



# РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА МЕТОДОЛОГІЄЮ AGILE

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>4 рік підготовки, 7 семестр</i>
Обсяг дисципліни	<i>Лекції: 36 год., комп'ютерний практикум: 18 год., самостійна робота: 66 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року (rozklad.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., ст. викладач, Хіцко Яна Володимирівна, iana.khitsko@gmail.com Комп'ютерний практикум: к.т.н., ст. викладач, Хіцко Яна Володимирівна, iana.khitsko@gmail.com</i>
Розміщення курсу	Google classroom: <a href="https://classroom.google.com/c/NDE3OTQxMzE5NjU5?cjc=ne2zis4">https://classroom.google.com/c/NDE3OTQxMzE5NjU5?cjc=ne2zis4</a>

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

*Вивчення дисципліни «Розроблення програмного забезпечення за методологією Agile» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач*

професійної діяльності, пов'язаної із розробленням програмного забезпечення згідно сучасних гнучких методологій.

**Метою** вивчення дисципліни «Розроблення програмного забезпечення за методологією Agile» є формування у здобувачів освіти здатності обирати методологію розробки програмного забезпечення відповідно до визначених вимог та середовища проектування і конструювання; застосовувати різноманітні практики Agile при проектуванні та конструюванні програмного забезпечення; проводити аналіз ефективності застосування практик та підпроцесів з сімейства Agile.

**Предметом** дисципліни «Розроблення програмного забезпечення за методологією Agile» є алгоритмічне забезпечення процесів аналізу, проектування, моніторингу та конструювання програмного забезпечення.

Дисципліна «Розроблення програмного забезпечення за методологією Agile» підсилює такі фахові компетенції освітньо-професійної програми:

**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;

**ФК04** - Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами;

**ФК05** - Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу;

**ФК08** - Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;

**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 розроблення програмного забезпечення;

**ФК12** - Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

Дисципліна «Розроблення програмного забезпечення за методологією Agile» сприяє формуванню ПРН:

**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 розроблення програмного забезпечення;

**ПРН06** - Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;

**ПРН09** - Вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення;

**ПРН10** - Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;

**ПРН12** - Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення;

**ПРН20** - Знати підходи щодо оцінювання та забезпечення якості програмного забезпечення;

**ПРН23** - Вміти документувати та презентувати результати розроблення програмного забезпечення.

## **2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Успішному вивченню дисципліни «Розроблення програмного забезпечення за методологією Agile» передуює вивчення дисциплін «Компоненти програмної інженерії. Частина 4. Якість та тестування програмного забезпечення», «Компоненти програмної інженерії. Частина 2. Моделювання програмного забезпечення. Аналіз вимог до програмного забезпечення», «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Розроблення програмного забезпечення за методологією Agile» теоретичні знання та практичні уміння забезпечують успішне виконання курсових та дипломних проєктів за спеціальністю 121 Інженерія програмного забезпечення.

## **3. Зміст навчальної дисципліни**

Дисципліна «Розроблення програмного забезпечення за методологією Agile» передбачає вивчення таких тем:

Тема 1. Життєвий цикл розробки програмного забезпечення

Тема 2. Методології Agile

Тема 3. Основні практики Agile

Модульна контрольна робота

Залік

## **4. Навчальні матеріали та ресурси**

### **Базова література:**

1. Електронний кампус НТУУ «КПІ ім. Ігоря Сікорського». Матеріали з дисципліни «Розроблення програмного забезпечення за методологією Agile». – Режим доступу : <http://login.kpi.ua>
2. Web-портал ФПМ. Архів матеріалів. Тека «Хіцко». – Режим доступу : [http://fpm.kpi.ua/archive/dir.do?sys\\_id=obj\\_2](http://fpm.kpi.ua/archive/dir.do?sys_id=obj_2)

### **Додаткова література:**

3. Kent Beck. "Manifesto for Agile Software Development" [електронний ресурс] / Kent Beck, James Grenning, Robert C. Martin, Mike Beedle, Jim Highsmith, Steve Mellor, Arie van Bennekum, Andrew

Hunt, Ken Schwaber, Alistair Cockburn, Ron Jeffries, Jeff Sutherland, Ward Cunningham, Jon Kern, Dave Thomas, Martin Fowler, Brian Marick // Agile Alliance, 2001. – режим доступу - <http://agilemanifesto.org/>.

4. [Steve McConnell \(1996\). Rapid Development: Taming Wild Software Schedules, Microsoft Press Books, ISBN 978-1-55615-900-8](#)
5. [DSDM Consortium. DSDM Atern: the Handbook \[електронний ресурс\] / DSDM Consortium, 2008. – режим доступу - <https://www.agilebusiness.org/resources/dsdm-handbooks/dsdm-atern-handbook-2008>.](#)
6. [Schwaber, Ken \(2004\). "SCRUM Development Process"\(PDF\). Advanced Development Methods - режим доступу - <http://www.jeffsutherland.org/oopsla/schwapub.pdf>](#)
7. ["Extreme Programming Rules". extremeprogramming.org.](#)
8. [Palmer, S.R., & Felsing, J.M. \(2002\). A Practical Guide to Feature-Driven Development. Prentice Hall. \(ISBN 0-13-067615-2\)](#)
9. [Cockburn, Alistair. Crystal Clear, A Human-Powered Methodology for Small Teams \[Text\] / Alistair Cockburn // Addison-Wesley Professional, 2004. – pp.336. - ISBN 0-201-69947-8.](#)
10. [Kanban: Successful Evolutionary Change for Your Technology Business, David J. Anderson. \(United States, Blue Hole Press, 2010. ISBN 978-0984521401](#)
11. [Scrumban: Essays on Kanban Systems for Lean Software Development, Corey Ladas. \(United States, Modus Cooperandi Press, 2009. ISBN 9780578002149](#)
12. [Mary Poppendieck; Tom Poppendieck \(2003\). Lean Software Development: An Agile Toolkit. Addison-Wesley Professional. ISBN 978-0-321-15078-3.](#)
13. [Cohn, Mike. "Planning Poker Cards: Effective Agile Planning and Estimation". Mountain Goat Software, 30 March 2016. - режим доступу - <https://www.mountaingoatsoftware.com/tools/planning-poker>](#)
14. [Eric Evans, 2015 Domain Driven Design, Definitions and Pattern Summaries. Режим доступу - \[https://domainlanguage.com/wp-content/uploads/2016/05/DDD\\\_Reference\\\_2015-03.pdf\]\(https://domainlanguage.com/wp-content/uploads/2016/05/DDD\_Reference\_2015-03.pdf\)](#)
15. [Leybourn, E. \(2013\). Directing the Agile Organisation: A Lean Approach to Business Management. London: IT Governance Publishing: 71–79.](#)

Ознайомитись з розділами, що стосуються наступних тем дисципліни: Матеріали знаходяться у вільному доступі в Інтернеті.

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
Тема 1. Життєвий цикл розробки програмного забезпечення		
1	Лекція 1. Життєвий цикл розробки програмного забезпечення	Фази проекту згідно методології PMI та процесу розробки програмного забезпечення. Моделі життєвого циклу. Водоспадна модель. Модель прототипування. Модель великого вибуху. V-подібна. Ітерація. Інкрементальні та ітеративні моделі. Спіральна модель. СРС - п.6 №1

<i>Тема 2. Методології Agile</i>		
3	<i>Лекція 2. Принципи Agile</i>	<i>Основні методології розробки програмного забезпечення. Основні принципи Agile. CPC - п.6 №2</i>
4	<i>Лекція 3. Методології RAD &amp; DSDM</i>	<i>Застосування методології RAD. Основні фази DSDM та її застосування. CPC - п.6 №3</i>
	<i>Лекція 4. Методологія SCRUM (Частина 1)</i>	<i>Scrum: основні поняття та цикл розробки. CPC - п.6 №4</i>
5	<i>Лекція 5. Методологія SCRUM (Частина 2)</i>	<i>Журнал побажань проекту, історія користувача та зустрічі в Scrum. CPC - п.6 №5</i>
6	<i>Комп'ютерний практикум 1. SCRUM</i>	<i>Завдання: Завдання: запланувати проект та розробити три його ітерації використовуючи процес SCRUM. CPC - п.6 №6</i>
7	<i>Лекція 6. Екстремальне програмування</i>	<i>Основні поняття та практики XP. Цикл розробки по XP. CPC - п.6 №7</i>
8	<i>Лекція 7. Розробка за функціями</i>	<i>Основні поняття та практики Feature driven development. Цикл розробки по FDD. CPC - п.6 №8</i>
9	<i>Лекція 8. Методології Crystal та методології Lean</i>	<i>Основні аспекти методологій Crystal methodologies, їх різновиди та відмінності. Основні поняття та принципи Lean. CPC - п.6 №9</i>
10	<i>Лекція 9. Методології Kanban та Scrumban</i>	<i>Методології Kanban та Scrumban, основні відмінності та умови застосування. CPC - п.6 №10</i>
<i>Тема 3. Основні практики Agile</i>		
11	<i>Лекція 10. Журнал побажань спринта та проекту</i>	<i>Ітеративна та інкрементна розробка. Журнал побажань спринта та проекту. CPC - п.6 №11</i>
12	<i>Комп'ютерний практикум 2. TRELLO</i>	<i>Завдання: зареєструватись в системі TRELLO, створити кілька ітерацій та завдань для кожної ітерації, внести зміни в завдання та змінити статус завдання. CPC - п.6 №12</i>
13	<i>Лекція 11. Користувацькі історії та story points</i>	<i>Історії користувача та методи їх оцінювання. CPC - п.6 №13</i>

14	Комп'ютерний практикум 3. Користувацькі історії та їх планування	Завдання: розробити користувацькі історії та оцінити їх за допомогою покеру планування. СРС - п.6 №14
15	Лекція 12. Планування та продуктивність згідно Agile	Покер планування та вимірювання продуктивності в Agile розробці. СРС - п.6 №15
16	Лекція 13. Доменне моделювання	Предметно-орієнтоване проектування, його фази та особливості. СРС - п.6 №16
17	Лекція 14. Сюжетне моделювання та тестування в Agile	Сюжетне моделювання та тестування в Agile. СРС - п.6 №17
18	Лекція 15. Інші практики Agile	Неперервна інтеграція. Крос-функціональна команда. СРС - п.6 №18
<i>Модульна контрольна робота</i>		

6.

#### 6. Самостійна робота студента/аспіранта

Дисципліна «Розроблення програмного забезпечення за методологією Agile» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин	Література
1	Підготовка до лекції 1	2	3, стор. 3-28; 4, стор. 25-28
2	Підготовка до лекції 2	2	5
3	Підготовка до лекції 3	2	6, стор. 10-25; 6
4	Підготовка до лекції 4	2	7, стор. 1-25
5	Підготовка до лекції 5	2	7, стор. 1-25
6	Підготовка до комп'ютерного практикуму 1	2	7, стор. 1-25
7	Підготовка до лекції 6	2	8
8	Підготовка до лекції 7	2	9, стор. 5-13
9	Підготовка до лекції 8	2	10, стор. 127-132

10	Підготовка до лекції 9	2	11, стор. 13-20
11	Підготовка до лекції 10	2	12, стор. 67-75
12	Підготовка до комп'ютерного практикуму 2	2	11, стор. 13-20; 12, стор. 67-75
13	Підготовка до лекції 11	2	13, стор. 161-164
14	Підготовка до комп'ютерного практикуму 3	2	13, стор. 161-164; 12, стор. 67-75
15	Підготовка до лекції 12	2	14
16	Підготовка до лекції 13	2	15
17	Підготовка до лекції 14	2	15
18	Підготовка до лекції 15	2	16, стор. 71-79
19	Підготовка до модульної контрольної роботи	6	1, стор. 3-28; 2, стор. 25-28; 7, стор. 1-25; 9, стор. 5-13; 10, стор. 127-132; 11, стор. 13-20; 12, стор. 67-75; 14-15; 16, стор. 71-79
20	Підготовка до заліку	6	1, стор. 3-28; 2, стор. 25-28; 7, стор. 1-25; 9, стор. 5-13; 10, стор. 127-132; 11, стор. 13-20; 12, стор. 67-75; 14-15; 16, стор. 71-79
21	Основні прийоми Rapid Application Development.	2	5, стор. 10-25
22	Цикл розробки за екстремальним програмуванням. Ping-pong програмування	2	8
23	Дослідження умов застосування Crystal Orange	2	10, стор. 127-132
24	Створення та життєвий цикл Kanban дошки в Trello	4	11, стор. 13-20
25	Планування задач методом Planning Poker.	4	14, стор. 1-8



26	Story-boarding	4	15, стор. 71-79
----	----------------	---	-----------------

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

- Відвідування лекційних занять є обов'язковим.
- Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби захисту робіт комп'ютерного практикуму.
- Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.
- Дотримання політики академічної доброчесності.
- Правила захисту робіт комп'ютерного практикуму: роботи повинні бути зроблені згідно варіанту здобувача освіти, що визначається його номером у списку групи.
- Правила призначення заохочувальних та штрафних балів є наступними.  
Заохочувальні бали нараховуються за творчий підхід у виконанні робіт комп'ютерного практикуму (максимальна кількість балів за всі роботи – 2 бали).  
Штрафні бали нараховуються за:  
- плагіат (код програми не відповідає варіанту завдання, ідентичність коду програми серед різних робіт) у роботах комп'ютерного практикуму: -5 балів за кожну спробу.

### 8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Протягом семестру студенти виконують 3 комп'ютерних практикуми. Максимальна кількість балів за кожний комп'ютерний практикум: 16 балів.

Бали нараховуються за:

- якість виконання лабораторної роботи (комп'ютерного практикуму): 0-6 балів;
- відповідь під час захисту лабораторної роботи (комп'ютерного практикуму): 0-6 балів;
- своєчасне представлення роботи до захисту: 0-4 бали.

Критерії оцінювання якості виконання:

- 6 балів – робота виконана якісно, в повному обсязі;
- 4-5 балів – робота виконана якісно, в повному обсязі, але має недоліки;
- 1-3 бали – робота виконана в повному обсязі, але містить незначні помилки;
- 0 балів – робота виконана не в повному обсязі, або містить суттєві помилки.

Критерії оцінювання відповіді:

- 6 балів – відповідь повна, добре аргументована;
- 4-5 бали – в цілому відповідь вірна, але має недоліки або незначні помилки;
- 1-3 бали – у відповіді є суттєві помилки;
- 0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання своєчасності представлення роботи до захисту:

- 4 бали – робота представлена до захисту не пізніше вказаного терміну;
- 3 бали – робота представлена до захисту протягом 1 тижня після вказаного терміну;
- 2 бали – робота представлена до захисту протягом 2 тижнів після вказаного терміну;
- 2 бали – робота представлена до захисту протягом 3 тижнів після вказаного терміну;



0 балів – робота представлена до захисту протягом 4 і більше тижнів після вказаного терміну.

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:

16 балів × 3 лаб. робіт (комп. практи.) = 48 балів.

Завдання на **модульну контрольну роботу** складається з 6 питань, відповідь на 4 з яких оцінюється 8 балами, на 2 - 10 балами.

Критерії оцінювання складніших запитань контрольної роботи:

9-10 балів – відповідь вірна, повна, добре аргументована;

7-8 балів – відповідь вірна, розгорнута, але не дуже добре аргументована;

5-6 балів – в цілому відповідь вірна, але має недоліки;

3-4 балів – у відповіді є незначні помилки;

1-2 бали – у відповіді є суттєві помилки;

0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання простіших запитань контрольної роботи:

7-8 балів – відповідь вірна, повна, добре аргументована;

5-6 балів – в цілому відповідь вірна, але має недоліки;

3-4 балів – у відповіді є незначні помилки;

1-2 бали – у відповіді є суттєві помилки;

0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

10 балів × 2 запитання + 8 балів × 4 запитання = 52 бали.

Рейтингова шкала з дисципліни дорівнює:

$R = R_c = 48 \text{ балів} + 52 \text{ бали} = 100 \text{ балів}$ .

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 8 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 16 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: залік

Умови допуску до семестрового контролю:

При семестровому рейтингу ( $r_c$ ) не менше 60 балів та зарахуванні усіх робіт комп'ютерного практикуму, студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою). В іншому разі він має виконувати залікову контрольну роботу.

Необхідною умовою допуску до залікової контрольної роботи є виконання і захист комп'ютерного практикуму.

Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи, при цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

Склад та критерії оцінювання залікової контрольної роботи:

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
-----------------	--------

100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

## **9. Додаткова інформація з дисципліни (освітнього компонента)**

*Перелік питань, які виносяться на семестровий контроль, наведено у Додатку 1.*

### **Робочу програму навчальної дисципліни (силабус):**

**Складено** к.т.н., ст викл., Хіцко Я.В.

**Ухвалено** кафедрою ПЗКС (протокол №8 від 25.01.2023)

**Погоджено** Методичною комісією факультету прикладної математики (протокол №6 від 27.01.2023)

*Додаток 1. Перелік питань, які виносяться на семестровий контроль*

- 1. Назвіть основні практики в Feature Driven Design (мінімум 3).*
- 2. Як проходить щоденна нарада в Scrum і які питання обговорюються?*
- 3. Чи розповсюджуються проміжні версії продукту в каскадній та інкрементній моделях розробки програмного забезпечення?*
- 4. Проєкти якого масштабу можуть використовувати Crystal Orange?*
- 5. Якими є умови застосування методології RAD?*
- 6. Чи є закінчення спринту в Scrumban?*
- 7. Чи є закінчення спринту в Kanban?*
- 8. Чим відрізняється user story від use case?*
- 9. Як чином оцінюється продуктивність команди в Scrum?*
- 10. Що таке анти-патерн 'watch the master' в парному програмуванні?*
- 11. Як можуть змінюватися вимоги в каскадній та інкрементній моделях розробки ПЗ?*
- 12. Чи може архітектор або менеджер проєкту заздалегідь визначати оцінки задач в процесі покеру планування?*
- 13. Для чого використовуються story points?*
- 14. Що таке принцип MoSCoW?*