

СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

**122 «Комп'ютерні науки»
КН-18**

2019 / 2020 навчальний рік

Технології # 2

1. Процеси життєвого циклу ПЗ.
2. Роботи процесу розробки.
3. Каскадна модель життєвого циклу ПЗ.
4. Ітеративна модель життєвого циклу ПЗ.
5. Циклічна модель життєвого циклу ПЗ.

https://github.com/eabshkvprof/2020_Mod_Prog_Techn

ISO/IEC 12207

Група стандартів **ISO/IEC 12207**: «System and software engineering – Software life cycle processes».

ДСТУ **ISO/IEC 12207:2016** «Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення»

ГОСТ Р ИСО/МЭК 12207-99 Процессы жизненного цикла программных средств.

http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71824

[ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств](#)

[ГОСТ Р ИСО/МЭК ТО 15271-2002 Руководство по применению ГОСТ Р ИСО/МЭК 12207](#)

ISO/IEC 12207. Термінологія

Система (system): комплекс, що складається з процесів, технічних і програмних засобів, пристроїв і персоналу, що володіє можливістю задовольняти встановленим потребам або цілям.

Програмно-апаратний засіб (firmware): поєднання технічних пристроїв і машинних команд або використовуваних обчислювальною машиною даних, що зберігаються на технічному пристрої у вигляді постійного програмного засобу.

ISO/IEC 12207. Термінологія

Програмний продукт (software product): набір машинних програм, процедур і, можливо, пов'язаних з ними документації і даних.

Програмна послуга (software servise): виконання робіт, завдань чи обов'язків, пов'язаних з програмним продуктом (розробка, супровід або експлуатація).

Програмний модуль (software unit): Окремо компілюємо частина програмного коду (програми).

ISO/IEC 12207. Термінологія

Замовлення (acquisition): процес придбання системи, програмного продукту або програмної послуги.

Замовник (acquirer): організація, яка набуває або отримує систему, програмний продукт або програмну послугу від постачальника.

Постачальник (supplier): організація, яка укладає договір із замовником на поставку системи, програмного продукту або програмної послуги.

Користувач (user) : особа чи організація, яка використовує діючу систему для виконання конкретної функції (Користувач → інші ролі: замовник, розробник, супроводжувач).

ISO/IEC 12207. Термінологія

Технічне завдання (statement of work): документ, який використовується замовником в як засіб для опису і визначення завдань, які виконуються при реалізації договору.

Випуск (release): конкретна версія елемента конфігурації, яка доступна для реалізації конкретної мети (наприклад, тестований випуск).

Версія (version): певний екземпляр об'єкту.

ISO/IEC 12207. Термінологія

Модель життєвого циклу (life cycle model):

структура, що складається з процесів, робіт і завдань, що включають в себе розробку, експлуатацію та супровід програмного продукту, що охоплює життя системи від встановлення вимог до неї до припинення її використання.

Процес (process): набір взаємопов'язаних робіт, які перетворюють вихідні дані у вихідні результати.

ISO/IEC 12207. Процеси

1. Основні процеси

1.1. Замовлення

1.2. Поставка

1.3. Розробка

1.4. Експлуатація

1.5. Супровід

3. Організаційні процеси

3.1. Керування

3.2. Створення інфраструктури

3.3. Вдосконалення

3.4. Навчання

2. Допоміжні процеси

2.1. Документування

2.2. Управління конфігурацією

2.3. Забезпечення якості

2.4. Верифікація

2.5. Атесатція

2.6. Сумісний аналіз

2.7. Аудит

2.8. Рішення проблем

ISO/IEC 12207. Основні процеси

Процеси, які реалізуються під управлінням основних сторін (замовник, постачальник, розробник, оператор, персонал супроводу), залучених в життєвий цикл програмних засобів.

1.1. **Замовлення:** роботи замовника - організації, яка набуває систему, програмний продукт або програмну послугу.

1.2. **Поставка:** роботи постачальника, тобто організації, яка поставляє систему, програмний продукт або програмну послугу замовнику.

ISO/IEC 12207. Основні процеси

1.3. Розробка: роботи розробника, тобто організації, яка **проектує і розробляє програмний продукт.**

1.4. Експлуатація: роботи оператора, тобто організації, яка забезпечує експлуатаційне обслуговування системи в заданих умовах в інтересах користувачів.

1.5. Супровід: роботи персоналу супроводу, тобто організації, яка надає послуги з супроводу програмного продукту (контрольовані зміни програмного продукту з метою збереження його початкового стану і функціональних можливостей).

ISO/IEC 12207. Допоміжні процеси

Допоміжний процес є цілеспрямованою **складовою частиною основного процесу**, забезпечує успішну реалізацію та якість виконання програмного проекту.

2.1. Документування: роботи по опису інформації, яка видається в процесі життєвого циклу.

2.2. Управління конфігурацією: роботи по управлінню конфігурацією системи.

2.3. Забезпечення якості: роботи за об'єктивним забезпечення того, щоб програмні продукти і процеси відповідали вимогам, встановленим для них, і реалізовувалися в рамках затверджених планів.

ISO/IEC 12207. Допоміжні процеси

2.4. **Верифікації:** роботи (замовника, постачальника або незалежної сторони) по верифікації програмних продуктів у міру реалізації програмного проекту.

2.5. **Атестація:** роботи (замовника, постачальника або незалежної сторони) по атестації програмних продуктів програмного проекту.

2.6. **Спільний аналіз:** роботи з оцінки стану і результатів будь-якої роботи. Даний процес може використовуватися двома будь-якими сторонами, коли одна зі сторін (що перевіряє) перевіряє іншу сторону (перевіряється) на спільній нараді.

ISO/IEC 12207. Допоміжні процеси

2.7. **Аудит:** роботи по визначенню відповідності вимогам, планам і договорам. Даний процес може використовуватися двома сторонами, коли одна зі сторін (що перевіряє) контролює програмні продукти або роботи іншого боку (перевіряється).

2.8. **Вирішення проблем:** процес аналізу та усунення проблем (включаючи невідповідності), незалежно від їх характеру і джерела, які були виявлені під час здійснення розробки, експлуатації, супроводу або інших процесів.

ISO/IEC 12207. Організаційні процеси

Процеси створення і реалізації основної структури, які охоплюють взаємопов'язані процеси життєвого циклу і відповідний персонал, а також для постійного вдосконалення даної структури і процесів.

3.1. Управління: Визначає основні роботи з управління, включаючи управління проектом, при реалізації процесів життєвого циклу.

3.2. Створення інфраструктури: роботи по створення основної структури процесу життєвого циклу.

ISO/IEC 12207. Організаційні процеси

3.3. Удосконалення: основні роботи, які організація (замовника, постачальника, розробника, оператора, персоналу супроводу або адміністратора іншого процесу) виконує при створенні, оцінці, контролі і удосконалення обраних процесів життєвого циклу.

3.4. Навчання: роботи за відповідним навчанням персоналу.

ISO/IEC 12207. 1.3. Розробка

1.3. Розробка

1.3.1. Підготовка процесу розробки

1.3.2. Аналіз вимог до системи

1.3.3. Проектування системної архітектури

1.3.4. Аналіз вимог до програмних засобів

1.3.5. Проектування програмної архітектури

1.3.6. Технічне проектування програмних засобів

1.3.7. Програмування та тестування програмних засобів

1.3.8. Збірка програмних засобів

1.3.9. Випробування програмних засобів

1.3.10. Збірка системи

1.3.11. Випробування системи

1.3.12. Забезпечення прийомки програмних засобів

ISO/IEC 12207. 1.3. Розробка

1.3.1. Підготовка процесу розробки: розробник повинен:

□ визначити чи вибрати модель життєвого циклу програмних засобів, відповідну області реалізації, величини та складності проекту. Повинні бути вибрані і структуровані в моделі життєвого циклу програмних засобів роботи і завдання процесу розробки. Повинен бути розроблено план проведення робіт.

□ !!!Документальне оформлення (2.1.)

ISO/IEC 12207. 1.3. Розробка

1.3.2. Аналіз вимог до системи.

Розробник повинен

□ виконати аналіз сфери застосування розроблюваної системи з точки зору визначення вимог до неї. Технічні вимоги до системи повинні охоплювати:

- функції і можливості системи;
- комерційні та організаційні вимоги;
- вимоги користувача;
- вимоги безпеки і захисту;
- ергономічні вимоги;
- вимоги до інтерфейсів;
- експлуатаційні вимоги;
- вимоги до супроводу;
- проектні обмеження та кваліфікаційні вимоги.

□ Технічні вимоги до системи повинні бути документально оформлені.

ISO/IEC 12207. 1.3. Розробка

1.3.3. **Проектування системної архітектури.**

Розробник повинен:

- ☐ Визначити загальну архітектуру системи (архітектура верхнього рівня).
- ☐ Вказати об'єкти технічних і програмних засобів і ручних операцій.
- ☐ Визначити та забезпечити розподіл всіх вимог до системи між об'єктами архітектури.
- ☐ Визначити об'єкти конфігурації технічних і програмних засобів і ручних операцій на основі об'єктів архітектури.
- ☐ Документально оформити прив'язку системної архітектури та вимог до системи щодо встановлених об'єктів.

ISO/IEC 12207. 1.3. Розробка

1.3.4. Аналіз вимог до програмних засобів :

Розробник повинен:

□ Встановити і документально оформити такі вимоги до програмних засобів, включаючи технічні вимоги до характеристик якості :

- функціональні та технічні вимоги, включаючи продуктивність, фізичні характеристики і навколишні умови, під які повинен бути створений програмний об'єкт архітектури (далі - програмний об'єкт);
- вимоги до зовнішніх інтерфейсів програмного об'єкта архітектури;
- кваліфікаційні вимоги;
- вимоги захисту, включаючи вимоги, що відносяться до допустимої точності інформації;

ISO/IEC 12207. 1.3. Розробка

1.3.4. Аналіз вимог до програмних засобів :

- вимоги безпеки, включаючи вимоги, що стосуються способів експлуатації та супроводу, впливу навколишнього середовища і травмобезпечності персоналу;
- ергономічні вимоги, включаючи вимоги, що відносяться до ручних операцій, взаємодії "людина-машина", персоналу і областям, які вимагають концентрації уваги людини, пов'язаних з чутливістю об'єкта до помилок людини і навченості персоналу;
- вимоги до визначення даних і бази даних;
- вимоги щодо введення в дію та приймання поставленого програмного продукту на об'єктах експлуатації та супроводу;
- вимоги до документації користувача;
- вимоги до експлуатації об'єкта користувачем;
- вимоги до обслуговування користувача.

ISO/IEC 12207. 1.3. Розробка

1.3.4. Аналіз вимог до програмних засобів :

□ Оцінити вимоги до програмних засобів за такими критеріям (при цьому результати оцінок повинні бути документально оформлені):

- врахування вимог до системи і проекту системи;
- зовнішня узгодженість з вимогами до системи;
- внутрішня узгодженість вимог до об'єктів між собою;
- можливість тестування вимог;
- здійсненність програмного проекту;
- можливість експлуатації і супроводу.

ISO/IEC 12207. 1.3. Розробка

1.3.5. **Проектування програмної архітектури:**

Розробник повинен:

□ Трансформувати вимоги до програмного об'єкту в архітектуру, яка описує загальну структуру об'єкта і визначає компоненти вимог до програмному об'єкту.

□ Забезпечити розподіл всіх вимог до програмному об'єкту між його компонентами і подальше їх уточнення з точки зору полегшення технічного проектування. Архітектура програмного об'єкта повинна бути **документально оформлена**.

□ Розробити і документально оформити загальні (ескізні) проекти зовнішніх інтерфейсів програмного об'єкта, інтерфейсів між компонентами та баз даних.

ISO/IEC 12207. 1.3. Розробка

1.3.5. **Проектування програмної архітектури:**

□ Розробити і **документально оформити** попередні версії документації користувача.

□ Визначити попередні загальні вимоги до випробувань (програмного об'єкта і графік збірки програмного продукту).

□ Оцінити архітектуру програмного об'єкта і ескізні проекти:

- врахування вимог до програмного об'єкту;
- зовнішня узгодженість з вимогами до програмного об'єкту;
- внутрішня узгодженість між компонентами програмного об'єкта;
- відповідність методів проектування і використовуваних стандартів;
- e) можливість технічного проектування;
- f) можливість експлуатації і супроводу.

ISO/IEC 12207. 1.3. Розробка

1.3.6. Технічне проектування програмних засобів.

Розробник повинен:

□ Розробити технічний проект для кожного компонента програмного об'єкта.

- Компоненти програмного об'єкта повинні бути уточнені на рівні програмних модулів, які можна програмувати (кодувати), компілювати і тестувати незалежно. Повинно бути забезпечено розподіл технічних вимог до компонентів програмного об'єкта між програмними модулями.

□ Розробити і документально оформити технічні проекти

- зовнішніх інтерфейсів програмного об'єкта;
- інтерфейсів між компонентами програмного об'єкта і між програмними модулями, баз даних;
- технічні проекти інтерфейсів повинені забезпечити виконання програмування без потреби у додатковій інформації.

ISO/IEC 12207. 1.3. Розробка

1.3.7. Програмування та тестування програмних засобів. Розробник повинен:

□ Розробити і документально оформити такі продукти:

- кожен програмний модуль і базу даних;
- процедури випробувань (тестування) та дані для тестування кожного програмного модуля і бази даних.

□ Протестувати кожен програмний модуль і базу даних, гарантуючи, що вони відповідають встановленим вимогам. Результати тестування повинні бути документально оформлені.

□ Уточнити документацію користувача (при необхідності).

□ Уточнити загальні вимоги до тестування і програму збірки програмних засобів.

ISO/IEC 12207. 1.3. Розробка

1.3.7. Програмування та тестування програмних засобів. Розробник повинен:

□ Оцінити запрограмовані елементи програмного об'єкта і результати їх тестування за наступними критеріями (при цьому результати оцінок повинні бути документально оформлені):

- врахування вимог до програмного об'єкту і проекту об'єкта в цілому;
- зовнішнє відповідність вимогам і проекту програмного об'єкта;
- внутрішнє відповідність між вимогами до програмних модулів;
- тестове покриття всіх модулів;
- відповідність методів програмування і використовуваних для них стандартів;
- можливість побудови та тестування;
- можливість експлуатації і супроводу.

ISO/IEC 12207. 1.3. Розробка

1.3.8. **Збірка програмних засобів.** Розробник повинен:

- ❑ Розробити план збірки для об'єднання програмних модулів і компонентів в програмний об'єкт. План повинен включати вимоги до випробувань (тестування), процедури тестування, контрольні дані, обов'язки виконавця і програму випробувань.
- ❑ Зібрати програмні модулі і компоненти і протестувати їх як продукти, розроблені відповідно до плану збірки.
- ❑ Забезпечити, щоб кожна збірка задовольняла вимогам до програмного об'єкту і щоб програмний об'єкт був повністю зібраний в результаті даної роботи.
- ❑ Документально оформити результати складання і тестування

ISO/IEC 12207. 1.3. Розробка

1.3.8. Збірка програмних засобів.

Розробник повинен:

□ Розробити і документально оформити для кожної кваліфікаційної вимоги до програмного об'єкту - набір тестів, контрольних прикладів (вихідні і вихідні дані, критерії тестування), процедури випробувань для проведення кваліфікаційних випробувань програмних засобів.

□ Забезпечити, щоб зібраний програмний об'єкт був готовий до кваліфікаційних випробувань.

ISO/IEC 12207. 1.3. Розробка

1.3.9. Кваліфікаційні випробування програмних засобів. Розробник повинен:

- ☐ Проводити кваліфікаційні випробування (тестування) на відповідність кваліфікаційним вимогам до програмного об'єкту.
- ☐ Забезпечити, щоб реалізація кожного встановленого вимоги до програмному об'єкту була перевірена на відповідність.
- ☐ Результати кваліфікаційних випробувань повинні бути документально оформлені.
- ☐ Повинен уточнити документацію користувача (при необхідності).
- ☐ Оцінити проект, запрограмований програмний об'єкт, проведені випробування, результати випробувань і документацію користувача.

ISO/IEC 12207. 1.3. Розробка

1.3.10. **Збірка системи.** Розробник повинен:

□ Зібрати об'єкти програмної конфігурації в єдину систему разом з об'єктами технічної конфігурації, ручними операціями і, при необхідності, з іншими системами.

□ Зібрана система повинна бути випробувана на відповідність встановленим вимогам. Результати повинні бути документально оформлені.

□ Для кожного кваліфікаційного вимоги до системи розробити і документально оформити: склад випробувань і контрольних прикладів (вихідні та вихідні дані, критерії випробувань) та процедури проведення кваліфікаційних випробувань системи.

□ Забезпечити, щоб зібрана система була готова до кваліфікаційних випробувань.

ISO/IEC 12207. 1.3. Розробка

1.3.11. Кваліфікаційні випробування системи.

Розробник повинен:

- ☐ Провести випробування у відповідності з кваліфікаційними вимогами, встановленими до системи.
- ☐ Забезпечити, щоб реалізація кожного вимоги до системи була випробувана на відповідність встановленим значенням і щоб система була готова до постачання.
- ☐ Документально оформити результати кваліфікаційних випробувань.
- ☐ Оцінити систему за наступними критеріями
 - тестове покриття вимог до системи;
 - відповідність очікуваних результатів;
 - можливість експлуатації і супроводу.

ISO/IEC 12207. 1.3. Розробка

1.3.12. **Забезпечення приймання програмних засобів.** Розробник повинен:

- ☐ Забезпечити проведення замовником оцінки готовності до приймання і приймальним випробуванням програмного продукту з крахуванням кваліфікаційних випробувань програмного продукту і кваліфікаційних випробувань системи.
- ☐ Результати оцінок готовності до приймання та приймальних випробувань повинні бути документально оформлені.
- ☐ Укомплектувати і поставити програмний продукт замовнику, дотримуючись умов договору.
- ☐ Забезпечити первинне і безперервне навчання і підтримку персоналу замовника.

ISO/IEC 12207. 1.3. Розробка

1.3.2. Вимоги до системи

1.3.11. Випробування системи

1.3.3. Проектування системної архітектури

1.3.10. Збірка системи

„СІСТЕМНІ” РОБОТИ

1.3.4. Аналіз вимог до ПЗ

1.3.9. Випробування програмних засобів

„ПРОГРАМНІ”
РОБОТИ

1.3.5. Проектування програмної архітектури

1.3.8. Збірка програмних засобів

1.3.6. Технічне проектування програмних засобів

1.3.7. ПРОГРАМУВАННЯ та
ТЕСТУВАННЯ

ISO/IEC 12207. 1.3. Розробка



ISO/IEC 12207. 1.3.7. Життєвий цикл

Модель життєвого циклу (life cycle model):

структура, що складається з процесів, робіт і завдань, що включають в себе розробку, експлуатацію та супровід програмного продукту, що охоплює життя системи від встановлення вимог до неї до припинення її використання.

Виділяють наступні процеси:

- Замовлення, поставка
- Аналіз
- Проектування
- Програмування
- Тестування
- Ввід в дію, експлуатація

ISO/IEC 12207. 1.3.7. Життєвий цикл

Каскадна модель

передбачає
послідовне
виконання всіх
етапів проекту в
строго
фіксованому
порядку.

Перехід на наступний етап
означає повне завершення
робіт на попередньому
етапі.



ISO/IEC 12207. 1.3.7. Життєвий цикл

Каскадна модель

- фіксований набір стадій;
- кожна стадія закінчується документованим результатом;
- наступна стадія починається лише після закінчення попередньої.

Недоліки:

- негнучкість;
- фаза повинна бути завершена до переходу до наступної;
- набір фаз фіксований;
- **! НЕМОЖЛИВО (важко) реагувати на зміни вимог.**

ISO/IEC 12207. 1.3.7. Життєвий цикл

Ітеративна (інкрементна) модель надає можливість повернень на попередні етапи.



Мета кожної ітерації – отримання працюючої версії програмної системи, що включає функціональність, визначену інтегрованим змістом усіх попередніх і поточної ітерації. На кожній ітерації продукт інкрементально нарощує функціональність. Результат фінальної ітерації → необхідний продукт.

ISO/IEC 12207. 1.3.7. Життєвий цикл

Ітеративна (інкрементна) модель

- Стадії повторюються неодноразово;

Недоліки:

- система часто погано структурована, проект «не прозорий»;
- після уточнення вимог відкидається частина раніше виконаної роботи;
- потрібні засоби для швидкого розроблення.

ISO/IEC 12207. 1.3.7. Життєвий цикл

Спіральна (еволюційна) модель: серія послідовних ітерацій.



ISO/IEC 12207. 1.3.7. Життєвий цикл

Спиральна модель:

- проект має контрольні точки – milestones;
- на кожному витку спіралі створюється прототип. На виході – продукт, що відповідає вимогам користувачів.

Переваги:

- ранній аналіз можливостей повторного використання;
- наявні механізми досягнення параметрів якості;
- модель дозволяє контролювати джерела проектних робіт і відповідних витрат;
- модель дозволяє вирішувати інтегровані завдання системного розроблення, які охоплюють програмну і апаратну складові створюваного продукту.

Недоліки:

- після уточнення вимог відкидається частина раніше виконаної роботи;
- потрібні засоби для швидкого розроблення.

**ПРОГРАМУВАННЯ НЕ Є ОСНОВНИМ ВИДОМ
ДІЯЛЬНОСТІ ПРИ СТВОРЕННІ ПРОМИСЛОВОЇ
СИСТЕМИ (ПРОГРАМНОГО ПРОДУКТУ)**

**ПРОГРАМУВАННЯ \approx 15-20% ЧАСУ
РОЗРОБКИ**

Рекомендована ЛІТЕРАТУРА

- **Томашевський О.М., Цегелік Г.Г., Вітер М.Б., Дудук В.Ш. Інформаційні технології та моделювання бізнес-процесів. Навч. посіб. – К.: «Видавництво «Центр учбової літератури», 2012. – 296 с.**
- **Карпенко М.Ю., Манакова Н.О., Гавриленко І.О. Технології створення програмних продуктів та інформаційних систем. Навч. посіб. - Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім.О.М. Бекетова, 2017. – 93 с.**
- **Алексенко О.В. Технології програмування та створення програмних продуктів. Конспект лекцій. – Суми, Сумський державний університет, 2013. – 133с.**

Рекомендована ЛІТЕРАТУРА

- **Иванова Г.С. Технология программирования.** Учебник. – М.: «Кнорус», 2018. – 336 с.

Посилання

- https://web.posibnyky.vntu.edu.ua/fksa/4bevez_proektuvannya_programnyh_zasobiv_system_upravlinnya/mr.htm
- http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71824
- [ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств](#)
- [ГОСТ Р ИСО/МЭК ТО 15271-2002 Руководство по применению ГОСТ Р ИСО/МЭК 12207](#)

Контрольні запитання

- Надайте визначення основних, допоміжних та організаційних процесів життєвого циклу ПЗ та перелік відповідних робіт.
- Надайте перелік та поясніть сутність робіт, які розробник повинен виконати на етапі аналізу вимог до системи та ПЗ.
- Надайте перелік та поясніть сутність робіт, які розробник повинен виконати на етапі проектування архітектури та технічного проектування ПЗ.
- Надайте перелік та поясніть сутність робіт, які розробник повинен виконати на етапі програмування та тестування ПЗ.

Контрольні запитання

- Надайте перелік та поясніть сутність робіт, які розробник повинен виконати на етапі збірки та випробувань ПЗ та системи.
- Поясніть сутність каскадної моделі життєвого циклу ПЗ, надайте переваги та недоліки моделі.
- Поясніть сутність ітеративної моделі життєвого циклу ПЗ, надайте переваги та недоліки моделі.
- Поясніть сутність спіральної моделі життєвого циклу ПЗ, надайте переваги та недоліки моделі.

The END
Mod 2. Lec 2.