

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Кафедра інформаційних технологій

Стрелковська І.В., Приходько С.Б., Григор'єва Т.І.

УПРАВЛІННЯ ПРОЕКТАМИ І ЯКІСТЮ
ПРОГРАМНИХ СИСТЕМ

Методичні рекомендації для самостійної роботи здобувачів

Одеса 2023

Затверджено Вченою Радою Міжнародного гуманітарного університету
(протокол № 1 від 4 вересня 2023 р.)

Стрелковська І.В., Приходько С.Б., Григор'єва Т.І.

Управління проектами і якістю програмних систем: методичні рекомендації для самостійної роботи здобувачів [Електронне видання]. / Стрелковська І.В., Приходько С.Б., Григор'єва Т.І. Кафедра інформаційних технологій Міжнародного гуманітарного університету. Одеса, 2023. – 26 с.

Методичні рекомендації з курсу «Управління проектами і якістю програмних систем» розроблено відповідно до навчального плану, вони складаються з навчальної програми курсу, методичних рекомендацій з проведення практичних занять і завдань для самостійної роботи здобувачів, списку рекомендованої літератури. Матеріали призначено для студентів факультету кібербезпеки, програмної інженерії та комп'ютерних наук Міжнародного гуманітарного університету, які в магістратурі вивчають галузь знань - «Інформаційні технології». Вивчення дисципліни «Управління проектами і якістю програмних систем» сприятиме залученню здобувачів до науково-дослідницької діяльності та підготовки ними публікацій, кваліфікаційних й інших наукових робіт.

Вивчення дисципліни "Управління проектами і якістю програмних систем" допоможе студентам зрозуміти як обирати методологію розробки програмного забезпечення, що найкраще відповідає потребам конкретного проекту, і як впроваджувати її в дію. Студенти дізнаються про методи тестування, метрики якості та принципи підвищення якості програмного забезпечення. Студенти зможуть ознайомитися з ключовими принципами управління проектами, такими як визначення цілей проекту, розробка графіків, управління ресурсами та командою, ризиками та звітністю. Вони зможуть навчитися створювати плани проекту, визначати завдання та виконувати їх.

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 5, загальна кількість годин – 150	Галузь 12 – Інформаційні технології Спеціальність – 121 – Інженерія програмного забезпечення	обов'язкова	
		Рік підготовки:	
		1-й	
		Семестр	
		1-й	1-й
Мова навчання – українська	Рівень вищої освіти – другий (магістерський) рівень	Лекції	
		42 год.	4 год.
		Практичні, семінарські	
		28 год.	4 год.
		Лабораторні	
		Самостійна робота та індивідуальні завдання	
		80 год.	112 год.
		Вид контролю:	
		залік	залік

Вивчення дисципліни **«Управління проектами і якістю програмних систем»** дозволяє сформувати у здобувачів освіти компетенції, необхідні для ефективної діяльності в ролях менеджера програмного проекту, члена команди або співробітника проектного офісу.

Мета вивчення навчальної дисципліни **«Управління проектами і якістю програмних систем»**: вивчення методів та інструментів області управління проектами, а також отримання компетентностей, необхідних для визначення та успішного досягнення цілей проектів з розробки програмного забезпечення шляхом керування обсягом робіт, ресурсами, часом, якістю, ризиками та змінами.

Заплановані результати навчання за навчальною дисципліною

Знати:

1. Принципи та методи сталого дотримання балансу між стратегічними цілями організації – розробника програмного забезпечення, цілями окремих програмних проєктів і цілями вдосконалення процесу розроблення та розвитку фахівців, прийнятного для всіх зацікавлених сторін організації.
2. Універсальні та спеціальні моделі життєвого циклу програмного проєкту й портфеля проєктів, рекомендовані актуальними міжнародними й національними де-юре й де-факто стандартами з проєктного менеджменту.
3. Базові критерії ефективності програмного проєкту та методи їх обґрунтованого оцінювання й ефективного управління перебігом проєкту на підставі отриманих оцінок.
4. Моделі та методи раціональної організаційної структури програмного проєкту та керування нею.
5. Підходи до побудови результативної системи управління програмними проєктами.

Вміти:

1. Вибирати технологію управління програмним проєктом, найбільш відповідну його меті за початкових умов, налаштовувати її згідно з поточним перебігом проєкту та керувати ним із вчасним виявленням й опрацюванням невизначеностей, ризиків і проблем за допомогою поширених інструментів управління проєктами, зокрема MS Project, Bitrix, Trello.
2. Ефективно координувати різноманітні програмні проєкти за допомогою системи управління проєктами для балансування мети проєкту, стратегічних цілей організації-розробника та цілей удосконалення процесу розроблення програмних продуктів й розвитку учасників проєкту.

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Управління проєктами з розробки програмних систем.

Вступ. Основні види діяльності з управління програмним забезпеченням: Управління процесом розробки програмного забезпечення; Створення плану розробки програмного забезпечення (SDP); Розробка звіту про дані ресурсів програмного забезпечення (SDRP); Вибір підходу до розробки програмного забезпечення; Управління програмними ризиками; Прийняття рішення щодо використання програмного забезпечення з відкритим кодом.

Інші завдання з управління програмним забезпеченням: Вибір програмних метрик; Інтеграція моделі зрілості можливостей (CMMI); Планування

програмного забезпечення в плані системної інженерії (SEP); Розробка програмного забезпечення; Оцінка програмного ризику; Оцінка розміру програмного забезпечення; Розгляд процесу оцінки програмного забезпечення; Оцінка можливості повторного використання програмного забезпечення; Рекомендації щодо програмного забезпечення; Рекомендації щодо використання державного програмного забезпечення.

Виконання завдань з розробки системної безпеки.

Особливості програмного забезпечення для управління IT-проектами. Планування та складання графіків проекту. Командна співпраця. Управління ресурсами. Документаційне забезпечення управління. Звітність.

Тема 2. Управління процесом розробки програмного забезпечення.

Переваги добре продуманого і добре реалізованого процесу розробки програмного забезпечення: гарантія якості, послідовність і повторюваність, співпраця та координація, управління ризиками, масштабованість і ефективність, постійне вдосконалення.

Етапи процесу розробки програмного забезпечення: Планування (Planning), Реалізація (Implementing), Тестування (Testing), Розгортання та обслуговування (Deployment and Maintenance).

План розробки програмного забезпечення (SDP). Підходи до розробки програмного забезпечення (Водоспадний, Інкрементальний, Спіральний, Agile та Scrum).

Метрики процесу розробки програмного забезпечення, які визначають в Плані розробки програмного забезпечення (SDP). Program Management Offices (PMO).

Мета плану розробки програмного забезпечення (SDP)

Створення плану розробки програмного забезпечення (SDP).

Розробка звіту про дані ресурсів програмного забезпечення (SDRP).

Тема 3. Програмні метрики як невід'ємна частина стандартних процесів розробки програмного забезпечення.

Програмні метрики, які мають бути визначені в Плані розробки програмного забезпечення (SDP).

Розмір програмного забезпечення. Зусилля на розробку ПЗ. Графік розробки ПЗ. Дефекти програмного забезпечення. Визначення вимог до ПЗ та стабільність. Штат розробників ПЗ. Розвиток програмного забезпечення (дизайн, кодування та тестування). Використання обчислювальних ресурсів.

Оцінювання розміру ПЗ. Оцінювання зусиль на розробку ПЗ. Оцінювання часу розробки ПЗ. Оцінювання кількості дефектів. Оцінювання програмного

ризик.

Тема 4. Управління якістю програмних систем

Забезпечення якості, контроль якості та тестування як основи управління якістю програмного забезпечення. Визначення якості як ступіні відповідності явним або неявним вимогам і очікуванням. Визначення термінів: Quality Assurance, Quality Control та Testing. Два основних рівня якості програмного забезпечення. Функціональна якість. Нефункціональна якість. Структурна якість програмного забезпечення та її забезпечення за допомогою перевірки коду, аналізу та рефакторингу. Забезпечення функціональної якості за допомогою контролю якості та тестування.

Основні принципи тестування програмного забезпечення. Роль тестування в життєвому циклі розробки програмного забезпечення. Методи тестування програмного забезпечення. Види тестування програмного забезпечення.

Оцінювання якості ПЗ. Програмні метрики, що використовуються для оцінювання якості програмних систем.

3. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	денна форма				Заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекц.	прак	сам. роб.		лекц.	прак	сам. роб.
Тема 1. Управління проєктами з розробки програмних систем.	36	10	6	20	42	2	-	40
Тема 2. Управління процесом розробки програмного забезпечення.	36	10	6	20	34	2	2	30
Тема 3. Програмні метрики як невід'ємна частина стандартних процесів розробки	38	10	8	20	40	-	2	38

програмного забезпечення.								
Тема 4. Управління якістю програмних систем	40	12	8	20	34	2	2	30
<i>Усього годин</i>	150	42	28	80	150	6	6	138
ПІДСУМКОВИЙ КОНТРОЛЬ - ЗАЛІК								

4. ПЛАН – КОНСПЕКТ ЛЕКЦІЙ ДИСЦИПЛІНИ

«Управління проектами і якістю програмних систем»

Тема 1. Програмний проект. Управління змістом у програмному проекті.
Розроблення установчих документів програмного проекту та його ініціювання

- Основні види діяльності з управління програмним забезпеченням:
 - Управління процесом розробки програмного забезпечення: Організація та керування процесом створення програмного забезпечення.
 - Створення плану розробки програмного забезпечення (SDP): Розробка докладного плану, який включає в себе кроки, ресурси, графіки та розподіл завдань.
 - Розробка звіту про дані ресурсів програмного забезпечення (SDRP): Збір та аналіз інформації про ресурси, витрати та результати розробки.
 - Вибір підходу до розробки програмного забезпечення: Визначення методології та процесу розробки.
 - Управління програмними ризиками: Ідентифікація, аналіз та керування ризиками, пов'язаними з проектом розробки ПЗ.
 - Прийняття рішення щодо використання програмного забезпечення з відкритим кодом: Вибір між розробкою "з нуля" та використанням відкритих рішень.
- Інші завдання з управління програмним забезпеченням:

- Вибір програмних метрик: Визначення метрик, які вимірюють продуктивність та якість програмного забезпечення.
 - Інтеграція моделі зрілості можливостей (СММІ): Впровадження моделі для покращення процесів розробки ПЗ.
 - Планування програмного забезпечення в плані системної інженерії (SEP): Врахування програмних аспектів у загальному плані розробки системи.
 - Розробка програмного забезпечення: Сам процес розробки ПЗ, включаючи дизайн, реалізацію та тестування.
 - Оцінка програмного ризику: Визначення ймовірних загроз та розробка стратегій їх запобігання.
 - Оцінка розміру програмного забезпечення: Визначення кількості коду, функціональних точок тощо.
 - Розгляд процесу оцінки програмного забезпечення: Визначення процедур для визначення якості та продуктивності ПЗ.
 - Оцінка можливості повторного використання програмного забезпечення: Визначення, чи може ПЗ бути використане у подальших проектах.
 - Рекомендації щодо програмного забезпечення: Розробка рекомендацій для покращення якості та продуктивності ПЗ.
 - Рекомендації щодо використання державного програмного забезпечення: Вказівки щодо використання урядових ПЗ у проектах.
3. Особливості програмного забезпечення для управління ІТ-проектами:
- Планування та складання графіків проекту: Розробка розкладу проекту та визначення завдань.
 - Командна співпраця: Управління командою та спільною роботою.
 - Управління ресурсами: Ефективне використання ресурсів проекту.
 - Документаційне забезпечення управління: Створення необхідних документів для відстеження та звітності.

Тема 2. Управління процесом розробки програмного забезпечення.

1. Переваги добре продуманого і добре реалізованого процесу розробки програмного забезпечення:

- Гарантія якості: Забезпечення високої якості продукту та відсутності помилок.
- Послідовність та повторюваність: Стандартизований процес здатний до повторення.
- Співпраця та координація: Забезпечення ефективної комунікації та співпраці між учасниками проекту.
- Управління ризиками: Ідентифікація, оцінка та керування ризиками на ранніх етапах розробки.
- Масштабованість і ефективність: Здатність адаптувати процес до різних масштабів проектів.
- Постійне вдосконалення: Можливість впровадження змін та вдосконалень на основі досвіду.

2. Етапи процесу розробки програмного забезпечення:

- Планування (Planning): Визначення мети проекту, обсягу та ресурсів, розроблення розкладу.
- Реалізація (Implementing): Написання, тестування та документування програмного коду.
- Тестування (Testing): Перевірка програмного забезпечення на відповідність вимогам та виявлення помилок.
- Розгортання та обслуговування (Deployment and Maintenance): Впровадження програмного продукту та надання підтримки.

3. План розробки програмного забезпечення (SDP): Це документ, який визначає обсяг робіт, графік та ресурси для реалізації програмного проекту. Він є важливим інструментом для управління процесом розробки.

4. Підходи до розробки програмного забезпечення: Водоспадний, Інкрементальний, Спіральний, Agile та Scrum - це різні методології та

підходи до розробки програмного забезпечення, кожен з яких має свої особливості та відмінності.

5. Метрики процесу розробки програмного забезпечення: Це кількісні показники, що використовуються для оцінки ефективності та якості розробки, наприклад, тривалість проекту, кількість виявлених помилок тощо.
6. Мета плану розробки програмного забезпечення (SDP): Метою SDP є забезпечення структурованого та керованого процесу розробки з метою вчасної та якісної поставки програмного продукту.
7. Створення плану розробки програмного забезпечення (SDP): Розробка SDP передбачає визначення обсягу робіт, розподіл ролей, встановлення критеріїв успіху та розробку розкладу.
8. Розробка звіту про дані ресурсів програмного забезпечення (SDRP): Це документ, який містить інформацію про використання ресурсів (час, бюджет, персонал) під час розробки програмного продукту.
9. Program Management Offices (PMO): PMO - це організаційна структура, яка відповідає за управління проектами та програмами в організації. PMO допомагає впроваджувати кращі практики та забезпечувати успішну реалізацію проектів.

Тема 3. Програмні метрики як невід'ємна частина стандартних процесів розробки програмного забезпечення.

1. Програмні метрики, які мають бути визначені в Плані розробки програмного забезпечення (SDP):
 - Розмір програмного забезпечення: Може бути виміряний у ряді одиниць, таких як рядки коду, функціональні точки тощо.
 - Зусилля на розробку ПЗ: Визначення кількості робочих годин або людських ресурсів, необхідних для реалізації проекту.
 - Графік розробки ПЗ: Визначення термінів завершення окремих етапів проекту та проекту в цілому.
 - Дефекти програмного забезпечення: Підрахунок кількості та серйозності виявлених дефектів.

- Визначення вимог до ПЗ та стабільність: Оцінка змін в вимогах до програмного продукту та їх вплив на стабільність проекту.
 - Штат розробників ПЗ: Визначення кількості та кваліфікації співробітників, які беруть участь у проекті.
 - Розвиток програмного забезпечення (дизайн, кодування та тестування): Вимірювання зусиль та часу, витраченого на кожен з етапів розробки.
2. Оцінювання розміру ПЗ: Для оцінки розміру програмного забезпечення використовуються різні метрики, які можуть бути адаптовані до конкретних потреб проекту.
 3. Оцінювання зусиль на розробку ПЗ: Визначення зусиль вимагає оцінювання праці, яку витрачає команда розробників.
 4. Оцінювання часу розробки ПЗ: Графік розробки дозволяє визначити, коли очікувати завершення проекту.
 5. Оцінювання кількості дефектів: Визначення кількості та серйозності дефектів, виявлених під час тестування та після впровадження.
 6. Оцінювання програмного ризику: Оцінка ризиків, пов'язаних з проектом, і розробка стратегій їх управління.

Тема 4. Управління якістю програмних систем

1. Забезпечення якості, контроль якості та тестування як основи управління якістю програмного забезпечення:
 - Забезпечення якості (Quality Assurance): Заходи та процеси, спрямовані на створення відповідних умов для досягнення високої якості.
 - Контроль якості (Quality Control): Механізми та процедури, призначені для визначення, чи відповідає програмне забезпечення стандартам якості.
 - Тестування: Систематична перевірка програмного забезпечення для виявлення дефектів та підтвердження відповідності вимогам.
2. Два основних рівня якості програмного забезпечення:
 - Функціональна якість: Відноситься до відповідності функціональних вимог та очікувань користувачів.

- Нефункціональна якість: Оцінює атрибути, які не відносяться до функцій, наприклад, продуктивність, безпека, надійність тощо.
3. Основні принципи тестування програмного забезпечення:
 - Тестування покриттям: Всі частини програми повинні бути перевірені.
 - Позитивне та негативне тестування: Тестування, що перевіряє, як програма повинна поводитися (позитивне) та як не повинна (негативне).
 - Групування тестів: Тестові випробування повинні бути груповані за категоріями, які тестують різні аспекти програми.
 4. Роль тестування в життєвому циклі розробки програмного забезпечення: Тестування є важливою частиною життєвого циклу розробки, яка включає в себе етапи планування, дизайну, реалізації, тестування та обслуговування.
 5. Методи тестування програмного забезпечення: Різні методи включають модульне тестування, інтеграційне тестування, системне та приймальне тестування, тестування на продуктивність, безпеку, забезпечення якості коду тощо.
 6. Види тестування програмного забезпечення: Сюди входять функціональне тестування, тестування на відповідність вимогам, тестування на стійкість, тестування на витривалість, альфа- та бета-тестування тощо.
 7. Оцінювання якості ПЗ: Використання програмних метрик, таких як кількість виявлених дефектів, покриття коду, продуктивність, надійність тощо, для оцінювання якості програмного забезпечення.

5. ПИТАННЯ ДО ПРАКТИЧНИХ ЗАНЯТЬ

Тема 1. Управління проєктами з розробки програмних систем.

Основні види діяльності з управління програмним забезпеченням: Управління процесом розробки програмного забезпечення; Створення плану розробки програмного забезпечення (SDP); Розробка звіту про дані ресурсів програмного забезпечення (SDRP); Вибір підходу до розробки програмного забезпечення; Управління програмними ризиками; Прийняття рішення щодо використання програмного забезпечення з відкритим кодом.

Інші завдання з управління програмним забезпеченням: Вибір програмних метрик; Інтеграція моделі зрілості можливостей (CMMI); Планування

програмного забезпечення в плані системної інженерії (SEP); Розробка програмного забезпечення; Оцінка програмного ризику; Оцінка розміру програмного забезпечення; Розгляд процесу оцінки програмного забезпечення; Оцінка можливості повторного використання програмного забезпечення; Рекомендації щодо програмного забезпечення; Рекомендації щодо використання державного програмного забезпечення.

Виконання завдань з розробки системної безпеки.

Особливості програмного забезпечення для управління ІТ-проектами. Планування та складання графіків проекту. Командна співпраця. Управління ресурсами. Документаційне забезпечення управління. Звітність.

Тема 2. Управління процесом розробки програмного забезпечення.

Програмні метрики, які мають бути визначені в Плані розробки програмного забезпечення (SDP).

Розмір програмного забезпечення. Зусилля на розробку ПЗ. Графік розробки ПЗ. Дефекти програмного забезпечення. Визначення вимог до ПЗ та стабільність. Штат розробників ПЗ. Розвиток програмного забезпечення (дизайн, кодування та тестування). Використання обчислювальних ресурсів.

Оцінювання розміру ПЗ. Оцінювання зусиль на розробку ПЗ. Оцінювання часу розробки ПЗ. Оцінювання кількості дефектів. Оцінювання програмного ризику. Як приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики. Як конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу.

Тема 3. Програмні метрики як невід’ємна частина стандартних процесів розробки програмного забезпечення.

Програмні метрики, які мають бути визначені в Плані розробки програмного забезпечення (SDP).

Розмір програмного забезпечення. Зусилля на розробку ПЗ. Графік розробки ПЗ. Дефекти програмного забезпечення. Визначення вимог до ПЗ та стабільність. Штат розробників ПЗ. Розвиток програмного забезпечення (дизайн, кодування та тестування). Використання обчислювальних ресурсів.

Оцінювання розміру ПЗ. Оцінювання зусиль на розробку ПЗ. Оцінювання часу розробки ПЗ. Оцінювання кількості дефектів. Оцінювання програмного ризику. Як здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника. Як планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення.

Тема 4. Управління якістю програмних систем

Забезпечення якості, контроль якості та тестування як основи управління якістю програмного забезпечення. Визначення якості як ступіні відповідності явним або неявним вимогам і очікуванням. Визначення термінів: Quality Assurance, Quality Control та Testing. Два основних рівня якості програмного забезпечення. Функціональна якість. Нефункціональна якість. Структурна якість програмного забезпечення та її забезпечення за допомогою перевірки коду, аналізу та рефакторингу. Забезпечення функціональної якості за допомогою контролю якості та тестування.

Основні принципи тестування програмного забезпечення. Роль тестування в життєвому циклі розробки програмного забезпечення. Методи тестування програмного забезпечення. Види тестування програмного забезпечення.

Оцінювання якості ПЗ. Програмні метрики, що використовуються для оцінювання якості програмних систем. Як розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення.

Як розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

Як забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.

6. САМОСТІЙНА РОБОТА

До самостійної роботи студентів включаються:

1. Знайомство з науковою та навчальною літературою відповідно зазначених у програмі тем.
2. Опрацювання лекційного матеріалу.
3. Підготовка до практичних занять.
4. Консультації з викладачем протягом семестру.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка до підсумкового контролю.

Тематика та питання до самостійної підготовки та індивідуальних завдань

Тема 1. Управління проєктами з розробки програмних систем.

Основні види діяльності з управління програмним забезпеченням: Управління процесом розробки програмного забезпечення; Створення плану розробки програмного забезпечення (SDP); Розробка звіту про дані ресурсів програмного забезпечення (SDRP); Вибір підходу до розробки програмного забезпечення; Управління програмними ризиками; Прийняття рішення щодо використання програмного забезпечення з відкритим кодом.

Інші завдання з управління програмним забезпеченням: Вибір програмних метрик; Інтеграція моделі зрілості можливостей (CMMI); Планування програмного забезпечення в плані системної інженерії (SEP); Розробка програмного забезпечення; Оцінка програмного ризику; Оцінка розміру програмного забезпечення; Розгляд процесу оцінки програмного забезпечення; Оцінка можливості повторного використання програмного забезпечення; Рекомендації щодо програмного забезпечення; Рекомендації щодо використання державного програмного забезпечення.

Виконання завдань з розробки системної безпеки.

Особливості програмного забезпечення для управління ІТ-проектами. Планування та складання графіків проекту. Командна співпраця. Управління ресурсами. Документаційне забезпечення управління. Звітність.

Тема 2. Управління процесом розробки програмного забезпечення.

Програмні метрики, які мають бути визначені в Плані розробки програмного забезпечення (SDP).

Розмір програмного забезпечення. Зусилля на розробку ПЗ. Графік розробки ПЗ. Дефекти програмного забезпечення. Визначення вимог до ПЗ та стабільність. Штат розробників ПЗ. Розвиток програмного забезпечення (дизайн, кодування та тестування). Використання обчислювальних ресурсів.

Оцінювання розміру ПЗ. Оцінювання зусиль на розробку ПЗ. Оцінювання часу розробки ПЗ. Оцінювання кількості дефектів. Оцінювання програмного ризику.

Тема 3. Програмні метрики як невід'ємна частина стандартних процесів розробки програмного забезпечення.

Програмні метрики, які мають бути визначені в Плані розробки програмного забезпечення (SDP).

Розмір програмного забезпечення. Зусилля на розробку ПЗ. Графік розробки

ПЗ. Дефекти програмного забезпечення. Визначення вимог до ПЗ та стабільність. Штат розробників ПЗ. Розвиток програмного забезпечення (дизайн, кодування та тестування). Використання обчислювальних ресурсів.

Оцінювання розміру ПЗ. Оцінювання зусиль на розробку ПЗ. Оцінювання часу розробки ПЗ. Оцінювання кількості дефектів. Оцінювання програмного ризику.

Тема 4. Управління якістю програмних систем

Забезпечення якості, контроль якості та тестування як основи управління якістю програмного забезпечення. Визначення якості як ступіні відповідності явним або неявним вимогам і очікуванням. Визначення термінів: Quality Assurance, Quality Control та Testing. Два основних рівня якості програмного забезпечення. Функціональна якість. Нефункціональна якість. Структурна якість програмного забезпечення та її забезпечення за допомогою перевірки коду, аналізу та рефакторингу. Забезпечення функціональної якості за допомогою контролю якості та тестування.

Основні принципи тестування програмного забезпечення. Роль тестування в життєвому циклі розробки програмного забезпечення. Методи тестування програмного забезпечення. Види тестування програмного забезпечення.

Оцінювання якості ПЗ. Програмні метрики, що використовуються для оцінювання якості програмних систем. Оцінювання якості ПЗ за допомогою програмних метрик RFC, CBO та WMC.

7. ВИДИ ТА МЕТОДИ КОНТРОЛЮ

Робоча програма навчальної дисципліни передбачає наступні види та методи контролю:

Види контролю	Складові оцінювання
поточний контроль , який здійснюється у ході: проведення практичних занять, виконання індивідуального завдання; проведення консультацій та відпрацювань.	50%
підсумковий контроль , який здійснюється у ході проведення іспиту.	50%

Методи діагностики знань (контролю)	фронтальне опитування; наукова доповідь, усне повідомлення, індивідуальне опитування, робота у групах, розв'язання задач і практичних завдань, іспит
--	--

8. Питання до заліку

1. Управління проектами з розробки програмних систем
2. Основні види діяльності з управління програмним забезпеченням
3. Завдання з управління програмним забезпеченням
4. Особливості програмного забезпечення для управління ІТ-проектами.
5. Етапи процесу розробки програмного забезпечення.
6. Метрики процесу розробки програмного забезпечення, які визначають в Плані розробки програмного забезпечення (SDP).
7. Program Management Offices (PMO).
8. Створення плану розробки програмного забезпечення (SDP).
9. Розробка звіту про дані ресурсів програмного забезпечення (SDRP).
10. Оцінювання розміру ПЗ.
11. Оцінювання зусиль на розробку ПЗ.
12. Оцінювання часу розробки ПЗ.
13. Оцінювання кількості дефектів.
14. Оцінювання програмного ризику.
15. Забезпечення якості, контроль якості та тестування як основи управління якістю програмного забезпечення.
16. Визначення якості як ступіні відповідності явним або неявним вимогам і очікуванням. Визначення термінів: Quality Assurance, Quality Control та Testing.
17. Два основних рівня якості програмного забезпечення. Функціональна якість. Нефункціональна якість.
18. Структурна якість програмного забезпечення та її забезпечення за допомогою перевірки коду, аналізу та рефакторингу.
19. Забезпечення функціональної якості за допомогою контролю якості та тестування.
20. Основні принципи тестування програмного забезпечення.
21. Роль тестування в життєвому циклі розробки програмного забезпечення.
22. Методи тестування програмного забезпечення.
23. Види тестування програмного забезпечення.
24. Оцінювання якості ПЗ.
25. Програмні метрики, що використовуються для оцінювання якості програмних систем.
26. Метод визначення ступеня гнучкості методології програмного проекту та його застосування.
27. Порівняння класів моделей процесу управління проектом.
28. Процесні моделі управління програмним проектом.
29. Стадійні моделі управління програмним проектом.

30. Наукові школи стратегічного управління (за Г. Мінцбергом).
31. Модель процесу управління портфелем проектів за стандартом PMI, за стандартом P2M PMAJ
32. Типи та функції офісу портфеля програмних проектів.
33. Поняття, класифікація та характеристики історій користувача у SCRUM.
34. Стандартизовані методи вимірювання функціонального розміру програмної системи.
35. Переваги й обмеження методів критичного ланцюга та доказового планування.
36. Планування проекту, релізу, ітерації у методології SCRUM.
37. Засоби ефективної мотивації та антипатерни демотивації виконавців програмного проекту.
38. Цілі та засоби управління комунікаціями в програмному проекті.
39. Підходи до вдосконалення процесу розроблення QIP та IDEAL.
40. Засоби підвищення результативності й економічної ефективності у гнучких методологіях розробки програмних продуктів.
41. Рамкова структура та функції корпоративного стандарту організації-розробника програмного забезпечення
42. Структура системи управління програмними проектами та процедура її запровадження в організації-розробнику програмного забезпечення.
43. Універсальні та спеціальні системи сертифікації виконавців програмних проектів.
44. Призначення, склад, виконавці, продукти та регламентуючі документи процесів управління інтеграцією у програмному проекті згідно з PMBOK PMI 6 ed.
45. Сутність і застосування підходу «розгортання якості» (SQFD) у програмному проекті.
46. Сутність, подібності й розбіжності підходів до управління ризиком невиконання програмного проекту: Continuous Risk Management (SEI) та Дисципліни управління ризиками MSF.
47. Засади ефективного керівництва виконавцями програмного проекту.
48. Призначення, склад, виконавці, продукти та регламентуючі документи процесів управління змістом програмного проекту згідно з PMBOK PMI 6 ed.
49. Призначення, сутність, подібності й розбіжності моделей ефективності ділових процесів за М. Болдріджем, Європейської моделі ділової досконалості, IPMA Delta.
50. Сутність та призначення основних подань програмного проекту.
51. Основні проблеми формування документу вимог у програмному проекті

- та засоби їх опрацювання.
52. Рольові моделі команди програмного проекту.
 53. Склад, сутність та призначення основних артефактів програмного проекту.
 54. Склад та структура внутрішнього ділового середовища програмного проекту
 55. Призначення, переваги й обмеження застосування моделей IPMA Delta та ORM3 PMI для управління якістю програмного проекту.
 56. Склад, сутність, подібності й розбіжності моделей зрілості процесу розроблення програмних систем.
 57. Сутність, подібності й розбіжності застосування підходів SRE SEI та Agena Risk до оцінювання ризику невиконання програмного проекту.
 58. Сутність підходу ситуаційного лідерства за Херші-Бланшаром та його застосування у програмному проекті.
 59. Призначення, склад, виконавці, продукти та засоби процесів управління часом у програмному проекті згідно з PMBOK PMI 6 ed.
 60. Як аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.
 61. Як ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення.
 62. Як забезпечувати якість програмного забезпечення.
 63. Сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення
 64. Ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

9. КРИТЕРІЇ ПІДСУМКОВОЇ ОЦІНКИ ЗНАНЬ СТУДЕНТІВ

Рівень знань оцінюється:

- «відмінно» / «зараховано» А - від 90 до 100 балів. Студент виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та семінарських заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, проявляє активність і творчість у науково-дослідній роботі;

- «добре» / «зараховано» В - від 82 до 89 балів. Студент володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні

відповіді. Був присутній на лекціях та семінарських заняттях, має конспект з виконаними завданнями до самостійної роботи, проявляє активність і творчість у науково-дослідній роботі;

- «добре» / «зараховано» C - від 74 до 81 балів. Студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи та активність у науково-дослідній роботі;

- «задовільно» / «зараховано» D - від 64 до 73 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи;

- «задовільно» / «зараховано» E - від 60 до 63 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.

- «незадовільно з можливістю повторного складання» / «не зараховано» FX – від 35 до 59 балів. Студент володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.

- «незадовільно з обов'язковим повторним вивченням дисципліни» / «не зараховано» F – від 1 до 34 балів. Студент не володіє навчальним матеріалом.

Таблиця відповідності результатів контролю знань за різними шкалами

100-бальною шкалою	Шкала за ECTS	За національною шкалою	
		екзамен	залік
90-100 (10-12)	A	Відмінно	зараховано
82-89 (8-9)	B	Добре	
74-81(6-7)	C		
64-73 (5)	D	Задовільно	
60-63 (4)	E		
35-59 (3)	Fx	незадовільно	не зараховано
1-34 (2)	F		

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Строкань, О.В. Управління IT-проектами [Текст]: конспект лекцій. Мелітополь, 2017. 120 с.
2. Основи управління IT проектами [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки»/ КПІ ім. Ігоря Сікорського ; уклад.: В. О. Кузьмініх, Р. А. Тараненко. – Електронні текстові дані (1 файл:1,998 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 75 с.
3. Катренко А.В. Управління IT-проектами. [Книга 1. Стандарти, моделі та методи управління проектами]: [підручник]. / А.В. Катренко – Львів: «Новий Світ-2000», 2013. – 550 с.
4. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Seventh Edition : Project Management Institute, 2021. – 250 p.
5. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 12th Edition. John Willey & Sons, Inc., 2017. – 848 p.

Допоміжна

1. Довгань Л.Є., Мохонько Г.А., Малик І.П. Управління проектами: навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2017. 420 с.
2. Ноздріна Л., Ящук В., Полотай О. Управління проектами. Київ : Центр навчальної літератури, 2020. 432 с.
3. Тарасюк Г.М. Управління проектами : навч. посіб. Київ : Каравела, 2019. 320 с
4. Приймак В. Управління проектами. Навчальний посібник./ В. Приймак. – К.: КНУ ім. Т. Шевченка, 2017. – 380 с.
5. Хігні Д. Основи управління проектами. Харків : Фабула, 2020. 272 с.
6. Строкань, О.В., Мірошніченко М.Ю. Управління IT-проектами: лабораторний практикум. Мелітополь: Видавничо-поліграфічний центр «Люкс», 2020. 135с
7. Bourque P. Guide to the Software Engineering Body of Knowledge. Version 3.0 / P. Bourque, R.E. Fairley – 2013. – 335 p.
8. "ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Project management - Redline," in *ISO/IEC/IEEE 16326:2019(E) - Redline* , vol., no., pp.1-83, 13 Dec. 2019.
9. "ISO/IEC/IEEE International Standard - Systems and software engineering--Life cycle management--Part 3: Guidelines for the application of ISO/IEC/IEEE 12207 (software life cycle processes)," in *ISO/IEC/IEEE 24748-3:2020(E)* , vol., no., pp.1-76, 23 Oct. 2020, doi: 10.1109/IEEEESTD.2020.9238526.
10. "ISO/IEC/IEEE International Standard - Software engineering -- Guidelines for the application of ISO 9001:2015 to computer software," in *ISO/IEC/IEEE 90003:2018(E)* , vol., no., pp.1-86, 30 Nov. 2018, doi: 10.1109/IEEEESTD.2018.8559961.
11. Jones C. Software Methodologies. A Quantitative Guide. / C.Jones. – CRC Press, 2018. – 579 p.
12. F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab and M. Abbas, "The Influence of Agile Methodology (Scrum) on Software Project Management," *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Toyama, Japan, 2019, pp. 145-149, doi:

- 10.1109/SNPD.2019.8935813.
13. Knaster R. SAFe 4.0 Distilled: Applying the Scaled Agile Framework for Lean Software and Systems Engineering / R. Knaster, D. Leffingwell –Addison-Wesley Professional, 2017 – 395 p.
 14. S. Apul, "Bimar Software Quality Portal: Experience and Lessons Learned," *2021 15th Turkish National Software Engineering Symposium (UYMS)*, Izmir, Turkey, 2021, pp. 1-3, doi: 10.1109/UYMS54260.2021.9659580.
 15. Lin Yang, Zhenyu Liu and Lizhi Cai, "Application and method of quality management for small-size software project," *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)*, Shenzhen, 2012, pp. 1-5, doi: 10.1049/cp.2012.2347.
 16. C. Chen, M. Shoga and B. Boehm, "Exploring the Dependency Relationships between Software Qualities," *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Sofia, Bulgaria, 2019, pp. 105-108, doi: 10.1109/QRS-C.2019.00032.
 17. Prykhodko, S. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis / Sergiy Prykhodko, Natalia Prykhodko, Lidiia Makarova // *Proceedings of International Conference “Advanced Computer Information Technologies” (ACIT-2018)*. CEUR Workshop Proceedings. – 2019. – Vol.2300. – Ceske Budejovice, CZECH REPUBLIC. CEUR-WS.org – P.199-202. ISSN 1613-0073. Режим доступа: <http://ceur-ws.org/Vol-2300/Paper48.pdf>
 18. Prykhodko, S. Mathematical Modeling of Effort of Mobile Application Development in a Planning Phase / Sergiy Prykhodko, Natalia Prykhodko, Kateryna Knyryk, Andrii Pukhalevych // *Proceedings of the 1st International Workshop on Information-Communication Technologies & Embedded Systems*, November, 14-15, 2019, Mykolaiv, Ukraine. CEUR Workshop Proceedings. – 2019. – Vol.2516. – CEUR-WS.org – P. 96-105. ISSN 1613-0073. Режим доступа: <http://ceur-ws.org/Vol-2516/paper7.pdf>
 19. Prykhodko, S. Estimating the Efforts of Mobile Application Development in the Planning Phase Using Nonlinear Regression Analysis / Sergiy Prykhodko, Natalia Prykhodko, Kateryna Knyryk // *Applied Computer Systems*. – 2020. – vol. 25, no. 2. – P. 172–179. – ISSN 2255-8683 (print). Режим доступа: <https://doi.org/10.2478/acss-2020-0019>
 20. Prykhodko S. Early LOC Estimation of Web Apps Created Using Yii Framework by Nonlinear Regression Models / Sergiy Prykhodko, Ivan Shutko, Andrii Prykhodko // *WSEAS Transactions on Computers*. – 2021. – Vol. 20. – P. 321-328. – Print ISSN: 1109-2750 – E-ISSN: 2224-2872. DOI: <http://dx.doi.org/10.37394/23205.2021.20.35>
 21. Prykhodko S.B. A nonlinear regression model to estimate the size of web apps created using the CakePHP framework / S. B. Prykhodko, I. S. Shutko, A. S. Prykhodko // *Radio Electronics, Computer Science, Control*. – 2021. – No. 4 (59). – P. 129-139. – ISSN: 1607-3274 – E-ISSN: 2313-688X – DOI: <http://dx.doi.org/10.15588/1607-3274-2021-4-12>
 22. Sabahat N. A Size Estimation Model for Board-Based Desktop Games / N. Sabahat, A.A. Malik, and F. Azam // in *IEEE Access*, vol. 5, 2017. – P. 4980-4990. Режим доступа: <https://doi.org/10.1109/ACCESS.2017.2678459>
 23. Prykhodko S. A Technique for Detecting Software Quality Based on the Confidence and Prediction Intervals of Nonlinear Regression for RFC Metric / S. Prykhodko, N. Prykhodko // *2022 IEEE 17th International Conference on Computer Sciences and*

Information Technologies (CSIT), 2022. – P. 499-502. DOI: <https://doi.org/10.1109/CSIT56902.2022.10000532>

Інформаційні ресурси

1. Національна бібліотека України імені В. І. Вернадського [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: НБУВ, 2013-2015. – Режим доступу: www.nbuv.gov.ua
2. Електронний каталог Національної парламентської бібліотеки України [Електронний ресурс]: [політемат. база даних містить відом. про вітчизн. та зарубіж. кн., брош., що надходять у фонд НПБ України]. – Електронні дані (803 438 записів). – Київ: Нац. парлам. б-ка України, 2002-2015. – Режим доступу: catalogue.nplu.org
3. Український інститут інтелектуальної власності [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: УІПВ, 2017. – Режим доступу: <http://www.uipv.org>
4. Приходько, С.Б. Трьохфакторне нелінійне регресійне рівняння для оцінювання трудомісткості розробки мобільних застосунків у фазі планування / С. Б. Приходько, Н. В. Приходько, К. О. Книрик // Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки», **Том 30 (69) № 5, 2019.** – С. 154-160. – Режим доступу: <https://doi.org/10.32838/2663-5941/2019.5-1/25>
5. Приходько, С.Б. Трьохфакторна нелінійна регресійна модель для оцінювання трудомісткості розробки мобільних застосунків / С. Б. Приходько, Н. В. Приходько, К. О. Книрик // Моделювання та інформаційні технології : зб. наук. пр. / Нац. акад. наук України, Ін-т пробл. моделювання в енергетиці ім. Г. Є. Пухова. – Київ: [б. в.], 2019. – Вип. 86. – С. 12-19. ISSN 2309-7647 (print) Режим доступу: <http://doi.org/10.5281/zenodo.3610632>
6. Приходько, С.Б. Удосконалення трьохфакторної нелінійної регресійної моделі для оцінювання трудомісткості розробки мобільних застосунків / С. Б. Приходько, Н. В. Приходько, К. О. Книрик // Проблеми інформаційних технологій. – 2019. – № 2 (026). – С. 21-31. – ISSN 1998-7005 (Print), ISSN 2313-0687 (Online). Режим доступу: <https://doi.org/10.35546/2313-0687.2019.26.21-31>
7. Приходько, С.Б. Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php-застосунків з відкритим кодом / С. Б. Приходько, Н. В. Приходько, Т. А. Фаріонова, М. В. Ворона // Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки» **Том 31 (70) № 1, 2020.** – С. 124-131. Режим доступу: <https://doi.org/10.32838/2663-5941/2020.1-1/23>
8. Приходько, С.Б. Чотирьохфакторна нелінійна регресійна модель для оцінювання розміру Java-застосунків з відкритим кодом / С. Б. Приходько, Н. В. Приходько, Т. Г. Смикодуб // Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки» **Том 31 (70) № 2, 2020.** – С. 157-162. Режим доступу: <https://doi.org/10.32838/2663-5941/2020.2-1/25>
9. Приходько, С.Б. Нелінійна регресійна модель для оцінювання розміру Web-застосунків, що створюються з використанням фреймворку Laravel / С. Б. Приходько, Н. В. Приходько, М. В. Ворона, І. О. Беловол // Інформаційні технології та комп'ютерна інженерія. – 2021. – Т.50. – № 1. – С. 115-121. – ISSN 1999-9941. Режим доступу: <https://doi.org/10.31649/1999-9941-2021-50-1-115-121>

10. Easterbrook, S. Selecting Empirical Methods for Software Engineering Research / S. Easterbrook, J. Singer, MA. Storey, D. Damian // In: Shull F., Singer J., Sjøberg D.I.K. (eds) Guide to Advanced Empirical Software Engineering. Springer, London. 2008. Режим доступа: https://doi.org/10.1007/978-1-84800-044-5_11
11. Prykhodko, N.V. The non-linear regression model to estimate the software size of open source Java-based systems / N. V. Prykhodko, S. B. Prykhodko // Radio Electronics, Computer Science, Control. – No.3 (46) – 2018. – P.158-166. – ISSN: 1607-3274 – E-ISSN: 2313-688X – Режим доступа: <https://doi.org/10.15588/1607-3274-2018-3-17>
12. Prykhodko, N.V. A multiple non-linear regression model to estimate the agile testing efforts for small Web projects / N. V. Prykhodko, S. B. Prykhodko // Radio Electronics, Computer Science, Control. – No.2 (49) – 2019. – P. 158-166. – ISSN: 1607-3274 – E-ISSN: 2313-688X. Режим доступа: <https://doi.org/10.15588/1607-3274-2019-2-17>
13. Prykhodko, S. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis / Sergiy Prykhodko, Natalia Prykhodko, Lidiia Makarova // Proceedings of International Conference “Advanced Computer Information Technologies” (ACIT-2018). CEUR Workshop Proceedings. – 2019. – Vol.2300. – Ceske Budejovice, CZECH REPUBLIC. CEUR-WS.org – P.199-202. ISSN 1613-0073. Режим доступа: <http://ceur-ws.org/Vol-2300/Paper48.pdf>
14. Prykhodko, S. Mathematical Modeling of Effort of Mobile Application Development in a Planning Phase / Sergiy Prykhodko, Natalia Prykhodko, Kateryna Knyrik, Andrii Pukhalevych // Proceedings of the 1st International Workshop on Information-Communication Technologies & Embedded Systems, November, 14-15, 2019, Mykolaiv, Ukraine. CEUR Workshop Proceedings. – 2019. – Vol.2516. – CEUR-WS.org – P. 96-105. ISSN 1613-0073. Режим доступа: <http://ceur-ws.org/Vol-2516/paper7.pdf>
15. Prykhodko, S. Estimating the Efforts of Mobile Application Development in the Planning Phase Using Nonlinear Regression Analysis / Sergiy Prykhodko, Natalia Prykhodko, Kateryna Knyrik // Applied Computer Systems. – 2020. – vol. 25, no. 2. – P. 172–179. – ISSN 2255-8683 (print). Режим доступа: <https://doi.org/10.2478/acss-2020-0019>
16. Prykhodko S. Early LOC Estimation of Web Apps Created Using Yii Framework by Nonlinear Regression Models / Sergiy Prykhodko, Ivan Shutko, Andrii Prykhodko // WSEAS Transactions on Computers. – 2021. – Vol. 20. – P. 321-328. – Print ISSN: 1109-2750 – E-ISSN: 2224-2872. DOI: Режим доступа: <http://dx.doi.org/10.37394/23205.2021.20.35>
17. Prykhodko S.B. A nonlinear regression model to estimate the size of web apps created using the CakePHP framework / S. B. Prykhodko, I. S. Shutko, A. S. Prykhodko // Radio Electronics, Computer Science, Control. – 2021. – No. 4 (59). – P. 129-139. – ISSN: 1607-3274 – E-ISSN: 2313-688X – Режим доступа: DOI: <http://dx.doi.org/10.15588/1607-3274-2021-4-12>
18. Sabahat N. A Size Estimation Model for Board-Based Desktop Games / N. Sabahat, A.A. Malik, and F. Azam // in IEEE Access, vol. 5, 2017. – P. 4980-4990. Режим доступа: <https://doi.org/10.1109/ACCESS.2017.2678459>

ЗМІСТ

1. Опис навчальної дисципліни.....	3
2. Програма навчальної дисципліни.....	4
3. Структура навчальної дисципліни.....	6
4. План – конспект лекцій дисципліни «реінжинірінг та оптимізація програмних систем».....	7
5. Питання до практичних занять.....	12
6. Самостійна робота.....	14
7. Види та методи контролю.....	16
8. Питання до заліку.....	17
9. Критерії підсумкової оцінки знань студентів.....	19
10. Рекомендована література.....	21

Навчальне видання

**Стрелковська Ірина Вікторівна
Приходько Сергій Борисович
Григор'єва Тетяна Ігорівна**

УПРАВЛІННЯ ПРОЕКТАМИ І ЯКІСТЮ
ПРОГРАМНИХ СИСТЕМ

Методичні рекомендації для самостійної роботи здобувачів

Українською мовою