

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«__» _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
УПРАВЛІННЯ ПРОГРАМНИМИ
ПРОЕКТАМИ
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	магістр
освітня програма	Програмне забезпечення систем
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2019/2020
Семестр	2
Кількість кредитів ECTS	5
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладач: **к. ф.-м. н., доцент Слабоспицька О.О.** (лекції, лабораторні заняття).

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» 20__ р.

на 20__/20__ н. р. _____ (_____) «__» 20__ р.

Розробник: Слабоспицька Ольга Олександрівна, к. ф.-м. н., с. н. с., доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни «Управління програмними проектами» – засвоєння студентами базових знань щодо актуальних методичних засад, математичних моделей і методів та інформаційних технологій ефективного виконання програмних проектів для сталого досягнення стратегічних цілей організації, яка розробляє конкурентоздатне програмне забезпечення, а також теоретична й практична підготовка студентів до ефективної діяльності в ролях менеджера програмного проекту, члена команди або співробітника проектного офісу.

2. Попередні вимоги до опанування навчальної дисципліни. Для успішного вивчення дисципліни «Управління програмними проектами» студенти повинні:

1. Знати:

1. Основні моделі життєвого циклу програмних систем.
2. Основні підходи та базові шаблони проектування програмних систем.
3. Основні концепції процедурного та об'єктно-орієнтованого програмування.
4. Програмні, організаційні й технологічні засоби забезпечення конкурентоздатної якості програмних систем та керування нею в процесі їх розроблення.
5. Базові методи дискретної оптимізації.
6. Базові методи математичної статистики.

2. Вміти:

1. Вибирати методологію розроблення та мову(и) реалізації програмної системи, найбільш відповідні її призначенню за наявних ресурсів.
2. Виконувати стадії розроблення програмних систем, створюючи їх робочі продукти з якістю, прийнятною для зацікавлених сторін, у парадигмах процедурного та об'єктно-орієнтованого програмування.
3. Розв'язувати задачі лінійного та булевого програмування.
4. Обчислювати стандартні статистичні характеристики вибірки даних (середнє, медіану, середньоквадратичне й стандартне відхилення, квартилі).

3. Володіти:

1. Базовими навичками застосування поширених інтегрованих середовищ розробки програмного забезпечення, зокрема Microsoft Visual Studio, J2EE.
2. Базовими навичками застосування відкритих середовищ розв'язання задач дискретної оптимізації, зокрема Network-Enabled Optimization System (<http://www.neos-server.org/neos/>), MatLab, MS Excel.

4. Успішно опанувати курси освітньо-наукової програми «Програмне забезпечення систем»:

1. Теоретичні основи та методи розробки інформаційних систем.
2. Методи забезпечення якості програмних систем.

А також курси з дискретної математики, програмної інженерії та розроблення інформаційних систем, які викладаються в рамках освітньої програми «Програмна інженерія» циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня «бакалавр».

3. Анотація навчальної дисципліни. Навчальна дисципліна «Управління програмними проектами» є складовою програми підготовки фахівців за другим (магістерським) рівнем вищої освіти галузі знань 12 Інформаційні технології зі спеціальності 121 Інженерія програмного забезпечення, освітньо-наукової програми «Програмне забезпечення систем».

Дисципліна викладається в 2 семестрі в обсязі **150 годин (5 кредитів ECTS)**, зокрема: лекції – 34 год., лабораторні заняття – 14 год., самостійна робота – 100 год., консультації – 2 год. Викладання дисципліни закінчується – **іспитом**.

В результаті вивчення навчальної дисципліни студенти повинні:

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, прийняттого 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 різні 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.

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Управління програмними проєктами» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області управління програмними проєктами відповідно до освітньої кваліфікації магістр з інженерії програмного забезпечення. Зокрема, розвивати:

- Здатність мотивувати людей та рухатися до спільної мети, працювати в команді співробітників (ЗК-4).
- Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності) (ЗК-5).
- Здатність удосконалювати свої навички на основі аналізу попереднього досвіду (ЗК-6).
- Здатність генерувати нові ідеї (креативність) (ЗК-7).
- Здатність приймати обґрунтовані рішення (ЗК-8).
- Здатність аналізувати предметні області, формувати, аналізувати та моделювати вимоги до програмного забезпечення (СК-1).
- Здатність ідентифікувати, класифікувати та описувати проєктні завдання, знаходити раціональні методи й підходи до їх розв'язання (СК-2).
- Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення (СК-4).
- Здатність оцінювати ступінь обґрунтованості застосування специфікацій, стандартів, правил і рекомендацій в професійній галузі та дотримуватися їх при реалізації процесів життєвого циклу програмного забезпечення (СК-5).

- Здатність ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами (СК-6).
- Здатність систематизувати професійні знання щодо створення і супроводження програмного забезпечення (СК-7).
- Здатність розробляти і координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення (СК-8).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	Знати методи реалізації стратегії організації-розробника програмного забезпечення за допомогою програмних проектів.	Лекції, самостійна робота.	Контрольна робота №1, іспит.	5%
PH1.2	Знати де-юре й де-факто стандартизовані універсальні й спеціальні моделі життєвого циклу програмного проекту.	Лекції, лабораторні заняття, самостійна робота.	Контрольна робота №1, іспит.	15%
PH1.3	Знати базові критерії ефективності програмного проекту та методи їх обґрунтованого оцінювання й застосування оцінок для управління ним.	Лекції, лабораторні заняття, самостійна робота.	Контрольні роботи №1, №2, іспит.	20%
PH1.4	Знати методи керування командою програмного проекту та очікуваннями його зацікавлених сторін.	Лекції, самостійна робота.	Контрольна робота №2, іспит.	10%
PH1.5	Знати рамкову структуру та методи побудови й застосування системи управління програмними проектами.	Лекції, самостійна робота.	Контрольна робота №2, іспит.	5%
PH2.1	Вміти вибирати й автоматизовано налаштувати технологію управління програмним проектом згідно з його метою й умовами виконання та керувати його перебігом.	Лабораторні заняття, самостійна робота.	Захист лабораторних робіт, іспит.	20%
PH2.2	Вміти координувати різні програмні проекти	Лекції, самостійна робота.	Контрольна робота №1, іспит.	5%

	в системі управління проектами.			
РН3.1	Зрозуміло формулювати проблеми вивчення теоретичного матеріалу й виконання лабораторних робіт у питаннях до викладача й колег та обґрунтовано відповідати на такі питання з їх боку.	Лекції, лабораторні заняття, години консультацій.	Захист лабораторних робіт.	5%
РН3.2	Послідовно й зрозуміло обґрунтовувати власні рішення в лабораторних роботах.	Лабораторні заняття.	Захист лабораторних робіт.	5%
РН4.1	Самостійно розробляти артефакти управління програмним проектом в лабораторних роботах.	Лабораторні заняття.	Захист лабораторних робіт.	5%
РН4.2	Самостійно аналізувати теоретичний матеріал і практичний досвід для обґрунтування лабораторних і контрольних робіт.	Лабораторні заняття, опрацювання рекомендованих інформаційних джерел.	Контрольні роботи №1, №2, захист лабораторних робіт.	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 1.4	РН 1.5	РН 2.1	РН 2.2	РН 3.1	РН 3.2	РН 4.1	РН 4.2
	Програмні результати навчання										
ПРН-1. Знати і системно застосовувати методи аналізу та моделювання прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування програмного забезпечення.		+	+	+		+	+		+	+	
ПРН-2. Обґрунтовувати вибір методів формування вимог до програмної системи, розробляти, аналізувати та систематизувати вимоги.	+	+	+			+			+	+	
ПРН-3. Знати і застосовувати базові концепції і методології моделювання інформаційних процесів.		+	+	+	+	+	+			+	
ПРН-4. Оцінювати і обирати методи і моделі розробки, впровадження, експлуатації програмних засобів та управління ними на всіх етапах життєвого циклу.	+	+	+	+	+	+		+	+	+	+
ПРН-5. Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати прийняті проектні рішення з точки зору якості кінцевого	+		+			+	+	+	+	+	+

програмного продукту.											
ПРН-9. Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення.	+	+	+	+	+	+	+		+		+
ПРН-10. Вміти приймати організаційно-управлінські рішення в умовах невизначеності.	+	+	+	+	+	+	+		+		+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів.

Семестрове оцінювання:

1. Контрольна робота 1: РН1.1, РН1.2, РН1.3, РН2.2, РН4.2 – **20 балів/12 балів.**
2. Контрольна робота 2: РН1.3, РН1.4, РН1.5, РН4.2 – **20 балів/12 балів.**
3. Лабораторна робота 1: РН3.1, РН3.2, РН4.1, РН 4.2 – **4 бали/2 бали.**
4. Лабораторна робота 2: РН3.2, РН4.1, РН4.2 – **4 бали/2 бали.**
5. Лабораторна робота 3: РН2.1, РН3.1, РН3.2, РН4.1, РН4.2 – **4 бали/2 бали.**
6. Лабораторна робота 4: РН2.1, РН3.1, РН3.2, РН4.1, РН4.2 – **3 бали/1 бал.**
7. Лабораторна робота 5: РН2.1, РН3.1, РН3.2, РН4.1, РН4.2 – **5 балів/3 бали.**

Підсумкове оцінювання (у формі іспиту):

- Максимальна кількість балів які можуть бути отримані студентом: 40 балів.
- Результати навчання, які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН1.4, РН1.5, РН2.1, РН2.2.
- Форма проведення і види завдань: письмова робота.
- Види завдань: 4 письмових завдання (2 тестові завдання, аналітичне та теоретичне питання).
- Для отримання загальної позитивної оцінки з дисципліни оцінка за екзамен має бути не меншою, ніж 24 бали.
- Студент не допускається до екзамену, якщо протягом семестру він набрав менше ніж 36 балів.
- Студент не допускається до екзамену, якщо протягом семестру він не виконав і не здав 100 % лабораторних робіт, передбачених планом.

Критерії оцінювання на іспиті.

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1	Тестове питання за теоретичним матеріалом курсу	12.5%	12.5%
Завдання 2		12,5%	12,5%
Завдання 3	Питання, що потребує аналітичного зіставлення теоретичного матеріалу курсу, лабораторних робіт, власного досвіду	37.5%	37.5%
Завдання 4	Питання за теоретичним матеріалом курсу	37.5%	37.5%
			100%

Питання для підготовки до іспиту.

Тестові питання:

1. Поєднання схемної, системної та сервісної моделей проекту передбачені в методології:
 - a) The Standard for Portfolio Management PMI;
 - b) P2M;
 - c) PRINCE2;
 - d) Concentric Project Management.
2. Виокремлення в організації-розробнику підрозділу, відповідального за прийняття стратегічних рішень у проектах, передбачено в моделі:
 - a) стандарту ISO 21500;
 - b) TenStep;
 - c) моделі «Око» UPMA;
 - d) стандарту PRINCE2.
3. Для створення в організації-розробнику єдиного інформаційного середовища виконання різнорідних програмних проектів найбільш придатна модель процесу управління:
 - a) ISO 21450;
 - b) The Standard for Portfolio Management PMI;
 - c) PRINCE2;
 - d) CPM.
4. Техніка FMEA належить до числа базових при організації програмного проекту згідно з моделлю:
 - a) Six Sigma;
 - b) ICB IPMA;
 - c) Ten Step;
 - d) P2M.
5. Для управління проектом розроблення ПС, критичної для підтримки ділових процесів організації-замовника, найбільш придатна модель:
 - a) описана в ДСТУ 3974-2000;
 - b) передбачена в CleanRoom;
 - c) передбачена в MSF;
 - d) системна модель COBHET.
6. За жорстких обмежень на терміни й ресурси, для узгодженого управління значною кількістю різномасштабних проектів розроблення ПС різного призначення найбільш придатна методологія:
 - a) The Standard for Program Management PMI;
 - b) P2M;
 - c) The Standard for Portfolio Management PMI;
 - d) Concentric Project Management.
7. Для вимірювання функціонального розміру програмних систем з потужними базами даних насамперед призначений метод:
 - a) Full Function Points;
 - b) Feature Points;
 - c) 3D Function Points;
 - d) Mark-II Function Points.
8. Загальна структура корпоративного стандарту управління програмним проектом в організації-розробнику включає:

- a) шаблони документів;
 - b) опис математичного інструментарію;
 - c) опис інструментальних засобів;
 - d) стратегічні цілі організації.
9. Узгоджене вдосконалення проектів, програм та портфелів підтримується моделлю:
- a) SPICE;
 - b) Г. Карцнера;
 - c) OPM3;
 - d) CMMI.
10. Для досягнення вибірових цілей вдосконалення в окремому програмному проекті доцільно використати:
- a) підхід QIP;
 - b) підхід IDEAL;
 - c) модель зрілості CMMI;
 - d) модель TRILLIUM.
11. Співвідношення динаміки запланованого об'єму (BCWS), осягнутого об'єму (BCWP) і дійсної вартості виконаних робіт (ACWP): $BCWP < ACWP$, $BCWP < BCWS$ свідчить, що у програмному проекті:
- a) темп виконання вище запланованого;
 - b) має місце відставання від бюджету й розкладу;
 - c) темп виконання нижче запланованого;
 - d) перевищений бюджет
12. Для встановлення пріоритетів історій користувача в SCRUM слід використати:
- a) техніку «Planning Poker»;
 - b) техніку балансування К. Вігера;
 - c) аналіз за Н. Кано;
 - d) методику CETIN.
13. Для перевірки бюджету, запропонованого замовником, під час складання Статуту програмного проекту можна використати:
- a) модель: COCOMO;
 - b) модель: SEER-SEM;
 - c) метод PERT;
 - d) метод освоєного об'єму.
14. Виокремлення двох спеціальних ролей для подання точки зору на програмний продукт з боку користувачів передбачено в методології
- a) RUP;
 - b) SCRUM;
 - c) DSDM;
 - d) FDD.
15. Система управління програмними проектами в компанії-розробнику обов'язково включає:
- a) корпоративний стандарт;
 - b) шаблони документів;
 - c) раду менеджерів програмних проектів;
 - d) систему менеджменту якості.
16. Згідно кращих практик виконання програмного проекту, Політика управління проектами у корпоративному стандарті має визначати:
- a) класифікацію проектів;
 - b) структуру декомпозиції робіт;

- c) функціональні ролі й відповідальності учасників проектів;
 - d) стратегічні цілі організації.
17. До засобів реалізації Загального управління якістю належить:
- a) цикл PDCA;
 - b) підхід QIP;
 - c) підхід IDEAL;
 - d) підхід Disciplined Agile Delivery.
18. Визначальною характеристикою програмного проекту, спільною для описового та нормативного підходів до управління ним, є:
- a) цілеспрямованість;
 - b) наявність вимог до якості результатів;
 - c) унікальність результатів проекту;
 - d) наявність куратора проекту.
19. Для передбачення поведінки кандидата на участь в програмному проекті, переважної за конфліктної ситуації під час його виконання, доцільно застосувати:
- a) методику STAR;
 - b) опитувальник Белбіна;
 - c) опитувальник Томаса;
 - d) опитувальник TBQ.
20. Для прогнозування значень ризиків невиконання програмного проекту під час складання його концепції можна використати:
- a) метод Software Risk Evaluation SEI;
 - b) підхід Riskology;
 - c) дисципліну управління ризиком в MSF;
 - d) ризико-орієнтовані метрики SATC NASA.
21. Засобом запобігання конфліктів менеджера програмного проекту з власниками ресурсів у слабкій матричній організації є:
- a) план управління проектом;
 - b) статут проекту;
 - c) ієрархічна структура робіт;
 - d) концепція.
22. Зрілість процесу управління програмним проектом в організації-розробнику можна оцінити за допомогою моделі:
- a) системної моделі COBHET;
 - b) SPICE;
 - c) OPM3 PMI;
 - d) CMMI.
23. Критерієм виокремлення пакету робіт в ієрархічній структурі робіт є:
- a) наявність єдиного виконавця;
 - b) наявність ресурсів для виконання;
 - c) можливість виконання без призупинення на виконання інших дій.
24. Для перевірки реалістичності терміну, запропонованого замовником, під час складання Статуту програмного проекту можна використати:
- a) модель COQUALMO;
 - b) метод критичного шляху;
 - c) метод PERT;
 - d) модель С. Снедакер.
25. За моделлю С. Мак Коннела, до критеріїв успішності програмного проекту належить:

- a) створення ІСР;
 - b) ведення реєстру ризиків проекту;
 - c) моніторинг показників виконання проекту;
 - d) ведення реєстру зацікавлених сторін.
26. Для прогнозування успішності програмного проекту з 15 виконавцями найбільш придатна модель:
- a) HERMES;
 - b) С. Снедакер;
 - c) Д. Хогі;
 - d) С. Мак Конелла.
27. Якщо наявна концепція проекту зі створення динамічного сайту, для прогнозу дати його завершення можна застосувати:
- a) модель Д. Клірі;
 - b) модель Д. Ріфера;
 - c) метод А. Альбрехта;
 - d) методику СЕТІН.
28. Опис вимог до компонентів програмної системи стосовно їх взаємодії із зовнішнім оточенням і між собою слід виконувати згідно з:
- a) IEEE 1362;
 - b) ІСР;
 - c) IEEE 1233-1998;
 - d) IEEE 830-1998.
29. Урахування ризиків неможливості реалізації вимог до створюваної програмної системи під час визначення їх пріоритетів можна здійснити за допомогою:
- a) аналізу за Н. Кано;
 - b) процедури К. Вігерса;
 - c) процесу EasyWinWin;
 - d) схеми MoSCoW.
30. Згідно з кращими практиками, ієрархічну структуру робіт успішного програмного проекту має формувати:
- a) куратор;
 - b) бізнес-аналітик;
 - c) менеджер і представники користувачів;
 - d) менеджер разом з командою.
31. Можливі причини появи дефектів у продукті можна впорядкувати за важливістю, користуючись:
- a) контрольними картами;
 - b) діаграмою Парето;
 - c) діаграмою Ісікави;
 - d) діаграмою розсіювання.
32. Для формування вимог до програмного продукту ринкового призначення доцільна техніка:
- a) EasyWinWin;
 - b) прототипування;
 - c) інтерв'ю;
 - d) аналізу впливів.

Аналітичні питання:

1. У чому полягає та якими документами регламентований розвиток рамкової моделі якості програмного забезпечення?
2. Які моделі якості слід застосувати в проєкті розроблення програмної системи з підвищеними вимогами до надійності?
3. Чим розрізняється організація управління програмним проєктом у слабкій, збалансованій і сильній матричній організації?
4. Які моделі слід застосувати для підвищення ефективності команди програмного проєкту?
5. Які моделі якості програмного забезпечення слід застосувати, щоб узгодити погляди споживачів, менеджера та розробників на якість створюваної програмної системи? Опишіть їх подібності й розбіжності.
6. Які моделі дозволяють відстежувати зрілість процесу програмування програмного забезпечення в організації-розробнику? Опишіть їх подібності й розбіжності.
7. За допомогою яких установчих артефактів проєкту можна запобігти «крадінню ресурсів» у програмному проєкті? Опишіть їх сутність.
8. Які властивості мають бути притаманні вимогам в успішному програмному проєкті? Які нормативні документи регламентують перелік цих властивостей?
9. Які методи можна використати для отримання інтервальної оцінки ризику невиконання програмного проєкту під час формування його статуту? Опишіть їх.
10. Які «японські» інструменти аналізу якості можна застосувати в програмному проєкті, для чого і яким чином?
11. Які моделі застосовні для кількісної оцінки успішності програмного проєкту? Опишіть їх подібності й розбіжності.
12. На підставі яких чинників визначається рівень деталізації ієрархічної структури робіт?
13. У яких процесах управління програмним проєктом за РМВоК РМІ і для чого застосовують його ієрархічну структуру робіт?
14. За допомогою яких технік можна підвищувати особисту ефективність учасника програмного проєкту? Стисло опишіть їх сутність.
15. Які методи можна використати для отримання інтервальної оцінки трудомісткості статичного сайту, якщо відомі вимоги до нього? Опишіть їх сутність.
16. Які кількісні характеристики має методологія управління програмними проєктами за А. Коберном і як можна застосувати їх оцінки?
17. Які ролі має виконувати менеджер у програмному проєкті?

Теоретичні питання:

1. Метод визначення ступеня гнучкості методології програмного проєкту та його застосування.
2. Порівняння класів моделей процесу управління проєктом.
3. Процесні моделі управління програмним проєктом.
4. Стадійні моделі управління програмним проєктом.
5. Наукові школи стратегічного управління (за Г. Мінцбергом).

6. Модель процесу управління портфелем проектів за стандартом PMI.
7. Модель процесу управління портфелем проектів за стандартом P2M PMAJ
8. Типи та функції офісу портфеля програмних проектів.
9. Поняття, класифікація та характеристики історій користувача у SCRUM.
10. Методи встановлення пріоритетів історій у SCRUM.
11. Базовий метод А. Альбрехта вимірювання функціонального розміру програмної системи та напрями розвитку базового методу.
12. Стандартизовані методи вимірювання функціонального розміру програмної системи.
13. Визначення розміру динамічних і статичних Web-застосунків та їх трудомісткості.
14. Визначення розміру історій користувача у SCRUM.
15. Оцінювання витрат на програмний проект за моделлю COCOMO II. Опрацювання обмежень моделі COCOMO II.
16. Обчислення та застосування індексів освоєного об'єму (earned value).
17. Застосування техніки освоєного розкладу (earned schedule).
18. Переваги й обмеження методів критичного ланцюга та доказового планування.
19. Засоби запобігання непродуктивних витрат ресурсів у Збережуючому розробленні, Теорії обмежень, Вчасному виробництві.
20. Планування проекту, релізу, ітерації у методології SCRUM.
21. Ефективне управління особистим часом виконавця програмного проекту за підходами Get Things Done та FACT.
22. Парадигма безперервного управління ризиком невиконання програмного проекту та приклади її реалізації в ітеративних і гнучких програмних проектах.
23. Засоби ефективної мотивації та антипатерни демотивації виконавців програмного проекту.
24. Цілі та засоби управління комунікаціями в програмному проекті.
25. Підходи до вдосконалення процесу розроблення QIP та IDEAL.
26. Засоби підвищення результативності й економічної ефективності у гнучких методологіях розробки програмних продуктів.
27. Рамкова структура та функції корпоративного стандарту організації-розробника програмного забезпечення
28. Структура системи управління програмними проектами та процедура її запровадження в організації-розробнику програмного забезпечення.
29. Універсальні та спеціальні системи сертифікації виконавців програмних проектів.
30. Призначення, елементи та кількісні характеристики методології управління програмним проектом за А.Коберном.
31. Призначення, склад, виконавці, продукти та регламентуючі документи процесів управління інтеграцією у програмному проекті згідно з PMBOK PMI 6 ed.
32. Сутність і застосування підходу «розгортання якості» (SQFD) у програмному проекті.
33. Сутність, подібності й розбіжності підходів до управління ризиком невиконання програмного проекту: Continuous Risk Management (SEI) та Дисципліни управління ризиками MSF.

34. Засади ефективного керівництва виконавцями програмного проекту.
35. Призначення, склад, виконавці, продукти та регламентуючі документи процесів управління змістом програмного проекту згідно з РМВОК РМІ 6 ed.
36. Призначення, сутність, подібності й розбіжності моделей ефективності ділових процесів за М. Болдріджем, Європейської моделі ділової досконалості, IPMA Delta.
37. Сутність та призначення основних подань програмного проекту.
38. Основні проблеми формування документу вимог у програмному проекті та засоби їх опрацювання.
39. Рольові моделі команди програмного проекту.
40. Склад, сутність та призначення основних артефактів програмного проекту.
41. Склад та структура внутрішнього ділового середовища програмного проекту
42. Призначення, переваги й обмеження застосування моделей IPMA Delta та OPM3 РМІ для управління якістю програмного проекту.
43. Склад, сутність, подібності й розбіжності моделей зрілості процесу розроблення програмних систем.
44. Сутність, подібності й розбіжності застосування підходів SRE SEI та Agena Risk до оцінювання ризику невиконання програмного проекту.
45. Сутність підходу ситуаційного лідерства за Херші-Бланшаром та його застосування у програмному проекті.
46. Призначення, склад, виконавці, продукти та засоби процесів управління часом у програмному проекті згідно з РМВОК РМІ 6 ed.

7.2 Організація оцінювання

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 7 тижня семестру.
2. Контрольна робота 2: до 14 тижня семестру.
3. Лабораторні роботи 1-3: до 7 тижня семестру.
4. Лабораторні роботи 4,5: до 14 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи у визначений викладачем термін.

У випадку відсутності студента з поважних причин відпрацювання та перескладання контрольних робіт здійснюються у відповідності до «Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу» від 1 жовтня 2010 року.

Студент має право захищати лабораторні роботи протягом усього навчального семестру.

7.3 Шкала відповідності оцінок.

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять.

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабораторні заняття	Само-стійна робота
Частина 1. Універсальні засади та методи управління проектами.				
1	Тема 1. Програмний проект як об'єкт управління.	2		6
2	Тема 2. Управління інтеграцією в програмному проекті.	2	2	8
3	Тема 3. Управління змістом у програмному проекті.	4	4	8
4				
5	Тема 4. Моделі програмного проекту.	2		8
6	Тема 5. Стратегічний менеджмент програмних проектів.	4		8
7				
Контрольна робота 1				2
Всього по частині 1		14	6	40
Частина 2. Моделі, методи та технології обґрунтованого кількісного управління програмними проектами.				
8	Тема 6. Засади, методи та засоби управління витратами.	4	2	10
9				
10	Тема 7. Засади, методи та засоби управління розкладом.	4	6	11
11				
12	Тема 8. Засади, методи та засоби інженерії якості програмного забезпечення.	2		8
13	Тема 9. Засади, методи та засоби управління ризиками.	2		6
14	Тема 10. Засади, методи та засоби управління персоналом, взаємодіями учасників та залученням зацікавлених сторін.	4		11
15				
16	Тема 11. Засади, методи та засоби вдосконалення програмних проектів.	2		6
17	Тема 12. Система управління програмними проектами в організації-розробнику програмного забезпечення	2		6
Контрольна робота 2				2
Всього по частині 2		20	8	60
Консультація			2	
ВСЬОГО		34	14	100

Загальний обсяг – **150** год, в тому числі:

Лекцій – **34** год.

Лабораторні заняття – **14** год.

Самостійна робота – **100** год.
Консультації – **2** год.

Лабораторні роботи.

Лабораторна робота 1: Розроблення установчих документів програмного проекту та його ініціювання.

Лабораторна робота 2: Виявлення, узгодження та документування вимог до програмної системи з урахуванням установчих документів ініційованого проекту.

Лабораторна робота 3: Автоматизована побудова ієрархічної структури робіт ініційованого програмного проекту на підставі його установчих документів і специфікації вимог.

Лабораторна робота 4: Автоматизоване прогнозування витрат на виконання ініційованого програмного проекту за моделлю Б. Боема СОСОМО

Лабораторна робота 5: Автоматизована побудова мережного графа ініційованого програмного проекту та визначення на ньому критичного шляху й складання розкладу проекту.

9. Рекомендовані джерела.

Основні:

1. Приймак В. Управління проектами. Навчальний посібник./ В. Приймак. – К.: КНУ ім. Т. Шевченка, 2017. – 380 с.
2. Катренко А.В. Управління ІТ-проектами. [Книга 1. Стандарти, моделі та методи управління проектами]: [підручник]. / А.В. Катренко – Львів: «Новий Світ-2000», 2013. – 550 с.
3. Бурков В.Н. (ред.) Математические основы управления проектами: Учебное пособие – М.: Высшая школа, 2005. – 423 с.
4. Селиховкин И. Управление ИТ-проектом. Эффективная система “с нуля” в любой организации – Санкт-Петербург, 2010. [Електронний ресурс]. Режим доступу: pmlead.ru/ITPM-Selikhovkin.pdf. – 90 с.
5. Архипенков С. Лекции по управлению программными проектами – М.: 2009. – 128 с. [Електронний ресурс]. – Режим доступу:
http://www.arkhipenkov.ru/resources/sw_project_management.pdf.
6. Вигерс К. Разработка требований к программному обеспечению. 3-е изд., дополненное / К. Вигерс, Дж. Битти – Пер. с англ. – М. Изд.-во «Русская редакция»; СПб.: БХВ-Петербург, 2014. – 736 с.
7. Вольфсон Б. Гибкое управление проектами и продуктами. / Б. Вольфсон – СПб.: Питер, 2015. – 144 с.
8. Перерва А. Путь IT-менеджера. Управление проектной средой и IT-проектами. / А. Перерва, С. Еранов, В. Иванова, С. Сергеев – СПб.: Питер, 2016. – 320 с.
9. Сазерленд Дж. Scrum. Навчись робити вдвічі більше за менший час / Дж. Сазерленд Клуб Сімейного Дозвілля, 2016.

Додаткові:

1. Руководство к Своду знаний по управлению проектами (Руководство PMBoK). 6-е издание. – PMI, 2017. – 756 p.
2. Bourque P. Guide to the Software Engineering Body of Knowledge. Version 3.0 / P. Bourque, R.E. Fairley – 2013. – 335 p.
3. Ципес Г. Проекты и управление проектами в практике современной компании. / Г. Ципес, А. Товб – М.: Олимп-бизнес, 2009. – 480 с.
4. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И. Управление программными проектами. Достижение оптимального качества при минимуме затрат – М.; СПб.; К.: Издательский дом «Вильямс», 2004. – 1126 с.
5. Jones C. Software Methodologies. A Quantitative Guide. / C.Jones. – CRC Press, 2018. – 579 p.
6. Офіційний сайт «Метод 123». [Електронний ресурс]. – Режим доступу: <http://www.method123.com>.
7. Офіційний сайт компанії TenStep. [Електронний ресурс]. – Режим доступу: <http://www.tenstep.com>.
8. Individual Excellence Baseline for achieving excellence in Projects and Programs. V. 4.0 – IPMA, 2015. – 432 p.
9. Project Excellence Baseline for achieving excellence in Projects and Programs. V.1.0 – IPMA, 2016. – 113 p.
10. Коберн А. Современные методы описания функциональных требований к системам – М., Лори, 2002. – 263 с.
11. 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.
12. Блог Д. Леффингвелла «Scaling Software Agility» – Режим доступу <http://scalingsoftwareagilityblog.com/>.
13. Кон М. Scrum: гибкая разработка ПО. – К.: Вильямс, 2011 – 576 с.
14. Кон М. Пользовательские истории: гибкая разработка программного обеспечения» – Вильямс, 2012 – 256 с.
15. Сайт М. Кона – Режим доступу <http://www.mountangoatsoftware.com/>.
16. Минцберг Г. Стратегическое сафари: Экскурсия по дебрям стратегического менеджмента / Генри Минцберг, Брюс Альстранд, Жозеф Лампель; Пер. с англ. – М.: Альпина Паблишер, 2013. – 367 с.
17. Аньшин В.М., Бархатов В.Д. Управление портфелем проектов: сравнительный анализ подходов и рекомендации по их применению – Управление проектами и программами– 2012. – № 01(29) – С. 20-40.
18. The Standard for Portfolio Management. Fourth Ed. – PMI, 2017. – 190 p.
19. Остервальдер А. Построение бизнес-моделей: Настольная книга стратега и инноватора / А. Остервальдер, И. Пинье – 3-е изд. – М.: Альпина Паблишер, 2012. – 288 с.
20. Бланк С. Четыре шага к озарению: Стратегии создания успешных стартапов / С. Бланк – М.: Альпина Паблишер, 2016. – 368 с.

21. Рис Э. Бизнес с нуля. Метод Lean Startup для быстрого тестирования идей и выбора бизнес-модели / М.: Альпина Паблишер, 2018 – 256 с.
22. Офіційний сайт проекту «Ремесло бізнес-моделювання». [Електронний ресурс]. – Режим доступу: <http://bizmodelgu.ru/>.
23. Офіційний сайт міжнародної спільноти користувачів одиниць функціональності (IFPUG) – Режим доступу <http://www.ifpug.org/>.
24. Мак Коннелл С. Сколько стоит программный проект – М.: «Русская Редакция», СПб.: Питер, 2007. – 297 с.
25. Сторінка «COCOMO® II» офіційного сайту Університету Південної Каліфорнії – Режим доступу http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html.
26. Rusk J. Earned Value for Agile Development – 2009. – Режим доступу <http://www.agilekiwi.com/EarnedValueForAgileProjects.pdf>.
27. Davis A. Earned schedule. An emerging Earned Value technique. / A.Davis, M.Higgins. – 2010. [Електронний ресурс] – Режим доступу
28. <http://www.apm.org.uk/sites/default/files/2010030220EVM20WP%200110.pdf>.
29. Лоуренс Л. Вовремя и в рамках бюджета. Управление проектами по методу критической цепи / Л. Лоуренс – Альпина Паблишер, 2010 – 354 с
30. Аллен Д. Как привести дела в порядок: искусство продуктивности без стресса. / Д. Аллен – М.: Манн, Иванов и Фербер, 2014. – 368 с.
31. Спольски Дж. Доказательное планирование. [Електронний ресурс] – Режим доступу: <http://local.joelonsoftware.com/wiki/Russian>.
32. Детмер У. Теория ограничений Голдратта: Системный подход к непрерывному совершенствованию / У. Детмер; Пер. с англ. –4-е изд. – М.: Альпина Паблишер, 2012. – 443 с.
33. Jones C. The Economics of Software Quality / C. Jones, J. Subramanyam, O. Bonsignour – Addison-Wesley, 2011. – 621 p.
34. Law E.L.-C. Maturing Usability. Quality in Software, Interaction and Value / E.L.-C. Law, E.T. Hvannberg, G. Cockton. – Springer-Verlag London Limited, 2008. – 446 p.
35. Имаи М. Кайдзен: Ключ к успеху японских компаний. / М. Имаи – Альпина Паблишер, 2016. – 274 с.
36. Джексон Т. Хосинканри: как заставить стратегию работать / Т. Джексон – М.: Институт комплексных стратегических исследований, 2008. – 248 с.
37. Кандалинцев В.Г. Инновационный бизнес: применение сбалансированной системы показателей/ В.Г. Кандалинцев. – М.: Издательский дом «Дело» РАНХиГС, 2015. – 158 с.
38. Баркалов С.А., Белоусов В.Е., Маилян Л.Р. Теория и практика управления качеством в социально-экономических системах: Монография – Воронеж: Изд.-во «Научная книга», 2013. – 580 с.
39. Alberts C.J. Risk Management Framework / C.J. Alberts, A.J. Dorofee. – Technical Report CMU/SEI-2010-TR-017 ESC-TR-2010-017 – Carnegie Mellon University, 2010. – 72 p.
40. Сторінка «Практичне управління ризиками» консультативної компанії Mosaic (Австралія). [Електронний ресурс] – Режим доступу: https://mosaicprojects.com.au/Practical_Risk_Management.html.

41. Листер Т. Вальсируя с медведями: управление рисками в проектах по разработке программного обеспечения / Т. Листер, Т. Де Марко – Компания p.m. Office, 2005 – 311 с.
42. Fenton N. Risk Assessment and Decision Analysis with Bayesian Networks. / N. Fenton, M. Neil. – CRC Press, 2012 – 524 p.
43. Жуков Ю.М. Технологии командообразования. / Ю.М. Жуков, А.В. Журавлев, Е.Н. Павлова – М: Аспект-Пресс, 2008 – 387 с.
44. Архипенков С. Руководство командой разработчиков программного обеспечения. Прикладные мысли. – 2008. – 79 с. [Электронный ресурс] – Режим доступа http://www.arkhipenkov.ru/resources/sw_team_management.pdf.
45. Рейнвотер Дж. Как пасти котов. Наставление для программистов, руководящих другими программистами / Дж. Рейнвотер – Питер, 2016. – 256 с.
46. Дерби Э. Agile ретроспектива. Как превратить хорошую команду в великую / Э. Дерби, Д. Ларсен – Альпина Паблишер, 2017. – 208 с.
47. Сайт Т. Євграшина. [Электронный ресурс] – Режим доступа: <https://tim.com.ua/about/>.
48. Сайт «Клуб Успешных Менеджеров Программистов». [Электронный ресурс] – Режим доступа: <http://www.happy-pm.com>.
49. Чанько А.Д. Команды в современных организациях (+CD): учебник /А.Д. Чанько; Высшая школа менеджмента СПбГУ. – СПб: Изд-во «Высшая школа менеджмента», 2011. – 408 с.
50. Mantle M.W. Managing the unmanageable: Rules, tools, and insights for managing people and teams. / M.W. Mantle, R. Lichty – Addison-Wesley Professional, 2012. – 422 p.
51. Сидоренко Е.В. Тренинг коммуникативной компетентности в деловом взаимодействии. / Е.В. Сидоренко. – Речь, 2008. – 116 с.
52. Рыженкова И.К. Профессиональные навыки менеджера: Повышение личной и командной эффективности / Рыженкова И.К. – 2-е изд. – М.: Эксмо, 2016. – 272 с. (Полный курс MBA).
53. Таратухина Ю. В. Деловая коммуникация в сфере информационных технологий: уч. пособие / Ю.В. Таратухина; под общ. ред. С.В. Мальцевой. – Москва: «ART-менеджер», 2011. – 200 с.
54. Mayfield P. Practical People Engagement: Leading Change through the Power of Relationships / P. Mayfield – 2013. – 284 p.
55. Gleeson P. Working with Coders. A Guide to Software Development for the Perplexed Non-Techie / P. Gleeson – Apress, 2017. – 227 p.
56. Дэвис Дж. Философия DevOps. Искусство управления IT / Дж. Дэвис, К. Дэниелс – СПб.: Питер, 2017. – 416 с.
57. de Kort W. DevOps on the Microsoft Stack /W.de Kort. – Apress, 2016.–296 p.
58. Ambler S. Introduction to Disciplined Agile Delivery. 2nd Ed. / S. Ambler – 2012. – 543 p.
59. Сайт С. Амблера [Электронный ресурс] – Режим доступа: <http://www.ambysoft.com/>.
60. Книберг Х. Scrum и XP: заметки с передовой. Второе издание / Х.Книберг – 2017. – 94 с. [Электронный ресурс] – Режим доступа: http://scrum.org.ua/wp-content/uploads/2008/12/scrum_xp-from-the-trenches-rus-final.pdf.
61. Reddy A. The Scrumban [r]evolution: getting the most out of Agile, Scrum and lean Kanban / A. Reddy – Addison-Wesley Professional, 2015. – 337 p.

62. Anderson D.J. Lessons In Agile Management: On the Road to Kanban / D.J. Anderson – Blue Hole Press, USA, 2012 – 408 p.
63. Поппендик М. Бережливое производство программного обеспечения. От идеи до прибыли / М. Поппендик, Т. Поппендик – М.; СПб.; К.: Вильямс, 2010. – 253 с.