

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

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньої програми

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

**СИЛАБУС ВИБІРКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Інформаційні технології розробки програмного забезпечення
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»
(код та найменування спеціальності)

Освітня програма: «Інженерія програмного забезпечення»
(найменування освітньої програми)

Рівень вищої освіти: *перший (бакалаврський)*

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  І. Б. Туркін
(науковий ступінь і вчене звання) (підпис) (ініціали та прізвище)

Погоджено з представником здобувачів освіти:

Представник студентського самоврядування  Д. В. Колодій
(підпис) (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. 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 – 5 семестр.

Обсяг дисципліни: 6 кредитів ЄКТС/180 годин, у тому числі аудиторних – 80 год, самостійної роботи здобувачів – 100 год.

Форма здобуття освіти – денна, заочна, дистанційна.

Дисципліна – вибіркова.

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

Види контролю – модульний контроль, іспит.

Мова викладання – українська.

Пререквізити – «Комп'ютерні мережі та HTML», «Об'єктно-орієнтоване програмування».

Кореквізити – «Міжнародні стандарти і менеджмент проектів програмного забезпечення» або «Економіка ІТ проектів», а також «Архітектура та проектування програмного забезпечення .Net» та «Якість програмного забезпечення та тестування».

3. Мета та завдання навчальної дисципліни

Мета вивчення: надання знань про використання інформаційних технологій у життєвому циклі програмного забезпечення.

Завдання: в результаті навчання студенти матимуть практичні навички з використання сучасних інструментальних засобів для аналізування стану і розроблення програмного забезпечення.

Компетентності, які набуваються:

Загальні компетентності:

- ЗК02. Здатність застосовувати знання у практичних ситуаціях.
- ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК12. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.

Фахові компетентності:

- ФК01. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.
- ФК02. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- ФК03. Здатність розробляти архітектури, модулі та компоненти програмних систем.
- ФК07. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.
- ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.
- ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
- ФК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.
- ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.
- ФК14. Здатність до алгоритмічного та логічного мислення.

Очікувані результати навчання:

- ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
- ПРН02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.
- ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.
- ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.
- ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.
- ПРН06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.
- ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.
- ПРН09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.
- ПРН10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.
- ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.
- ПРН16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.
- ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.
- ПРН18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.
- ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.
- ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.
- ПРН21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.
- ПРН22. Знати та вміти застосовувати методи та засоби управління проектами.
- ПРН26. Вміти конфігурувати компоненти, організувати робочий процес розгортання та тестувати рішення в його остаточному операційному середовищі.

4. Зміст навчальної дисципліни

Модуль 1.

Змістовний модуль №1. Основні принципи командної розробки програмного забезпечення.

Тема 1. Введення. Промисловий підхід до розробки програмного забезпечення (ПЗ). Життєвий цикл програмного продукту. Основи програмної інженерії. *Вводяться основні поняття і проблематика командної розробки програмного забезпечення. Розкриваються основні методики промислового підходу до розробки ПЗ. Розглядається з точки зору життєвого циклу програмного продукту залучення учасників проекту у процесі розробки.*

Тема 2. Специфіка командної розробки. Методології створення ПЗ. RUP, XP, MSF. Інструментальні середовища підтримки розробки. *Розглядаються основні принципи роботи у команді та інструментальні засоби взаємодії учасників проекту. Вводиться поняття інтеграційного серверу збірки проекту, централізованого репозиторія зберігання артефактів проекту.*

Тема 3. Моделі і дисципліни MSF. Модель архітектури, модель проектної групи, модель процесів. *Надається поняття моделі проектної групи MSF (MSF Team Model), що описує підхід до організації працюючого над проектом персоналу та його діяльності з метою максимізації успішності проекту. Розглядаються рольові кластери, їх області компетенції та зони відповідальності, а також рекомендації членам проектної групи, що дозволяють їм успішно здійснити свою місію щодо втілення проекту в життя.*

Тема 4. Дисципліни управління проектами, ризиками, готовністю. MSF CMMI і Agile. *Вводяться поняття готовності проекту до релізу. Оцінювання ризиків зміщення терміну готовності проекту.*

Модульний контроль

Модуль 2.

Змістовний модуль №2. Підходи до проектування архітектури програмного забезпечення.

Тема 5. Етапи моделі процесів, проектування архітектури програмного забезпечення (ПЗ). Огляд та аналіз інформації. *Розглядається послідовність дій команди проекту при розробці та впровадженні ПЗ на кожному етапі. Представлені описи робіт, а також шаблони та зразками документів, що супроводжують проектування ПЗ.*

Тема 6. Створення загальної картини рішення. Концептуальний дизайн. Логічний дизайн. *Розглядається формування принципової ідеї, опрацювання стратегій і підходів, що будуть використані для проектування, а також майбутні принципи взаємодії різних учасників комунікації.*

Тема 7. Етапи моделі процесів, реалізація програми. Фізичний дизайн. Презентаційний рівень. Рівень даних. Специфікації безпеки. *Розглядається розділення моделі на рівні. Дається класифікація деталізації проектування для кожного рівня процесів реалізації.*

Тема 8. Стабілізація і розгортання. Стабілізація рішення. Тестування. Випуск. Пілотна експлуатація. Розгортання в промисловому середовищі. Аналіз та обговорення проекту. *Надається опис процесу стабілізації проекту, вводиться поняття заморожки кодування, нульового рівня помилок та завершеності технічної документації. Підведення підсумків успішного проекту та зберігання результатів проекту для подальших проектів організації.*

Модульний контроль

Змістовний модуль №3. SOLID. CI / CD. Версіонування. Сертифікація.

Тема 9. Наскрізні процеси підтримки розробки. Управління проектами, ресурсами, командою. Розглядаються інформаційні технології, що супроводжують комунікації на проекті поміж розробниками. Надається приклад оцінювання трудоємності задач.

Тема 10. Управління ризиками, вимогами і якістю. Управління змінами та конфігураціями. Розглядаються вплив змiну вимог на різних стадіях готовності проекту. Описується алгоритм фіксації нових вимог та оновлення існуючих проектних документів.

Тема 11. Верифікація і тестування. Документування. Навчання персоналу. Розглядається модульне тестування. Ізоляція класів. Створення класів заглушок замість реальних класів, що вилучаються з тестування. Оцінка покриття коду тестами. Засоби представлення мануальних тестів, чек листи та тестові випадки.

Тема 12. Принципи успішної розробки ПЗ. Аналіз результатів домашнього завдання. Висновок. Підведення підсумків ітераційної розробки у команді. Оцінка залучення у процес розробки кожного учасника команди. Обговорення ролей у команді та засобів комунікації на проекті.

Модульний контроль

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

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. (Основні принципи командної розробки програмного забезпечення)					
Тема 1. (Введення. Промисловий підхід до розробки програмного забезпечення)	9	3		3	3
Тема 2. (Специфіка командної розробки. Методології створення ПЗ. RUP, XP, MSF)	17	2		5	10
Тема 3. (Моделі і дисципліни MSF. Модель архітектури, модель проектної групи, модель процесів)	17	2		5	10
Тема 4. (Дисципліни управління проектами, ризиками, готовністю. MSF CMMI і Agile)	16	3		3	10
Разом за змістовим модулем 1	59	10		16	33

Змістовий модуль 2. (Підходи до проектування архітектури програмного забезпечення)						
Тема 1. (Етапи моделі процесів, проектування архітектури ПЗ. Огляд та аналіз інформації)	9	2		5	2	
Тема 2. (Створення загальної картини рішення. Концептуальний дизайн. Логічний дизайн)	9	4		3	2	
Тема 3. (Етапи моделі процесів, реалізація програми. Фізичний дизайн. Презентаційний рівень. Рівень даних. Специфікації безпеки)	22	4		3	15	
Тема 4. (Стабілізація і розгортання)	22	2		5	15	
Разом за змістовим модулем 2	62	12		16	34	
Змістовий модуль 3. (SOLID. CI / CD. Версіонування. Сертифікація)						
Тема 1. (Наскрізні процеси підтримки розробки)	6	2		3	1	
Тема 2. (Управління ризиками, вимогами і якістю)	7	2		3	2	
Тема 3. (Верифікація і тестування. Документування. Навчання персоналу)	23	3		5	15	
Тема 4. (Принципи успішної розробки ПЗ. Аналіз результатів виконання РГР)	23	3		5	15	
Разом за змістовим модулем 3	59	10		16	33	
Усього годин	180	32		48		100

Таблиця 2 – Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1.	Основні компоненти Visual Studio Team System (VSTS). Методика установки компонентів VSTS. Організація інструментального середовища розробки.	6
2.	Завдання на розробку програмного рішення. Формування команд, ролі та зони відповідальності учасників	6
3.	Планування проекту. Організація взаємодії, інтерфейси, артефакти, шаблони, документи	6
4.	Діяльність учасників на різних етапах життєвого циклу розробки	6
5.	Проектування та реалізація програмного рішення. Стабілізація і розгортання.	6
6.	Робоче місце архітектора проекту, основні функції і можливості, зв'язок з розробниками і тестерами проекту	6
7.	Робоче місце розробника, основні функції та інструментальні засоби. Зв'язок з архітектором проекту і тестером	6

Продовження таблиці 2

№ з/п	Назва теми	Кількість годин
8.	Робоче місце тестера проекту, основні функції тестера і засоби тестування, зв'язок з розробником і архітектором проекту	6
	Разом	48

Таблиця 3 – Теми для самостійної роботи

№ з/п	Назва теми	Кількість годин
1.	Налагодження багтрекінгової системи	33
2.	Підняття серверу збірки проекту	17
3.	Налагодження TeamCity	27
4.	Інтеграція проекту до систем безперервної інтеграції	23
	Разом	100

5. Індивідуальні завдання

Розрахунково-графічна робота (РГР). Основна вимога – продемонструвати сучасні підходи до розробки програмного забезпечення (технології, методи, інструментальні засоби, фреймворки) на реальних існуючих проектах: https://mentor.khai.edu/pluginfile.php?file=%2F31421%2Fmod_resource%2Fcontent%2F1%2FHomework.pdf.

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.

7. Методи контролю

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.
4. Індивідуальна розрахунково-графічна робота (РГР).
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
Змістовний модуль 3			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	4...8	3	12...24
Модульний контроль	14...20	1	14...20
Виконання і захист РГР (РР, РК)	8..10	1	8..10
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 основні відомості про системи контролю версій; необхідний словник термінів для його застосування під час командного розроблення програмного забезпечення й використання веб-сервісів для хостингу проєктів і їхнього спільного розроблення, що ґрунтуються на системі контролю версій (Mercurial, Git або ін.); узагальнену архітектуру систем контролю версій; планування проєкту та відстеження змін у ньому, а також злиття версій вихідного коду проєкту; застосування Unit-тестування, а також патернів впровадження залежностей класів; основні відомості про делегати, анонімні методи, лямбда-вирази, а також переваги використання Моq-об'єктів; переваги версіонування програмних продуктів і створення інсталяторів, основні відомості про сертифікати прикладних програм, центри сертифікації, а також показано на прикладах підпис інсталятора сертифікатом. 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 відомості про системи контролю версій; необхідний словник термінів для його застосування під час командного розроблення програмного забезпечення й використання веб-сервісів для хостингу проєктів і їхнього спільного розроблення, що ґрунтуються на системі контролю версій (Mercurial, Git або ін.); узагальнену архітектуру систем контролю версій; планування проєкту та відстеження змін у ньому, а також злиття версій вихідного коду проєкту; застосування Unit-тестування, а також патернів впровадження залежностей класів; основні відомості про делегати, анонімні методи, лямбда-вирази, а також переваги використання Моq-об'єктів; переваги версіонування програмних продуктів і створення інсталяторів, основні відомості про сертифікати прикладних програм, центри сертифікації, а також показано на прикладах підпис інсталятора сертифікатом. 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 уміти 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=&doc name fld=&docname cond=beginwith&theme context=&theme cond=all theme&the me_id=10001&is_ttp=0&combiningAND=0&step=20&tpage=1

Сторінка дисципліни знаходиться за посиланням:

<https://mentor.khai.edu/course/view.php?id=226>

11. Рекомендована література

Базова

1. Version Control Systems [Електронний ресурс]. – Доступ: <https://www.geeksforgeeks.org/version-control-systems/>

2. 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/>

3. What is the Difference Between Continuous Integration, Continuous Deployment and Continuous Delivery? [Електронний ресурс]. – Доступ: <https://www.cloudbees.com/continuous-delivery/continuous-integration>
4. 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>
5. Create an installer from Visual Studio [Електронний ресурс]. – Доступ: <https://www.advancedinstaller.com/user-guide/tutorial-ai-ext-vs.html>
6. Ericsson A.B. Secure Socket Layer. – ERLANG, September 20, 2016. – 35 p.

Додаткова

1. Дюваль, Поль М., Матіас ІІІ, 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.
2. У. Ройс. Управління проектами зі створення програмного забезпечення. Уніфікований підхід. – Лорі, 2002.
3. К. Бек. Екстремальне програмування. – Вид-во «Пітер», 2002.
4. S. Guckenheimer, J.Perez. Software Engineering with Microsoft Visual Studio Team System. – Addison-Wesley Professional, 2006.
5. Шторгіна О. С. Системи контролю версій. Мережеві технології. Технології Інтернет. Лекція 2. – 82 с.
6. Аналіз вимог і створення архітектури рішень на основі Microsoft .NET, Російська редакція, 2004.
7. Тестування Дот Ком, або посібник по жорсткому поводженню з багами в інтернет-стартапах. – М. : Справа, 2007. – 312 с.
8. Сергій Орлик. Програмна інженерія. Тестування програмного забезпечення. – 2004 – 2005. – 16 с.
9. Марк Сіман. Впровадження залежностей. - Вид-во «Пітер», 2013. – 624 с.
10. Липа В. В. Сертифікація програмних засобів: навч. посіб. – М.: Сінтег, 2010. – 348 с.
11. PMBOK – Project Management Body of Knowledge, USA, 2005.
12. Ф. Брукс. Міфічний людино-місяць, або як створюються програмні системи. – «Символ-Плюс», 2006.
13. Е. Йордан. Шлях камікадзе. – Лорі, 2001.
14. А. Н. Терехов. Технологія програмування. – «Інтуїт Біном», 2007.

12. Інформаційні ресурси

1. CI/CD Full Course | CI/CD Tutorial | Continuous Integration And Continuous Delivery | Simplilearn <https://www.youtube.com/watch?v=h9K1NnqwUvE>
2. 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>

3. 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.

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

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

6. <https://hgbook.bacher09.org/html/index.html>

7. <https://dou.ua/lenta/articles/mercurial-step-by-step-dvcs-intro/>

8. Ben Lynn. Волшебство Git. Руководство выпущено под GNU General Public License 3-й версии. Исходный текст находится в хранилище Git и может быть получен командой: \$ git clone git://repo.or.cz/gitmagic.git #.

<http://www-cs-students.stanford.edu/~blynn/gitmagic/intl/ru/index.html>

9. <http://tortoisehg.bitbucket.org/screenshots.html#mq>

10. <http://tortoisehg.bitbucket.org/>

11. <https://habrahabr.ru/post/154255/>

12. <http://rus-linux.net/MyLDP/BOOKS/Architecture-Open-Source-Applications/Vol-2/git-02.html>

13. Іртегов Д. В. Лекції по інструментах управління конфігурацією <http://parallels.nsu.ru/~fat/subversion.ppt>

14. Керування версіями у Subversion <http://svnbook.red-bean.com/>

15. Сайт із матеріалами курсу: <http://www.inteks.ru/PM/>

16. <https://confluence.atlassian.com/bamboo/mercurial-289277014.html>

17. https://developer.mozilla.org/en-US/docs/Mercurial/Using_Mercurial

18. Огляд систем контролю версій http://all-ht.ru/inf/prog/p_0_1.html

19. Рейтинг систем контролю версій <http://tagline.ru/version-control-systems-rating/>

20. Порівняння систем контролю версій. http://mitra.ru/events/news/website_3.html

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

22. [https://msdn.microsoft.com/ru-ru/library/stxxt3fd\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/stxxt3fd(v=vs.90).aspx)

23. Нумерація версій <https://habrahabr.ru/post/119400/>

24. S. Guckenheimer, J. Perez. Software Engineering with Microsoft Visual Studio Team System. – Addison-Wesley Professional, 2006.

25. http://wixtoolset.org/documentation/manual/v3/howtos/updates/major_upgrade.html

26. [https://msdn.microsoft.com/en-us/library/windows/desktop/aa370037\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa370037(v=vs.85).aspx)

27. <https://www.safaribooksonline.com/library/view/wix-36-a/9781782160427/ch13s04.html>

28. <http://helpnet.flexerasoftware.com/installshield22helplib/helplib/MajorMinorSmall.htm> – 2015.