

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньо-наукової
програми

 І. Б. Туркін
(підпис) (ініціали та прізвище)
« 30 » 08 2021 р.

**СИЛАБУС ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Основи DevOps

(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

(цифр і найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»

(код та найменування спеціальності)

Освітньо-наукова програма: «Інженерія програмного забезпечення»

(найменування освітньої програми)

Рівень вищої освіти: *другий (магістерський)*

Силабус введено в дію з 01.09.2021 року

Харків – 2021 р.

Розробник: Кузнецова Ю. А., доцент каф. 603, к.т.н., доцент
(прізвище та ініціали, посада, науковий ступінь і вчене звання) 
(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри інженерії
програмного забезпечення (№ 603)
(назва кафедри)

Протокол № 2 від « 31 » серпня 2021 р.

Завідувач кафедри, д.т.н., професор
(науковий ступінь і вчене звання) 
(підпис) І. Б. Туркін
(ініціали та прізвище)

Погоджено з представником здобувачів освіти:

Представник студентського самоврядування 
(підпис) Д. В. Колодій
(ініціали та прізвище)

1. Загальна інформація про викладача



Кузнецова Юлія Анатоліївна доцент каф. 603, к.т.н., доцент.

Основні навчальні курси, що веде викладач: «Людино-машинна взаємодія» (24 години), «Інформаційні технології розроблення програмного забезпечення» (32 години), «Логічне та функціональне програмування» (32 години) щорічно, а також «Проектування інтерфейсу користувача» (24 години) з 1 вересня 2021 року та «Основи DevOps» (з 2-го семестру 2022 р.) Розробила методичне забезпечення лекційного матеріалу, лабораторних і практичних робіт.

Має **79** публікацій, з них **74** наукових та **11** навчально-методичного характеру, у тому числі **15** наукових праць у фахових наукових виданнях України та **7** публікацій, які включено до наукометричних баз **Scopus** і **WebOfScience**.

Напрями наукових досліджень: інтернет речей, доповнена реальність, високонавантажені системи, контейнеризація та безсерверні обчислення, UI/UX дизайн, інтерактивне мультимедійне навчальне тестування.

2. Опис навчальної дисципліни

Семестр, в якому викладається дисципліна – 2 семестр.

Обсяг дисципліни: 5 кредитів ЄКТС/150 годин, у тому числі аудиторних – 56 год, самостійної роботи здобувачів – 94 год.

Форма здобуття освіти – денна, заочна, дистанційна.

Дисципліна – обов'язкова.

Види навчальної діяльності – лекції, лабораторні роботи, розрахункова робота, самостійна робота здобувача.

Види контролю – модульний контроль, іспит.

Мова викладання – українська.

Пререквізити – «Інтелектуальна власність», «Експертологія», «Екологія програмного забезпечення», «Екологія програмного забезпечення (КП)», «Проектування програмного забезпечення систем реального часу».

Кореквізити – «Організаційне проектування та автоматизація процесів управління підприємствами», «Технології візуалізації інформації», «Надійність програмно-апаратних комплексів», «Науково-дослідна робота магістра (КП)», «Дипломне проектування», «Переддипломна практика».

3. Мета та завдання навчальної дисципліни

Мета: надання знань про основні поняття, концепції й інструменти мережевого обладнання, застосунків та сервісів. Застосування Agile-підходу задля усунення організаційних і тимчасових бар'єрів між командами розробників та іншими учасниками життєвого циклу програмного забезпечення (тестувальниками, адміністраторами, техпідтримкою).

Завдання: надання практичних навичок використання інструментів автоматизації для підвищення ефективності процесів розробки і експлуатації програмного забезпечення за рахунок їх безперервної інтеграції та активної взаємодії профільних фахівців.

Компетентності, які набуваються:

Загальні компетентності:

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність спілкуватися іноземною мовою як усно, так і письмово.

ЗК04. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами інших галузей знань/видів економічної діяльності).

ЗК05. Здатність генерувати нові ідеї (креативність).

Фахові компетентності:

ФК01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.

ФК02. Здатність розробляти і реалізовувати наукові та/або прикладні проєкти у сфері інженерії програмного забезпечення.

ФК06. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення.

ФК07. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах.

ФК08. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення.

ФК10. Здатність визначати мету та стратегію розвитку програмного продукту відповідно до політики організації в частині сталого розвитку.

ФК11. Здатність забезпечувати використання інновацій та поліпшень, які підвищують конкурентоспроможність або ефективність.

ФК12. Здатність демонструвати вищому керівництву бізнес-переваги потенційних змін.

Очікувані результати навчання:

ПРН01. Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення

ПРН03. Будувати і досліджувати моделі інформаційних процесів у прикладній області.

ПРН04. Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення.

ПРН06. Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

ПРН07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.

ПРН08. Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.

ПРН09. Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.

ПРН11. Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.

ПРН12. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики.

ПРН13. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

ПРН14. Прогнозувати розвиток програмних систем та інформаційних технологій.

ПРН15. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

ПРН18. Вміти документувати результати розробок та аналізу для представлення їх колегам, керівництву, замовнику.

4. Зміст навчальної дисципліни

Модуль 1.

Змістовий модуль № 1. Основи DevOps. Організація взаємодії між окремими співробітниками та командами.

Тема 1. Основи DevOps. *Культура розгортання програмного забезпечення. Еволюція культури розгортання програмного забезпечення. Історії, що ілюструють практики DevOps. Розвиток методологій розробки програмного забезпечення. Гнучка інфраструктура. Методології розробки програмного забезпечення. Методології експлуатації та системні методології. Концепції розробки, релізу та розгортання програмного забезпечення. Концепції, що стосуються інфраструктури. Культурні концепції. Загальні омани, пов'язані з DevOps. Антишаблони DevOps. Основні засади DevOps.*

Тема 2. Співробітництво всередині однієї команди. *Визначення співробітництва. Індивідуальні відмінності та навички. Можливості по досягненню конкурентних переваг. Наставництво. Знайомство з образом мислення. Образ мислення та навчаючі організації. Роль зворотного зв'язку. Огляди та рейтинги. Стилі спілкування та вирішення конфліктів. Емпатія та довіра. Персонал та кадрові ресурси. Омани, пов'язані зі співробітництвом. Вирішення проблем, пов'язаних зі співробітництвом.*

Тема 3. Близькість між командами. *Демонстраційний приклад з розробки програмного забезпечення. Мережі. Фактори створення команди. Командна та організаційна структура. Пошук точок дотику між командами. Переваги посиленої близькості. Вимоги до близькості. Співробітництво та кооперація. Омани, пошук та вирішення проблем.*

Модульний контроль

Модуль 2.

Змістовий модуль № 2. Впровадження DevOps. Інструменти, масштабування, існуючі практики.

Тема 4. Інструменти впровадження DevOps. *Інструменти розробки програмного забезпечення. Інструменти автоматизації. Моніторинг. Еволюція екосистеми інструментів. Значення інструментів для людей. Визначення інструментів. Вибір необхідних інструментів для вирішення реальних проблем. Область охоплення проєктів з відкритим кодом. Стандартизація інструментів. Послідовні процеси аналізу інструментів. Виключення зі стандартизації. Даремність інструментів. Вплив інструментів на культуру. Вибір інструментів. Аудит екосистеми інструментів. Усунення інструментів. Практики. Проблеми, пов'язані з мотивацією та процесом прийняття рішень. Омани, пошук та усунення проблем, пов'язаних з інструментами.*

Тема 5. Масштабування і його роль в процесі впровадження DevOps. *Знайомство з масштабуванням. Розгляд корпоративних DevOps-практик. Міркування з виконання масштабування. Організаційна структура. Командна гнучкість. Життєвий цикл організації. Складність та зміни. Масштабування команд та стратегія зростання. Масштабування організації. Практики. Омани та усунення проблем, пов'язаних з масштабуванням.*

Тема 6. Об'єднання культур DevOps. *Важливість історій. Теорія та практика DevOps. Взаємодія між організаціями. Заохочення близькості між організаціями. Впровадження ефективних DevOps-практик.*

Модульний контроль

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

До самостійної роботи здобувачів належить опрацювання лекційного матеріалу, підготовка до практичних занять, а також самостійне вивчення тем та питань, які не викладаються на лекційних заняттях.

В таблицях 1 – 3 подано розподіл та обсяг аудиторної та самостійної робіт здобувачів.

Таблиця 1 – Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	Денна форма				
	Усього	У тому числі			
		л	п	лаб.	с.р.
1	2	3	4	5	6
Змістовий модуль 1.					
<i>(Основи DevOps. Організація взаємодії між окремими співробітниками та командами)</i>					
Тема 1. (Основи DevOps. Методології експлуатації та системні методології. Концепції розробки, релізу та розгортання програмного забезпечення)	27	6		4	17
Тема 2. (Співробітництво всередині однієї команди. Наставництво)	24	5		4	15
Тема 3. (Близькість між командами. Демонстраційний приклад з розробки програмного забезпечення)	24	5		4	15
Разом за змістовим модулем 1	75	16		12	47
Змістовий модуль 2.					
<i>(Впровадження DevOps. Інструменти, масштабування, існуючі практики)</i>					
Тема 4. (Інструменти впровадження DevOps. Інструменти автоматизації. Моніторинг. Еволюція екосистеми інструментів)	27	6		4	17
Тема 5. (Масштабування і його роль в процесі впровадження DevOps)	24	5		4	15
Тема 6. (Об'єднання культур DevOps. Впровадження ефективних DevOps-практик)	24	5		4	15
Разом за змістовим модулем 2	75	16		12	47
Усього годин	150	32		24	94

Таблиця 2 – Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1.	Встановлення та налаштування віртуальної машини з Linux. Аналіз, оптимізація та аварійні роботи в Linux	2
2.	Програмування та автоматизація. Система керування конфігураціями Ansible	4
3.	Хмарні технології та мікросервісна архітектура. Деплой тестової програми в кластері Kubernetes: контейнеризовані програми в Kubernetes і Docker, а також застосування систем управління віртуалізацією	10
4.	Налагоджування процесів безперервної інтеграції та доставки коду. Налаштування системи моніторингу	4
5.	Управління за методологією Agile. Використання при роботі над програмним продуктом таких інструментів, як: user stories, jobs to be done, lean canvas	4
	Разом	24

Таблиця 3 – Теми для самостійної роботи

№ з/п	Назва теми	Кількість годин
1.	Налагодження багтрекінгової системи	29
2.	Підняття серверу збірки проекту	15
3.	Налагодження TeamCity	27
4.	Інтеграція проекту до систем безперервної інтеграції	23
	Разом	94

5. Індивідуальні завдання

Розрахункова робота (PP). Розгорнути з нуля інфраструктуру та Ваш додаток у хмарі за допомогою GitLab CI/CD, використовуючи концепцію «Інфраструктура як код» (IaC, Terraform).

Завдання проекту:

- підняти 3 оточення у хмарі, включаючи kubernetes (k8s) та бази даних (DBaaS) з використанням Terraform;
- налаштувати Pipeline для Вашої програми, включаючи тестування та канарковий деплой (Canary Deployment);
- налаштувати логування, трасування та моніторинг програми.

6. Методи навчання

1. За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: лабораторна робота, домашнє завдання.

2. За характером пізнавальної діяльності тих, хто навчається, – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.

3. За логікою пізнання – індуктивний, дедуктивний, метод аналогій, метод вивідних знань.

4. Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні питання до лабораторної роботи, тестування (традиційне та машинне).

Під час викладання дисципліни буде використовуватися презентаційна модель проведення лекційних занять з використанням проектору.

При викладанні навчальної дисципліни для активізації навчального процесу передбачено застосування сучасних навчальних технологій, таких, як: *проблемні лекції; робота в малих групах; презентації; кейс-метод.*

Проблемні лекції спрямовані на розвиток логічного мислення студентів. Коло питань теми лекції обмежується двома-трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов широкого відображення в підручниках, використовується досвід закордонних навчальних закладів з роздаванням студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються.

При викладанні лекційного матеріалу студентам пропонуються питання для самостійного розмірковування. При цьому лектор задає запитання, які спонукають студента шукати розв'язання проблемної ситуації. Така система примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді.

На початку проведення проблемної лекції необхідно чітко сформулювати проблему, яку необхідно вирішити студентам. При викладанні лекційного матеріалу слід уникати прямої відповіді на поставлені запитання, а висвітлювати лекційний матеріал таким чином, щоб отриману інформацію студент міг використовувати при розв'язанні проблеми.

Міні-лекції передбачають викладення навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції проводяться, як правило, як

частина заняття-дослідження. На початку проведення міні-лекції за вказаними темами лектор акцентує увагу студентів на необхідності представити викладений лекційний матеріал у так званому структурно-логічному вигляді. На розгляд виносяться питання, які зафіксовані у плані лекцій, але викладаються вони стисло.

Лекційне заняття, проведене у такий спосіб, пробуджує у студента активність та увагу при сприйнятті матеріалу, а також спрямовує його на використання системного підходу при відтворенні інформації, яку він одержав від викладача. Проблемні лекції та міні-лекції доцільно поєднувати з такою формою активізації навчального процесу, як робота в малих групах.

Робота в малих групах дає змогу структурувати лекційні або лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування.

Після висвітлення проблеми (при використанні проблемних лекцій) або стислого викладання матеріалу (при використанні міні-лекцій) студентам пропонується об'єднуватися у групи по 5-6 осіб та презентувати наприкінці заняття своє бачення та сприйняття матеріалу.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань.

Однією з позитивних рис презентації та її переваг при використанні в навчальному процесі є обмін досвідом, який здобули студенти при роботі у певній малій групі.

Кейс-метод – метод аналізу конкретних ситуацій, який дає змогу наблизити процес навчання до реальної практичної діяльності спеціалістів і передбачає розгляд виробничих, управлінських та інших ситуацій, складних конфліктних випадків, проблемних ситуацій, інцидентів у процесі вивчення навчального матеріалу.

7. Методи контролю

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.
4. Індивідуальна розрахункова робота (РР).
5. Форма підсумкового контролю успішності навчання: іспит (письмово).

8. Критерії оцінювання та розподіл балів, які отримують здобувачі

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	4...8	2	8...16
Модульний контроль	8...10	1	8...10
Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 2			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	3...6	2	6...12
Модульний контроль	4...8	1	4...8
Усього за семестр			60...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з двох теоретичних питань (кожне питання 30 балів) та одного практичного питання (питання оцінюється в 40 балів).

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Здати індивідуальне завдання та основні лабораторні. здати тестування. Знати основи операційної системи Linux, принципи створення комп'ютерних мереж, основні мережеві протоколи, основні сервіси на Linux для підприємства. Мати базові знання про системи віртуалізації, контейнеризацію (віртуалізація на рівні ядра ОС), основи хмарних технологій, мікросервісної архітектури та контейнеризації.

Вміти обирати програмні засоби супроводження розробки програмного забезпечення; за допомогою програмних засобів планувати та спостерігати за ходом виконання робіт, їх контроль і коректування прийнятих рішень.

Добре (75-89). Твердо знати мінімум, здати всі лабораторні роботи, індивідуальне завдання та тестування. Досконало знати основні сервіси на Linux

для підприємства: зберігання даних, роутер на Linux, забезпечення безпеки, мережеві сервіси, DNS, PKI та OpenVPN, Пошта, балансування навантаження, сервіси високої доступності. Добре орієнтуватися у системах віртуалізації (технології ефективної віртуалізації, гіпервізори 2-го типу (інтегровані з хост-системою), гіпервізори 1-го типу (bare metal hypervisors), вкладена віртуалізація та віртуалізація введення-виводу. Добре розумітися на контейнеризації, хмарних технологіях (види хмарних сервісів (*-as-a-Service), Cloud-native архітектурі та Infrastructure as Code (IaC), а також мікросервісній архітектурі та контейнеризації (Docker, зберігання даних та ресурси, мережеві абстракції Kubernetes). Досконало вміти обирати програмні засоби супроводження розробки програмного забезпечення; за допомогою програмних засобів планувати та спостерігати за ходом виконання робіт, їх контроль і коректування прийнятих рішень.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та вміти застосовувати їх.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

9. Політика навчального курсу

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем. Питання, що стосуються академічної доброчесності, розглядає викладач або за процедурою, визначеною у Положенні про академічну доброчесність.

10. Методичне забезпечення

Підручники, навчальні посібники, навчально-методичні посібники, конспекти лекцій, методичні рекомендації з проведення лабораторних робіт тощо, які видані в Університеті, знаходяться за посиланням:

http://library.khai.edu/catalog?clear_all_params=0&mode=BookList&lang=rus&xt=no&theme_path=0%2C1665%2C14849%2C13426%2C10001&themes_basket=&ttp_themes_basket=&disciplinesearch=no&top_list=1&fullsearch fld=&author fld=&docname fld=&docname cond=beginwith&theme context=&theme cond=all theme&theme_id=10001&is_ttp=0&combiningAND=0&step=20&tpage=1

Сторінка дисципліни знаходиться за посиланням:

<https://mentor.khai.edu/enrol/index.php?id=6754>

11. Рекомендована література

Базова

1. Michael Hüttermann. DevOps for Developers. Apress, 2012. – 184 p.
2. Viktor Farcic. The DevOps 2.0 Toolkit. Viktor Farcic, 2016 – 404 p.
3. Len Bass, Ingo Weber, Liming Zhu. DevOps A Software Architect's Perspective. Addison-Wesley, 2015. – 338 p.
4. DevOps For Dummies. IBM Limited Edition, 2014 – 51 p.
5. Я. Ван Бон, Г. Кеммерлинг, Д. Пондман. Введение в ИТ сервис-менеджмент. М. : Компания «ИТ-Expert», 2013. – 225 с.
6. Ингланд Р. Введение в реальный ITSM : пер. с англ. М.: Лайвбук, 2011. – 132 с.
7. Humble, Jez, and Farley, David. Continuous Delivery . Upper Saddle River, NJ: Addison-Wesley, 2010.
8. Poppendieck, Mary, and Thomas David Poppendieck. Implementing Lean Software Development . Upper Saddle River, NJ: Addison-Wesley, 2007.
9. Walls, Mandi. Building a DevOps Culture . Sebastopol, CA: O'Reilly Media, 2013.
10. Dekker, Sidney. Field Guide to Understanding Human Error . Farnham, UK: Ashgate Publishing, 2006.
11. Version Control Systems [Електронний ресурс]. – Доступ: <https://www.geeksforgeeks.org/version-control-systems/>
12. 15 BEST Version Control Software (Source Code Management Tools 2021): Best Version Control Software Tools and Systems [Електронний ресурс]. – Доступ: <https://www.softwaretestinghelp.com/version-control-software/>
13. What is the Difference Between Continuous Integration, Continuous Deployment and Continuous Delivery? [Електронний ресурс]. – Доступ: <https://www.cloudbees.com/continuous-delivery/continuous-integration>
14. Software testing: Find software errors and verify that an application or system is fit for use [Електронний ресурс]. – Доступ: <https://www.ibm.com/topics/software-testing>

Додаткова

1. Greaves, Karen, and Samantha Laing. Collaboration Games from the Growing Agile Toolbox . Victoria, BC: Leanpub/Growing Agile, 2014.
2. Cowie, Jon. Customizing Chef . Sebastopol, CA: O'Reilly Media, 2014.
3. Dixon, Jason. Monitoring with Graphite . Sebastopol, CA.: O'Reilly Media, 2015.
4. 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.
5. Reed, J. Paul. DevOps in Practice . Sebastopol, CA: O'Reilly Media, 2013.
6. Дюваль, Поль М., Матіас III, Стівен М., Гловер, Ендрю. Безперервна інтеграція: покращення якості програмного забезпечення та зниження ризику. : Пров. з англ. – М.: ТОВ “І.Д. Вільямс”, 2008. – 240 с. : іл. – Парал. тит. англ.
7. У. Ройс. Управління проектами зі створення програмного забезпечення. Уніфікований підхід. – Лорі, 2002.
8. К. Бек. Екстремальне програмування. – Вид-во «Пітер», 2002.
9. Тестування Дот Ком, або посібник по жорсткому поводженню з багами в інтернет-стартапах. – М. : Справа, 2007. – 312 с.
10. PMBOK – Project Management Body of Knowledge, USA, 2005.
11. Ф. Брукс. Міфічний людино-місяць, або як створюються програмні системи. – «Символ-Плюс», 2006.

12. Інформаційні ресурси

1. Allspaw, John. «A Mature Role for Automation: Part 1». KitchenSoap.com, September 21, 2012. <http://bit.ly/allspaw-automation>
2. Caum, Carl. «Continuous Delivery vs. Continuous Deployment: What's the Diff?» Puppet blog, August 30, 2013. <http://bit.ly/cd-vs-cd>
3. Coutinho, Rodrigo. «In Support of DevOps: Kanban vs. Scrum». DevOps.com, July 29, 2014. <http://bit.ly/kanban-v-scrum>
4. Humble, Jez. «Deployment pipeline anti-patterns». <http://bit.ly/humbleantipatterns>
5. Kim, Gene. Kanbans and DevOps: «Resource Guide for Phoenix Project (Part 2)» IT Revolution Press, N.d. <http://bit.ly/kanbans-devops>
6. Arrested DevOps <https://www.arresteddevops.com/>
7. DevOps Cafe Podcast with John Willis and Damon Edwards <http://devopscafe.org/>
8. CI/CD Full Course | CI/CD Tutorial | Continuous Integration And Continuous Delivery | Simplilearn <https://www.youtube.com/watch?v=h9K1NnqwUvE>

9. Pro Git. Scott Chacon. 2012-08-31. This is the PDF file for the Pro Git book contents. It is licensed under the Creative Commons Attribution-Non Commercial-Share Alike 3.0 license. I hope you enjoy it, I hope it helps you learn Git, and I hope you'll support Apress and me by purchasing a print copy of the book at Amazon: <http://tinyurl.com/amazonprogit>

10. Ben Collins-Sussman. Version Control with Subversion. For Subversion 1.7 (Compiled from r5206). Ben Collins-Sussman Brian W. Fitzpatrick C. Michael Pilato. 2011, 433 p. This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

11. Бен Коллінз-Сассмен, Брайан Фитцпатрик. Ідеальна ІТ-компанія. Як з «Гітов» зібрати команду програмістів. – ТОВ Видавництво «Пітер», 2014. – 230 с

12. Mercurial: The Definitive Guide by Bryan O'Sullivan <http://hgbook.red-bean.com/read/>

13. Куликов Святослав. Тестирование программного обеспечения. Базовый курс. – EPAM Systems, 2016. – 289 с. http://svyatoslav.biz/software_testing_book/

14. S. [Guckenheimer](#), J.[Perez](#). Software Engineering with Microsoft Visual Studio Team System. – Addison-Wesley Professional, 2006.