освіти

Студент, згідно Положення ЧНУ «Про неформальну освіту» може отримати додаткові бали, або бути звільненим від окремих видів роботи з окремих тем, якщо у нього наявні сертифікати про неформальну освіту з проблем, які вивчаються на дисципліні «Основи DevOps / DevNet».

Також, як виконані види роботи з відповідних тем зараховуються студенту бали за наукові публікації у матеріалах науково-практичних конференцій та фахових чи апробаційних виданнях.

7.3. Політика курсу

Самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей).

Академічна доброчесність: посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації.

Відвідування: Відвідування занять є обов'язковим. Засвоєння пропущеної теми лекції з поважної причини перевіряється під час складання підсумкового контролю.

Пропуск лекції з неповажної причини відпрацьовується студентом (співбесіда, реферат тощо). Пропущені практичні та лабораторні заняття, незалежно від причини пропуску, студент відпрацьовує згідно з графіком консультацій.

8. Рекомендована література

Фахова (основна)

1. Лісовенко І.Д. Основи DevOps / DevNet: лабораторний практикум. І.Д. Лісовенко, Ю. В. Танасюк. – Чернівці : Чернівецький національний університет, 2022 – 85 с. (електронне видання).
2. Mikael Krief. Learning DevOps. – Packt Publishing, 2019. – 489 p.
3. Scott Chacon, Ben Straub. Pro Git. – Apress, 2014. – 441 p.
4. Моует Е. Використання Docker / пер. з англ. А.В. Снастіна; наук. ред. А.А.Маркелов. –К.: ДК Прес, 2017. – 354 с.
5. Арундел Джон. Kubernetes для DevOps: розгортання, запуск и масштабування в хмарі. – К.: Прес, 2020. – 384 с.
6. Пол Беррі. Python. Легкий для сприйняття довідник. Вид-во Фабула, 2021. - 624 с.
7. Newmarch, Jan. Network Programming with Go: Essential Skills for Using and Securing Networks. – Apress, 2017.
8. Pavel Yosifovich, Mark Russinovich, David Solomon, Alex Ionescu. Windows Internals, Part 1: System architecture, processes, threads, memory management, and more, 7th Edition – Microsoft Press, 2017. – 800 p.
9. Evi Nemeth. UNIX and Linux System Administration Handbook, 5th Edition / Evi Nemeth, Garth Snyder, Trent Hein, Ben Whaley, Dan Mackin. – Addison-Wesley Professional, 2017. – 1232 p.
10. Michael Hüttermann. DevOps for Developers. Apress, 2012. – 184 p.
11. Viktor Farcic. The DevOps 2.0 Toolkit. Viktor Farcic, 2016 – 404 p.
12. Len Bass, Ingo Weber, Liming Zhu. DevOps A Software Architect’s Perspective. Addison-Wesley, 2015. – 338p.
13. DevOps For Dummies. IBM Limited Edition, 2014 – 51 p.

Допоміжна

14. Burns B. Designing Distributed Systems. Sebastopol : O’Reilly Media, 2018. 149 p.
15. Terrell R. Concurrency in .NET. Shelter Island: Manning, 2018. 534 p.
16. Steen M., Tanenbaum A. Distributed-systems. New York : Pearson Education. – 2017. – 596 с.
17. Troelsen A., Japikse P. Pro C# 7: With .NET and .NET Core. New York : APress. Inc., 2017. 1372 с.
18. Kozyra N. Mastering Go Web Services. – Packt Publishing, 2015. – 264 p.
19. Greaves, Karen, and Samantha Laing. Collaboration Games from the Growing Agile Toolbox . Victoria, BC: Leanpub/Growing Agile, 2014.
20. Cowie, Jon. Customizing Chef . Sebastopol, CA: O’Reilly Media, 2014.
21. Dixon, Jason. Monitoring with Graphite . Sebastopol, CA.: O’Reilly Media, 2015.

22. Forsgren, Nicole, and Jez Humble. «The Role of Continuous Delivery in IT and Organizational Performance». In the Proceedings of the Western Decision Sciences Institute (WDSI), Las Vegas, Nevada, October 27, 2015.
23. Reed, J. Paul. DevOps in Practice . Sebastopol, CA: O'Reilly Media, 2013.

9. Інформаційні ресурси

1. DevNet Associate (DEVASC) 1.0 – Cisco Networking Academy. URL : <https://lms.netacad.com/course/view.php?id=1247415>. (дата звернення: 20.08.2022).
2. Allspaw, John. «Resilience-engineering: Part 1». KitchenSoap.com, September 21, 2012. URL : <https://www.kitchensoap.com/2011/04/07/resilience-engineering-part-i/> (дата звернення: 20.08.2022).
3. Caum, Carl. «Continuous Delivery vs. Continuous Deployment: What's the Diff?». Puppet blog, August 30, 2013. . URL : <http://bit.ly/cd-vs-cd> (дата звернення: 20.08.2022).
4. Coutinho, Rodrigo. «In Support of DevOps: Kanban vs. Scrum». DevOps.com, July 29, 2014. . URL : <http://bit.ly/kanban-v-scrum> (дата звернення: 20.08.2022).
5. Kim, Gene. Kanbans and DevOps: «Resource Guide for Phoenix Project (Part 2).» IT Revolution Press, N.d. URL : <http://bit.ly/kanbans-devops> (дата звернення: 20.08.2022).
6. Arrested DevOps. URL : <https://www.arresteddevops.com/> (дата звернення: 20.08.2022).