НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп’ютерних наук

«ЗАТВЕРДЖУЮ»
Декан факультету інформаційних технологій
_____ Олена ГЛАЗУНОВА
«___» _____ 20 23 р.

«СХВАЛЕНО»
на засіданні кафедри комп’ютерних наук
Протокол № 12 від «01» 06 20 23 р.
Завідувач кафедри
_____ Белла ГОЛУБ

«РОЗГЛЯНУТО»
Гарант ОП_«Інженерія програмного забезпечення»_
Гарант ОП
_____ Белла ГОЛУБ

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Спеціальність : 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

Факультет інформаційних технологій

Розробник: старший викладач кафедри комп’ютерних наук Бородкін Г.О.

Київ 2023

1 Опис навчальної дисципліни

Аналіз вимог до програмного забезпечення

Галузь знань, спеціальність, освітній ступінь		
Освітній ступінь	Бакалавр	
Спеціальність	121 «Інженерія програмного забезпечення»	
Освітня програма	121 «Інженерія програмного забезпечення»	
Характеристика навчальної дисципліни		
Вид	Обов’язкова	
Загальна кількість годин	120	
Кількість кредитів ECTS	4	
Кількість змістових модулів	2	
Курсовий проект (робота) (за наявності)	-	
Форма контролю	Іспит	
Показники навчальної дисципліни для денної та заочної форм навчання		
	денна форма навчання	заочна форма навчання
Курс (рік підготовки)	2	
Семестр	3	
Лекційні заняття	30 год	
Практичні, семінарські заняття		
Лабораторні заняття	30 год	
Самостійна робота	60 год	
Індивідуальні завдання		
Кількість тижневих аудиторних годин для денної форми навчання	4	

2 Мета та завдання навчальної дисципліни

Мета: Ознайомити студента з процесом аналізу і управління вимогами до програмних продуктів.

Набуття компетентностей:

інтегральна компетентність (ІК):

Здатність розв’язувати складні задачі і проблеми інженерії програмного забезпечення, що передбачає проведення досліджень з елементами наукової новизни та/або здійснення інновацій в умовах невизначеності вимог.

загальні компетентності (ЗК):

К05. Здатність вчитися і оволодівати сучасними знаннями.

К09. Прагнення до збереження навколишнього середовища.

К10. Здатність діяти соціально відповідально та свідомо.

фахові (спеціальні) компетентності (ФК):

К13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

К16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

К17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

К20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення.

К23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

К25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

Програмні результати навчання (ПРН):

ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПР10. Проводити передпроєктне обстеження предметної області, системний аналіз об'єкта проєктування в області природоохоронної галузі та сільського господарства.

ПР11. Вибирати вихідні дані для проєктування, керуючись формальними методами опису вимог та моделювання.

ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.

2 Програма та структура навчальної дисципліни для:

– повного терміну денної (заочної) форми навчання;

– скороченого терміну денної (заочної) форми навчання.

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
ЗМІСТОВНИЙ МОДУЛЬ 1. АНАЛІЗ ВИМОГ												
Тема 1 Вступ до дисципліни	8	2		2		5						
Тема 2. Поняття вимог. Класифікації вимог	8	2		2		5						
Тема 3. Властивості вимог	8	2		2		5						
Тема 4. Процес аналізу вимог	8	2		2		5						
Тема 5. Контекст завдання аналізу вимог	8	2		2		5						
Тема 6. Виявлення вимог	8	2		2		5						
Разом за змістовим модулем 1	**54**	**12**		**12**		**30**						
ЗМІСТОВНИЙ МОДУЛЬ 2. УПРАВЛІННЯ ВИМОГАМИ												
Тема 7. Формування бачення	8	2		2		3						
Тема 8. Класифікація і специфікування вимог	8	2		2		3						
Тема 9. Розширений аналіз вимог	8	2		2		4						
Тема 10. Ілюстровані сценарії і прототипи	8	2		2		3						
Тема 11. Документування вимог	8	2		2		3						
Тема 12. Перевірка вимог	8	2		2		4						
Тема 13. Управління вимогами	8	2		2		3						
Тема 14. Вдосконалення процесів роботи з вимогами	8	2		2		3						
Тема 15. Вимоги в управлінні проектом	8	2		2		4						
Разом за змістовим модулем 2	**66**	**18**		**18**		**30**						
Усього годин	**120**	**30**		**30**		**60**						
Курсовий проект (робота) з ***(якщо є в робочому навчальному плані)***		**-**	**-**	**-**		**-**		**-**	**-**	**-**		**-**
Усього годин	120	30		30		60						

5 ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин
1	Концептуальні вимоги	2
2.	Класифікації вимог	2
3.	Властивості вимог	2
4.	Процес аналізу вимог	2
5.	Контекст завдання аналізу вимог	2
6.	Виявлення вимог	2
7.	Формування бачення	2
8.	Класифікація і специфікування вимог	2
9.	Розширений аналіз вимог	2
10.	Ілюстровані сценарії і прототипи	2
11.	Документування вимог	2
12.	Перевірка вимог	2
13.	Управління вимогами	2
14.	Вдосконалення процесів роботи з вимогами	2
15.	Вимоги в управлінні проектом	2
	Разом	30

6 ТЕМИ САМОСТІЙНОЇ РОБОТИ

№ з/п	Назва теми	Кількість годин
1	Концептуальні вимоги	5
2.	Класифікації вимог	5
3.	Властивості вимог	5
4.	Процес аналізу вимог	5
5.	Контекст завдання аналізу вимог	5
6.	Виявлення вимог	5
7.	Формування бачення	3
8.	Класифікація і специфікування вимог	3
9.	Розширений аналіз вимог	4
10.	Ілюстровані сценарії і прототипи	3
11.	Документування вимог	3
12.	Перевірка вимог	4
13.	Управління вимогами	3
14.	Вдосконалення процесів роботи з вимогами	3
15.	Вимоги в управлінні проектом	4
	Разом	60

7 Зразки контрольних питань, тестів для визначення рівня засвоєння знань студентами.

Комплект тестів для визначення рівня засвоєння знань студентами

знаходиться за посиланням:

http://elearn.nubip.edu.ua/course/view.php?id=891

Зразки тестових питань
для визначення рівня засвоєння знань студентами

1. ***Бізнес модель продукту включає***
   1.1. Функції продукту, Прибуток, Відносини з користувачем
   1.2. Значення продукту, Відносини з користувачем, Користувач, Дохід
   1.3. Функції продукту, Дохід, Прибуток
2. ***Виберіть правильне твердження: High Level Requirements***
   2.1. Погоджують із зацікавленими особами, які розбираються в технічних деталях
   2.2. Не задають як ми будемо реалізовувати систему
   2.3. Задають як ми будемо реалізовувати систему

2.4. Погоджують із зацікавленими особами, які не розбираються в технічних деталях

3. ***High Level Requirements включає***
   3.1. Features, Needs
   3.2. Features, Function, Needs
   3.3. Needs, Function
4. ***Виберіть твердження, що підходять під визначення USECASE***
   4.1. Сценарій використання
   4.2. варіант використання
   4.3. прецедент використання
5. ***Виберіть правильне твердження***
   5.1. Прецеденти служать для документування функціональних вимог до програмних систем.
   5.2. Прецедент описує деякий цілісний фрагмент поведінки системи, не вдаючись при цьому особливо внутрішньої структури суб'єкта.
   5.3. Прецедент описує взаємодію програмної системи з діючими особами у вигляді послідовності повідомлень.
6. ***Виберіть правильне твердження про USECASE***
   6.1. опис поведінки системи, коли вона взаємодіє з кимось (або чимось) із зовнішнього середовища.
   6.2. різновид відносини залежності між базовим варіантом використання і його спеціальним випадком.
   6.3. визначає взаємозв'язок базового варіанту використання з іншим варіантом використання, функціональне поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.
   6.4. моделює відповідну спільність ролей.
7. ***Вимоги до ПЗ – це***
   7.1. Вимоги як користувачів (зовнішня поведінка системи), так і розробників (деякі приховані параметри).
   7.2. Умови або можливості, необхідні користувачеві для вирішення проблем і досягнення цілей
   7.3. Умови або можливості, якими повинна володіти система або системні компоненти, щоб виконати контракт або задовольняти стандартам, специфікаціям або іншим формальним документам
8. **Оберіть потрібне.** ***Яка з відповідей на питання: «Чи готові Ви придбати продукт?» зараховується як готовність користувача заплатити за ПЗ:***
   8.1. Я точно куплю
   8.2. Я майже впевнений, що я куплю
   8.3. Я сумніваюся, що я куплю
   8.4. Я точно не куплю
9. ***Пункти concept statement***

9.1. Для чого призначений, Як працює, Макет, Ціна
9.2. Як працює, Макет, Назва, Для чого призначений, Ціна
9.3. Як працює, Для чого призначений, Ціна, Назва
9.4. Назва, Як працює, Макет, Ціна значен, Ціна, Назва

10. Створення concept statement (вибрати правильний порядок)

10.1. Ідея, Споживачі і зацікавлені особи, Рішення, визначення проблеми
10.2. Ідея, визначення проблеми, Рішення
10.3. Ідея, визначення проблеми, Споживачі і зацікавлені особи
10.4. Ідея, визначення проблеми, Рішення, Споживачі і зацікавлені особи

11. Тестування концепції включає

11.1. Потреби користувача
11.2. Проблеми користувача
11.3. Властивості продукту
11.4. Макет продукту

12. Функціональні вимоги визначають

12.1. те як користувач взаємодіє з системою
12.2. ставлять як ми будемо реалізовувати систему ми будемо реалізовувати систему
12.3. не ставлять як ми будемо реалізовувати систему будемо реалізовувати систему

13. Що буде якщо ми не поставимо HLRs?

13.1. Багато доробок за результатами складання продукту (ПО, системи)
13.2. Замовник платить не за обсяг роботи, а за людино-години, витрачені командою підрядника на розробку і впровадження програмного забезпечення.
13.3. Не виходить визначити повний обсяг робіт або терміни їх виконання.

14. Вимоги до ПЗ складаються з трьох рівнів

14.1. Так
14.2. Ні

15. Оцінити дохід можна за допомогою

15.1. High Level Requirements
15.2. Concept statement
15.3. Функціональні вимоги

8 МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни використовуються такі методи навчання:

М1. Лекція (проблемна, інтерактивна).

М2. Лабораторна робота – для використання набутих знань до розв'язування практичних завдань.

М3. Проблемне навчання – створення проблемної ситуації для зацікавленого і активного сприйняття матеріалу.

М4. Проектне навчання (індивідуальне).

М8. Дослідницький метод.

9 Форми контролю

МК1. Тестування.

МК4. Методи усного контролю.

МК5. Екзамен.

МК7. Звіт

10 Розподіл балів, які отримують студенти. Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «Положення про екзамени та заліки у НУБіП України» (наказ про уведення в дію від 26 квітня 2023 р. протокол № 10)

Шкала оцінювання

Рейтинг студента, бали	Оцінка національна за результати складання	
	екзаменів	заліків
90-100	відмінно	зараховано
74-89	добре	
60-73	задовільно	
0-59	незадовільно	не зараховано

Для визначення рейтингу студента із засвоєння дисципліни $\mathbf{R_{ДИС}}$ (до 100 балів) одержаний рейтинг з атестації $\mathbf{R_{АТ}}$ (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $\mathbf{R_{НР}}$ (до 70 балів): $\mathbf{R_{ДИС} = R_{НР} + R_{АТ}}$.

11 Навчально-методичне забезпечення

1. Електронний навчальний курс «Аналіз вимог до програмного забезпечення» http://elearn.nubip.edu.ua/course/view.php?id=891

12. Рекомендовані джерела інформації

Основна

1. Лавріщева К.М. Програмна інженерія. Підручник –К., «НАНУ» – 2008.–319 с

2. I. Alexander and L. Beus-Duekic, Discovering Requirements: How to Specify Products and Services, Wiley, 2009.
3. K.E. Wiegers, Software Requirements, 2th ed., Microsoft Press, 2003.
4. INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 3.2.2, International Council on Systems Engineering, 2012.
5. A. van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley, 2009.
6. Бабенко Л.П., Лавріщева К.М. Основи програмної інженерії.– Навч. посібник.–К.: Знання, 2001. –269 с.
7. Лавріщева К.М. Основні напрямки досліджень в програмній інженерії і шляхи їхнього розвитку // Проблеми програмування. – 2003. – № 3–4. – С. 44–58.
8. I. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2011.
9. S. Friedenthal, A. Morre, and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, 2nd ed., Morgan Kaufmann, 2012.
10. Guckkenheimer S., Peter J. Software Engineering With Microsoft Visual Studio. Team System. – Adison Wesley, 2006. – 273 p.
11. D. Budgen, Software Design, 2nd ed., Addison-Wesley, 2003.
12. I. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2011.
13. R.S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill, 2010.

Допоміжна

14. O. Gotel and C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem", *Proc 1st Int'l Conf. Requirements Eng.*, IEEE, 1994.
15. N.A. Maiden and C. Ncube, "Acquiring COTS Software Selection Requirements", *IEEE Software*, vol. 15, no. 2, Mar.-Apr. 1998, pp. 46-56.
16. C. Potts, K. Takahashi, and A.I. Antón, "Inquiry-Based Requirements Analysis", IEEE Software, International Council on Systems Engineering, vol. 11, no. 2, Mar. 1994, pp. 21-32.
17. Zave P., Jackson M. Four Dark Corners of Requirements Engineering // ACM Transactions on Software Engineering, January 1997.– № 1.
18. Pinheiro Francisco A. C., Goguen Joseph A.. An Object-Oriented tool for Tracing Requirements // Software.– Mach 1996.– № 3.
19. I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process, Addison-Wesley Professional, 1999.
20. J. Nielsen, Usability Engineering, Morgan Kaufmann, 1993.
21. G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.

13 Інформаційні ресурси та стандарти

1. ЕНК із дисципліни знаходиться за електронною адресою: **https://elearn.nubip.edu.ua/enrol/index.php?id=891**
2. ISO/IEC/IEEE., 24765:2010 Systems and Software Engineering—Vocabulary, ISO/IEC/IEEE, 2010.
3. IEEE, IEEE Std. 1069-2009 Standard for Information Technology—Systems Design—Software Design Descriptions, IEEE, 2009.
4. IEEE Std., 12207-2008 (a.k.a. ISO/IEC 12207:2008) 'Standard for Systems and Software Engineering—Software Life Cycle Processes, IEEE, 2008.
5. IEEE Guide to the Software Engineering Body of Knowledge. (1) - SWEBOK®, 2004
6. IEEE Guide to the Software Engineering Body of Knowledge. - SWEBOK®, 2015
7. ISO 9000 Системи керування якістю — Основні положення і словник.
8. ISO 9000-1 Стандарти з керування якістю та забезпечення якості. Ч.1. Настанови щодо вибору та застосування.
9. ISO 9000-2 Стандарти з керування якістю та забезпечення якості. Ч.2. Настанови щодо застосування ISO 9001, ISO 9002, ISO 9003.
10. ISO 9000-3 Стандарти з керування якістю та забезпечення якості – Ч.3. Настанови щодо застосування ISO 9001 під час розроблення, постачання та супроводження програмного забезпечення.
11. ISO 9000-4 Стандарти з керування якістю та забезпечення якості – Ч.4. Настанови щодо керування програмою надійності.
12. ISO 9001-2001 Системи керування якістю. Вимоги.
13. ISO 9003 Система якості. Модель забезпечення якості в процесі контролю готової продукції та її випробуваннях.
14. ISO/IEC 9126-1:2001 Програмна інженерія. Якість продукту.Ч.1. Модель якості.
15. ISO/IEC TR 9126-2:2003 Програмна інженерія. Якість продукту.Ч.2. Зовнішні метрики
16. ISO/IEC TR 9126-3:2003 Програмна інженерія. Якість продукту.Ч.3. Внутрішні метрики.
17. ISO/IEC TR 9126-4:2004 Програмна інженерія. Якість продукту.Ч.4. Метрики якості при використанні.
18. ISO/IEC 14598-2:2000 Програмна інженерія. Оцінювання програмного продукту. Ч.2. Планування та керування.
19. ISO/IEC 14598-3:2000 Програмна інженерія. Оцінювання програмного продукту. Ч.3. Процес для розробників.
20. ISO/IEC 14598-4:1999 Програмна інженерія. Оцінювання продукту. Ч.4. Процес для замовників.
21. ISO/IEC 14598-6:2001 Програмна інженерія. – Оцінка продукту – Ч.6. Документація модулів оцінювання.

22. ISO/IEC 19761:2003 Програмна інженерія. COSMIC-FFP. Метод
23. ISO/IEC 25000:2005 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(SQuaRE). Настанова щодо оцінювання і вимоги до якості програмного продукту.
24. ISO/IEC 25001:2007 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(SQuaRE). Планування та керування.
25. ISO/IEC 25020:2007 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(SQuaRE). Настанова та еталонна модель вимірювання.
26. ISO/IEC TR 25021:2007 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(ОВЯПП - SQuaRE). Елементи вимірювання якості.
27. ISO/IEC 25030:2007 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(SQuaRE). Вимоги якості.
28. ISO/IEC 25051:2006 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(SQuaRE). Вимоги до якості готових комерційних продуктів програмних засобів та інструкції по тестуванню.
29. ISO/IEC 25062:2006 Програмна інженерія. Оцінювання і вимоги до якості програмного продукту(SQuaRE). Загальний промисловий формат, який використовується при тестуванні звітів.
30. ДСТУ ISO/IEC TR 9126-2 Програмна інженерія. Якість продукту.Ч.2 Зовнішні метрики.
31. ДСТУ ISO/IEC TR 9126-3 Програмна інженерія. Якість продукту.Ч.3 Внутрішні метрики.
32. ДСТУ ISO/IEC TR 9126-4 Програмна інженерія. Якість продукту.Ч.4 Метрики якості при використанні.
33. ДСТУ ISO/IEC 15939 Інженерія систем і програмних засобів. Процес вимірювання.