

Харківський національний університет ім.В.Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та практичної системотехніки

УХВАЛЕНО

Вченою радою факультету
комп'ютерних наук, протокол № 4
від «03» грудня 2019 р.
Голова Вченої ради _____



Лазурик В.Т. Лазурик В.Т.

Назва курсу	Методи і технології інженерії програмного забезпечення
Викладач (-і)	професор Толстолузька Олена Геннадіївна, доцент Бакуменко Ніна Станіславовна
Профайл викладача (-ів)	http://www-csd.univer.kharkov.ua/about-us/sub-faculty/tase-department/personalnij-sklad/
Контактний тел.	Кафедральний 707-50-22
E-mail:	elena.tolstoluzka@karazin.ua
Сторінка курсу в системі дистанційного навчання	
Консультації	Очні консультації: розклад в університеті (на кафедрі) Онлайн-консультації: через e-mail

ЗМІСТ

1.	Коротка анотація до курсу	3
2.	Мета та цілі курсу	3
3.	Формат курсу	3
4.	Результати навчання	3
5.	Обсяг курсу	4
6.	Ознаки курсу	4
7.	Пререквізити	4
8.	Технічне та програмне забезпечення /обладнання	4
9.	Політики курсу	4
10.	Схема курсу	5
11.	Система оцінювання та вимоги	11
12.	Рекомендована література	12

1. Коротка анотація до курсу – курс «Методи і технології інженерії програмного забезпечення» спрямована на опанування студентами сучасних інформаційних технологій для розробки програмного забезпечення, аналізу предметної області, формулювання вимог, проектування, верифікації та тестування програмного забезпечення.

2. Мета та цілі курсу – засвоєння студентами основних термінів та понять, які використовуються в сучасних методах і технологіях інженерії програмного забезпечення; моделей життєвого циклу програмного забезпечення; методики оцінки вартості програмного забезпечення.

3. Формат курсу - очний

4. Результати навчання –

знати:

- умови виникнення інженерії програмного забезпечення та її місце в контексті інших інженерій;
- основні терміни та поняття, які використовуються в сучасних методах і технологіях інженерії програмного забезпечення;
- культуру інженерії програмного забезпечення і моделі зрілості;
- складові життєвого циклу програмного забезпечення;
- складові інженерії програмного забезпечення; їх застосування і зв'язок.

уміти:

- формувати підходи до свідомого застосування моделей життєвого циклу програмного забезпечення;
- практично використовувати методи інженерії програмного забезпечення;
- отримувати оцінки вартості програмного забезпечення.

5. Обсяг курсу

Вид заняття	Загальна кількість годин
лекції	32
практичні	32
самостійна робота	56

6. Ознаки курсу:

Рік викладання	семестр	спеціальність	Курс (рік навчання)	Нормативний\ вибірковий
----------------	---------	---------------	------------------------	----------------------------

2019	2	Комп'ютерна інженерія	3	нормативний (Н)
------	---	-----------------------	---	-----------------

7. Пререквізити – перед вивченням курсу студент повинен знати вищу математику, алгоритмізація та програмування, прикладне програмне забезпечення, об'єктно-орієнтоване програмування, системне програмування.

8. Технічне та програмне забезпечення /обладнання - лабораторні роботи проводяться у спеціалізованому комп'ютерному класі на 15 робочих місць, який оснащений сучасними комп'ютерами з ліцензованою операційною системою Windows та програмним забезпеченням (мова програмування C++).

9. Політики курсу

Академічна доброчесність - це сукупність етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання та/або наукових (творчих) досягнень. Дотримання академічної доброчесності здобувачами освіти передбачає самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання, а саме виконання лабораторних робіт за своїм варіантом. Звіти з лабораторних робіт виконуються кожним студентом і захищаються індивідуально.

Тиж. / акад.год.	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, ПЗ,ЛР, СРС) / Формат** **аудиторна, СРС	Матеріали	Завдання, год
Розділ 1. Інженерія програмного забезпечення. Основні поняття				
Тиж. 1 9 год.	Тема 1. Вступ. Умови виникнення інженерії програмного забезпечення. Лекція 1. Введення понять інженерії програмного забезпечення. Визначення принципів діяльності в межах інженерної галузі.	Лекція / аудиторна	Презентація лекції	2 год

	СРС. Аналіз умов виникнення кризи в програмному забезпеченні в сімдесяті роки минулого століття.	<i>CPC / CPC</i>	Конспект лекції, <u>М.О.Сидоров Вступ до інженерії програмного забезпечення.-К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. -112 с.</u>	опрацювати лекцію, ознайомитись з літературою, 5 год
	ПЗ1. Умови виникнення інженерії програмного забезпечення	Практичне заняття.	Конспект лекції, <u>М.О.Сидоров Вступ до інженерії програмного забезпечення.-К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. - 112 с. (С. 3-15)</u>	опрацювати лекцію, ознайомитись з літературою, скласти доповідь 2 год.
Тиж. 2 9 год.	Лекція 2. Визначення принципів діяльності в межах інженерної галузі.	<i>Лекція / аудиторна</i>	Презентація лекції	2 год
	СРС. Аналіз фаз розвитку інженерної галузі.	<i>CPC</i>	Конспект лекції, Striuk A. M. Software engineering: first 50 years of formation and development // Computer Science & Software Engineering : Proceedings of the 1st Student Workshop (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018 / Edited by : Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Solo-viev, Andrii M. Striuk. – P. 11-36. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2292). – режим доступу : http://ceur-ws.org/Vol-2292/paper01.pdf	опрацювати лекцію, ознайомитись з літературою, 5 год.

	ПЗ 2. Програмна інженерія як наукова дисципліна.	Практичне заняття.	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год
Тижд. 3 9 год	<i>Тема 2.</i> Інженерія програмного забезпечення - інженерна галузь. Основні аспекти інженерії програмного забезпечення.	<i>Лекція / аудиторна</i>	Презентація лекції	2 год
	СРС. Аналіз фаз розвитку інженерної галузі.	<i>СРС</i>	Конспект лекції, Striuk A. M. Software engineering: first 50 years of formation and development // Computer Science & Software Engineering : Proceedings of the 1st Student Workshop (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018 / Edited by : Arnold E. Kiv, Serhiy O. Semerikov, Vladimir N. Solo-viev, Andrii M. Striuk. – P. 11-36. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2292). – режим доступу : http://ceur-ws.org/Vol-2292/paper01.pdf	опрацювати лекцію, ознайомитись з літературою, 5 год.
	ПЗ 3. Програмна інженерія як інженерна дисципліна.	Практичне заняття.	Конспект лекції, самостійно підібрана література	опрацювати лекцію, ознайомитись з літературою, скласти доповідь

				2 год
Тиж. 4 0 год	Тема 3. Культурний аспект інженерії програмного забезпечення. Лекція 4. Культура інженерії програмного забезпечення.	Лекція / аудиторна	Презентація лекції	2 год
	СРС. Чинники культури інженерії програмного забезпечення.	СРС	Конспект лекції	Опрацювати лекцію, ознайомитись з літературою, 5 год..
	ПЗ 4. Культурний аспект інженерії програмного забезпечення.	Практичне заняття.	Конспект лекції	Опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 5 9 год	Лекція 5. Моделі зрілості процесів, що відбуваються на підприємстві.	Лекція / аудиторна	Презентація лекції	2 год.
	СРС. Моделі культур організацій Константіноса та Де Грака.	СРС	Конспект лекції, <u>М.О.Сидоров Вступ до інженерії програмного забезпечення.-К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2010. - 112 с. (С. 15-20)</u>	опрацювати лекцію, ознайомитись з літературою, 5 год..
	ПЗ 5. Продукти інженерії програмного забезпечення, продукція і програмне забезпечення.	Практичне заняття.	Конспект лекції	опрацювати лекцію, ознайомитись з літературою,

				2 год.
Тиж. 6 9 год	Тема 4. Продукти інженерії програмного забезпечення, продукція і програмне забезпечення. Лекція 6. Системи програмного забезпечення.	Лекція / аудиторна	Презентація лекції	2 год.
	СРС. Моделі зрілості можливостей.	СРС	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 5 год.
	ПЗ 6. Якість продукту і якість процесу.	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Розділ 2. Моделі життєвого циклу програмного забезпечення. процеси, продукти, ресурси				
Тиж. 7 9 год	Тема 5. Пряма та обернена інженерія. Лекція 7. Інструменти прямої інженерії.	Лекція / аудиторна	Презентація лекції	2 год
	СРС. Стандарти, які визначають поняття продукт інженерії програмного забезпечення.	СРС	Конспект лекції, ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes, режим доступу: https://www.iso.org/standard/43447.html	опрацювати лекцію, ознайомитись із літературою, 5 год.

	ПЗ 7. Пряма інженерія. Інструменти прямої інженерії.	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 8 9 год	Лекція 8. Оборнена інженерія	<i>Лекція / аудиторна</i>	Презентація лекції	2 год.
	СРС. Стандарти, які визначають поняття продукт інженерії програмного забезпечення.	<i>CPC</i>	Конспект лекції, ISO/IEC/IEEE 16326:2009 Systems and software engineering – Project management режим доступу: https://www.iso.org/standard/41977.html	опрацювати лекцію, ознайомитись із літературою, 5 год.
	ПЗ 8. Оборнена інженерія. Методи оборненої програмної інженерії.	Практичне заняття.	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 9 9 год	<i>Тема 6.</i> Емпірична інженерія програмного забезпечення. Лекція 9. Методи емпіричної інженерії програмного забезпечення.	<i>Лекція / аудиторна</i>	Презентація лекції	2 год
	СРС. Системний підхід для визначення програмної системи.	<i>CPC</i>	Конспект лекції	опрацювати лекцію, ознайомитись із літературою, 5 год

	ПЗ 9. Показники та критерії оцінки систем.	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год..
Тиж. 10 9 год	Лекція 10. Засоби емпіричної інженерії програмного забезпечення.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Супровід системи. Особливості створення нової системи.	<i>СРС</i>	Конспект лекції	опрацювати лекцію, ознайомитись із літературою, 5 год.
	ПЗ 10. Вибір моделі життєвого циклу	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Розділ 3. Моделі, методи і засоби оцінювання вартості програмного забезпечення				
Тиж. 11 10 год.	<i>Тема 7.</i> Методи і моделі оцінювання вартості програмного забезпечення. Лекція 11. Одиниці розміру програмного забезпечення.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Метод функціональної	<i>СРС</i>	Конспект лекції	опрацювати

	декомпозиції.			лекцію, 6 год.
	ПЗ 11. Робочі процеси створення програмного забезпечення.	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
	Лекція 12. Інженерія вимог до програмних систем.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
Тиж. 12 10 год.	СРС. Об'єктно-орієнтована інженерія вимог	<i>СРС</i>	Конспект лекції	опрацювати лекцію, 6 год.
	ПЗ 12. Case-засоби проектування програмного забезпечення	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 13 10 год.	Лекція 13. Стандартна оцінка показників якості програмного забезпечення	<i>Лекція / аудиторна</i>	Конспект лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Керування якістю програмних систем.	<i>СРС</i>	Презентація лекції	опрацювати лекцію, 6 год.

	ПЗ 13. Оцінювання показників якості програмного забезпечення.	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 14 10 год.	Лекція 14. Керування інженерією програмного забезпечення.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Засоби емпіричної інженерії програмного забезпечення..	<i>СРС</i>	Конспект лекції	опрацювати лекцію, 6 год.
	ПЗ 14. Керування інженерією програмного забезпечення.	Практичне заняття	Конспект лекції	опрацювати лекцію, ознайомитись з літературою, 2 год.
Тиж. 15 10 год	Лекція 15. Неалгоритмічні методи і алгоритмічні моделі.	<i>Лекція / аудиторна</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Методи і моделі оцінювання вартості програмного забезпечення	<i>СРС</i>	Конспект лекції	опрацювати лекцію, 6 год.
	ПЗ 22. Критерій максиміна. Критерій Гурвіца.	Практичне заняття	Конспект лекції	опрацювати лекцію,

				ознайомитись з літературою, підготувати звіт, 2 год.
Тиж. 16 10 год.	Тема 8. Засоби оцінювання вартості програмного забезпечення. Лекція 16. Модель COCOMO. Модель Путнема (SLIM).	Лекція <i>Аудит.</i>	Презентація лекції	опрацювати лекцію, переглянути презентацію, 2 год
	СРС. Засоби оцінювання ПЗ, засновані на моделях SLIM і COCOMO.	СРС	Конспект лекції	опрацювати лекцію, 6 год.
	Контрольна робота	Контр. робота / аудиторна	Варіанти завдань КР	Повторити матеріал тем 1-8

***якщо література подається в скороченому вигляді, то розшифрування подаєте вкінці

11. Система оцінювання та вимоги

Загальна система оцінювання курсу.

Поточний контроль, самостійна робота, практичні та лабораторні заняття					залікова робота	Сума
Розділ 1	Розділ 2	Розділ 3	Контрольна робота, передбачена навчальним планом	Разом		
15	15	10	20	60	40	100

Загальна система	участь в роботі впродовж семестру/екзамен - 60/40
------------------	---

оцінювання курсу	Розподіл балів, що присвоюються студентам з навчальної дисципліни, є сумою балів за виконання всіх видів завдань та самостійну роботу плюс бали, отримані під час екзамену. Впродовж семестру студент за виконання всіх завдань може отримати до 60 балів і до 40 балів під час складання екзамену.
Вимоги до контрольної роботи	Завдання з контрольної роботи повинні бути виконані в повному обсязі. При оцінюванні роботи враховується повнота виконання, здатність студента формулювати закони та закономірності, структурувати судження, обґрунтовувати виконані операції, алгоритми, правила визначення понять, робити висновки, описи, тощо. Максимальна оцінка за контрольну роботу – 20 балів. Без виконаних контрольної роботи студент не отримує екзаменаційної оцінки.
Семінарські заняття	
Умови допуску до підсумкового контролю	Умовою допуску студента до підсумкового семестрового контролю (екзамену) є наявність прийнятих і зарахованих звітів з практичних завдань і наявність мінімальної кількості балів (не менше ніж 20) з поточного тестування та контролю. Згідно рішення кафедри теоретичної та прикладної системотехніки факультету комп'ютерних наук до екзамену не допускаються студенти, які мають заборгованість по практичним або контрольним роботам. В кінці курсу всі бали підсумовуються і студент має можливість (в разі успішного виконання всіх завдань) отримати максимальну підсумкову оцінку 100 балів.

Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для екзамену	для заліку
90 – 100	відмінно	зараховано
70-89	добре	

50-69	задовільно	
1-49	не задовільно	не зараховано

12. Рекомендована література

Основна література

1. Kane С. Тестирование программного обеспечения / С. Капе. - М: DiaSoft, 2001. - 542с.
2. Саммервил И. Инженерия программного обеспечения / И. Саммервил. - М.: Вильяме, 2002. - 720 с.
3. Сидоров Н.А. Повторное использование, переработка и восстановление программного обеспечения / Н.А. Сидоров // Управляющие системы и машины. - 2000. - № 3, 4. - С, 27 - 37.
4. Van Veendabl E. Standard glossary of term used in Software testing / E. Van Veendabl. - ISTQB. - 2007. Vol. 1,2. - June. - 30 p.

Допоміжна література

5. Bosch J. Design and use of software architectures / J. Bosch. -Addison Wesley, 2000. - 325 p.
6. Black R. Critical Testing Processes / R. Black. - Addison-Wesley, 2003.
7. Blum B.A. Taxonomy of Software Development Methods / B.A. Blum // Coinmunication of the ACM. - 1994. - Vol. 37, n. 11,-P. 82 - 94.
8. Budgen D. Software design: Reading / D. Budgen. - Addison-Wesley, 1994.-320 p.
9. Glass R.L. Extrime programming the good, the bad. and the bottom line / R.L. Glass // IEEE. Software. - 2001, - Vol.18, n.7. - P.11 -P. 112,- 111 - 112.
10. Georgiadou E. Software Process and Product Improvement: A Historical Perspective / E. Georgiadou // Кибернетика и системный анализ. - 2003. № 1, - P. 147 - 177,
11. Jacobson I. The Unified Software Development Process / I. Jacobson, G. Booch, J. Rumbaugh. - Addison-Wesley, 1999. - 310 p.
12. Jonsson P. Software Reuse. Architecture. Process and Organization for Business Success. Person Education Asia / P. Jonsson. - 2002. — 497 p.
13. Martin J. Rapid Application Development / J. Martin. -Macmillan, 1991. -250 p.

14. Perry D. Models of Software Development Environments / D. Perry, G. Kaiser // IEEE Trans. On Soft. Engin. - 1991. - Vol. 17, n. 3.-P. 283-295.
15. Railich V. Software cultures and evolution / V. Railich, N. Wilde // Computer. 2001. - Sept. - P.25 - 28.
16. Rombach H.D, Software specifications: a framework / H. D. Rombach. - Carnegie mullon Univ: SET, 1990. - 30 p.
17. Rajlich W. A stage Model for the Software Life Cycle / W. Rajlich, K. Bennett // Computer. - 2000. - July. - P. 77 - 70.
18. Spillner A. Software Testing Foundations / A. Spillner, T. Linz, H. Schafer. - Dpunkt: Verlag, 2007. - 277 p.
19. Sidorov N.A. Software Stylistics / N. A. Sidorov // Proceedings of NAU. - 2005. - 2(24), - P. 98 - 103.
20. Toriik Matsumoto K. An Environment for Computer - Aided Empirical Software Engineering / Matsumoto K. Gingerz Toriik // IEEE Trans, On Software Eng. - 1999. - Vol. 25, №. 4 - P. 474 - 485 .
21. Creting a software engineering culture / K. Wiegers // Dorset House Publishing. - New York, 2003. - 358 p.

13. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. TOP500 Supercomputing sites : Project ranks and details the 500 most powerful computer systems in the world [Electronic resource]. Available from : <http://www.top500.org>.
2. <http://www.cs.wisc.edu/condor/>
3. <http://setiathome.ssl.berkeley.edu/>
4. <http://www.Distributed.net/>
5. <http://mersenne.org/>
6. <http://www.globus.org/>
7. <http://www.eu-datagrid.org/>
8. <http://cyb.univ.kiev.ua/library/books/lavrishcheva-6.pdf>
9. <https://www.you-books.com/book/M-O-Sidorov/Vstup-do-inzhenerii-programnogo-zabezpechennya>
10. <http://ceur-ws.org/Vol-2292/paper01.pdf>