

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

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Гарант освітньої програми



(підпис)

I.В. Шевченко
(ініціали та прізвище)

« 31 » 08 2021 р.

**РОБОЧА ПРОГРАМА ВИБІРКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Інформаційні технології розробки програмного забезпечення
(назва навчальної дисципліни)

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

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

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

Форма навчання: денна

Рівень вищої освіти: початковий рівень (короткий цикл) вищої освіти

Харків 2021 рік

Розробник: Кузнецова Ю.А., доц. каф. 603, канд. техн. наук., доц.
(прізвище та ініціали, посада, наукова ступінь та вчене звання)



Робочу програму розглянуто на засіданні кафедри інженерія програмного забезпечення

(назва кафедри)

Протокол № 2 від « 31 » 08 2021 р.

Завідувач кафедри д-р техн. наук., проф.
(науковий ступінь і вчене звання)


(підпис)

І.Б. Туркін
(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 4,5	Галузь знань <u>12 «Інформаційні технології»</u> (шифр та найменування) Спеціальність <u>121 «Інженерія програмного забезпечення»</u> (код та найменування) Освітня програма <u>«Інженерія програмного забезпечення»</u> (найменування) Рівень вищої освіти: <u>початковий рівень</u> <u>(короткий цикл) вищої освіти</u>	<i>Вибіркова</i>
Кількість модулів – 2		Навчальний рік
Кількість змістових модулів – 3		2021 / 2022
Індивідуальне завдання: не передбачено навчальним планом (назва)		Семестр
		3-й
Загальна кількість годин: 48 ¹⁾ / 135		Лекції ¹⁾
		24 годин
		Практичні, семінарські¹⁾
		0 годин
		Лабораторні ¹⁾
Кількість тижневих годин для денної форми навчання: аудиторних – 3 самостійної роботи студента – 5,44		24 годин
		Самостійна робота
		87 годин
		Вид контролю: модульний контроль, іспит

Співвідношення кількості годин аудиторних занять до самостійної роботи становить:

48 / 87.

¹⁾ Аудиторне навантаження може бути зменшене або збільшене на одну годину в залежності від розкладу занять.

2. 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:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК5. Здатність вчитися і оволодівати сучасними знаннями.

ЗК7. Здатність працювати в команді.

Фахові компетентності:

ФК1. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК4. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

Очікувані результати навчання:

ПРН2. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН8. Застосовувати на практиці ефективні підходи до детального проектування програмного забезпечення.

ПРН16. Вміти документувати та презентувати результати розробки програмного забезпечення.

Пререквізити – «Основи програмної інженерії», «Основи програмування», «Програмування мовою C#».

Кореквізити – «Якість програмного забезпечення та тестування», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування».

3. Програма 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 заморозки 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 вилючаються 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

4. 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)	7	2		2	3
Тема 2. (Специфіка командної розробки. Методології створення ПЗ. RUP, XP, MSF)	14	2		2	10
Тема 3. (Моделі і дисципліни MSF. Модель архітектури, модель проектної групи, модель процесів)	14	2		2	10
Тема 4. (Дисципліни управління проектами, ризиками, готовністю. MSF CMMI і Agile)	14	2		2	10
Разом за змістовим модулем 1	49	8		8	33
Змістовий модуль 2. (Підходи до проектування архітектури програмного забезпечення)					
Тема 1. (Етапи моделі процесів, проектування архітектури ПЗ. Огляд та аналіз інформації)	6	2		2	2
Тема 2. (Створення загальної картини рішення. Концептуальний дизайн. Логічний дизайн)	6	2		2	2
Тема 3. (Етапи моделі процесів, реалізація програми. Фізичний дизайн. Презентаційний рівень. Рівень даних. Специфікації безпеки)	15	2		2	11
Тема 4. (Стабілізація і розгортання)	16	2		2	12
Разом за змістовим модулем 2	43	8		8	27
Змістовий модуль 3. (SOLID. CI / CD. Версіонування. Сертифікація)					
Тема 1. (Наскрізні процеси підтримки розробки)	5	2		2	1
Тема 2. (Управління ризиками, вимогами і якістю)	6	2		2	2
Тема 3. (Верифікація і тестування. Документування. Навчання персоналу)	16	2		2	12

Тема 4. (Принципи успішної розробки ПЗ. Аналіз результатів виконання РГР)	16	2		2	12
Разом за змістовим модулем 3	43	8		8	27
Усього годин	135	24		24	87

5. Теми семінарських занять

№ з/п	Назва теми	Кількість годин
1		-
2		-
3		-
	Разом	0

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1		-
2		-
	Разом	0

7. 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.	3
2.	Завдання на розробку програмного рішення. Формування команд, ролі та зони відповідальності учасників	3
3.	Планування проекту. Організація взаємодії, інтерфейси, артефакти, шаблони, документи	3
4.	Діяльність учасників на різних етапах життєвого циклу розробки	3
5.	Проектування та реалізація програмного рішення. Стабілізація і розгортання.	3
6.	Робоче місце архітектора проекту, основні функції і можливості, зв'язок з розробниками і тестерами проекту	3

7.	Робоче місце розробника, основні функції та інструментальні засоби. Зв'язок з архітектором проекту і тестером	3
8.	Робоче місце тестера проекту, основні функції тестера і засоби тестування, зв'язок з розробником і архітектором проекту	3
	Разом	24

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1.	Налагодження багтрекінгової системи	25
2.	Підняття серверу збірки проекту	15
3.	Налагодження TeamCity	25
4.	Інтеграція проекту до систем безперервної інтеграції	22
	Разом	87

9. Індивідуальні завдання (виконується обов'язково)

Не передбачено навчальним планом.

10. 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.

11. 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).

12. Критерії оцінювання та розподіл балів, які отримують студенти

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 студента 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 проектів і їхнього спільного розроблення, що основані на системі контролю версій (Mercurial, Git або ін.); узагальнену архітектуру систем контролю версій; планування проекту та відстеження змін у ньому, а також злиття версій вихідного коду проекту; 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 проектів і їхнього спільного розроблення, що основані на системі контролю версій (Mercurial, Git або ін.); узагальнену архітектуру систем контролю версій; планування проекту та відстеження змін у ньому, а також злиття версій вихідного коду проекту; 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

13. Методичне забезпечення

1. Інформаційні технології розроблення програмного забезпечення : навч. посіб. / Ю. А. Кузнецова, І. Б. Туркін ; М-во освіти і науки України, Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харк. авіац. ін-т". – Харків. – Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харк. авіац. ін-т", 2017. – 151 с.

2. Інформаційні технології розроблення програмного забезпечення : навч. посіб. з виконання лабораторних робіт / М. О. Данова, Ю. А. Кузнецова, М. О. Сьомочкін ; М-во освіти і науки України, Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харк. авіац. ін-т". – Харків. – Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харк. авіац. ін-т", 2020. – 92 с.

Дистанційний курс дисципліни розроблено у системі дистанційного навчання Mentor, яку впроваджено в Національному аерокосмічному університеті імені М. Є. Жуковського «ХАІ», доступ до курсу за посиланням: <https://mentor.khai.edu/course/view.php?id=5804>

14. 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 р.

Додаткова

1. 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.
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.

15. Інформаційні ресурси

1. CI/CD Full Course | CI/CD Tutorial | Continuous Integration And Continuous Delivery | Simplilearn <https://www.youtube.com/watch?v=h9K1NnqwUvE>
2. Бен Коллінз-Сассмен, Брайан Фитцпатрик. Ідеальна ІТ-компанія. Як з «гіков» зібрати команду програмістів. – ТОВ Видавництво «Пітер», 2014. – 230 с
3. Mercurial: The Definitive Guide by Bryan O'Sullivan <http://hgbook.red-bean.com/read/>
4. <https://hgbook.bacher09.org/html/index.html>
5. <https://dou.ua/lenta/articles/mercurial-step-by-step-dvcs-intro/>

6. 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>
7. <http://tortoisehg.bitbucket.org/screenshots.html#mq>
8. <http://tortoisehg.bitbucket.org/>
9. <http://rus-linux.net/MyLDP/BOOKS/Architecture-Open-Source-Applications/Vol-2/git-02.html>
10. Керування версіями у Subversion <http://svnbook.red-bean.com/>
11. <https://confluence.atlassian.com/bamboo/mercurial-289277014.html>
12. https://developer.mozilla.org/en-US/docs/Mercurial/Using_Mercurial
13. Куликов Святослав. Тестирование программного обеспечения. Базовый курс. – ЕРАМ Systems, 2016. – 289 с.
http://svyatoslav.biz/software_testing_book/
14. [https://msdn.microsoft.com/ru-ru/library/stxxt3fd\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/stxxt3fd(v=vs.90).aspx)
15. S. Guckenheimer, J. Perez. Software Engineering with Microsoft Visual Studio Team System. – Addison-Wesley Professional, 2006.
16. http://wixtoolset.org/documentation/manual/v3/howtos/updates/major_upgrade.html
17. [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)
18. <https://www.safaribooksonline.com/library/view/wix-36-a/9781782160427/ch13s04.html>
19. <http://helpnet.flexerasoftware.com/installshield22helplib/helplib/MajorMinorSmall.htm> – 2015.