

ОСНОВИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лекція № 4 Процеси і технології створення ПЗ

1.1 Життєвий цикл ПЗ і його процеси. Поняття про моделі життєвого циклу.

1.2 Базові методології створення ПЗ.

1.3 Процеси створення програмних продуктів невеликої складності.

1.1 ЖИТТЄВИЙ ЦИКЛ ПЗ І ЙОГО ПРОЦЕСИ. ПОНЯТТЯ ПРО МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ.

Технології розробки ПЗ: ключові поняття

Технологія розробки програмного забезпечення - це система інженерних принципів для створення економічного ПО з заданими характеристиками якості. Будь-яка технологія розробки ПЗ базується на деякій методології або сукупності методологій.

Під **методологією** розуміється система принципів і способів організації процесу розробки програмних засобів. Мета методології розробки ПЗ - впровадження методів розробки ПП, які забезпечують досягнення заданих характеристик якості. В даний час широку популярність набули два базових принципа розробки ПП: **модульний** і **об'єктно-орієнтований**.

Розробка модульних ПП ґрунтується на використанні **структурних методів проектування**, метою яких є розбиття за деякими правилами проектованого програмного засобу на структурні компоненти. До структурних методів проектування відносяться такі класичні методи, як **структурне програмування**, **проектування «зверху-вниз»** і **«знизу-вверх»**, **метод розширення ядра**, а також ряд сучасних методів і методологій розробки ПЗ.

Об'єктно-орієнтована розробка базується на застосуванні об'єктно-проектних методів, до яких відносяться методології об'єктно-орієнтованого аналізу, проектування і програмування.

ЖИТТЄВИЙ ЦИКЛ І ЙОГО ПРОЦЕСИ

Відповідно до стандарту *ISO / IEC 12207 2008* - Системна і програмна інженерія - Процеси життєвого циклу програмних засобів

Життєвим циклом (ЖЦ) програмного продукту або системи є сукупність процесів, робіт і завдань, яка включає в себе розробку, експлуатацію та супровід програмного засобу або системи і охоплює їхнє життя від формулювання концепції до утилізації ПП. ЖЦ ПП складається з процесів. Кожен процес ЖЦ включає набір робіт. Кожна робота поділяється на набір задач.

З поняттям ЖЦ ПС і систем тісно пов'язане поняття **моделі ЖЦ**.

Модель життєвого циклу - це формалізована, абстрактна конструкція, яка відображає взаємозв'язок і послідовність виконання процесів ЖЦ і їх основних складових.

Процеси ЖЦ ПП діляться на три групи:

основні;

допоміжні;

організаційні.

Основні процеси ЖЦ

Основні процеси життєвого циклу - це процеси, які реалізуються під управлінням основних сторін, які беруть участь в життєвому циклі програмних засобів. Основними сторонами є замовник, постачальник, розробники, користувачі і персонал супроводу програмних продуктів. До основних процесів відноситься п'ять процесів:

замовлення;

поставка;

розробка;

експлуатація;

супровід.

Процеси розробки

Процес розробки складається з робіт і завдань, які виконуються розробником. Даний процес містить тринадцять робіт:

- 1) підготовка процесу розробки;
- 2) аналіз вимог до системи;
- 3) проектування системної архітектури;
- 4) аналіз вимог до програмних засобів;
- 5) проектування програмної архітектури;
- 6) технічне проектування програмних засобів;
- 7) програмування і тестування програмних засобів;
- 8) збірка програмних засобів;
- 9) кваліфікаційні випробування програмних засобів;
- 10) збірка системи;
- 11) кваліфікаційні випробування системи;
- 12) введення в дію програмних засобів;
- 13) приймання програмних засобів.

Допоміжні і організаційні процеси ЖЦ

Допоміжні процеси життєвого циклу - це процеси, які є цілеспрямованими складовими частинами інших процесів і призначені для забезпечення умов успішної реалізації і якості виконання програмного проекту. До допоміжних процесів відноситься вісім процесів:

документування;

управління конфігурацією;

забезпечення якості;

верифікація;

атестація;

спільний аналіз;

аудит;

вирішення проблем.

Організаційні процеси життєвого циклу - це процеси, призначені для створення в деякій організації та вдосконалення організаційних структур, що охоплюють процеси життєвого циклу і відповідний персонал.

До організаційних процесів відноситься чотири процесу:

управління;

- *управлінські;*

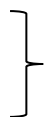
створення інфраструктури;

- *інфраструктурні*

удосконалення;

- *процеси розвитку*

навчання.



1.2 Базові методології створення ПЗ.

Методології створення ПЗ

На початковому етапі розвитку обчислювальної техніки ПЗ розроблялися за принципом «кодування - усунення помилок»

На сучасності етапі широко використовуються три базові методології розробки ПО:

- *каскадна;*
- *інкрементна;*
- *еволюційна.*

Існують тісні взаємозв'язки між *моделлю ЖЦ*, обраної при реалізації процесу розробки ПС, використовуваними методологіями і *технологіями* розробки ПС і рівнем якості розробленого програмного продукту.

Каскадна методологія

Каскадна методологія являє собою одноразовий прохід всіх етапів розробки. Дана методологія базується на повному визначенні всіх вимог до програмного засобу або системи на початку процесу розробки. Кожен етап розробки починається лише після завершення попереднього етапу. Повернення до вже виконаних етапів не передбачається. проміжні продукти розробки в якості версії програмного засобу не розглядаються.

Використання даної методології найбільш ефективно в наступних випадках:

- 1) при розробці проектів з чіткими, незмінними протягом ЖЦ вимогами і зрозумілою реалізацією;
- 2) при розробці проектів невисокої складності;.
- 3) при виконанні великих проектів в якості складової частини моделей ЖЦ, що реалізують інші методології розробки.

Інкрементна методологія

Інкрементна методологія являє собою багаторазовий прохід етапів розробки із запланованим поліпшенням результату.

Дана методологія базується на повному визначенні всіх вимог до програмного засобу (системи) на початку процесу розробки. Однак повний набір вимог реалізується поступово відповідно до плану в послідовних циклах розробки.

Результат кожного циклу називається *інкрементом*.

Перший інкремент реалізує базові функції програмного засобу. В подальших інкрементах функції програмного засобу поступово розширюються, поки не буде реалізований весь набір вимог. Відмінності між інкрементами сусідніх циклів в ході розробки поступово зменшуються. Результат кожного циклу розробки може поширюватися як чергова версія поставки програмного засобу або системи.

Використання даної стратегії найбільш ефективно в наступних випадках:

- 1) при розробці проектів, в яких більшість вимог можна сформулювати заздалегідь, але частина з них можуть бути уточнені через визначений період часу;
- 2) при розробці складних проектів з заздалегідь сформульованими вимогами; для них розробка системи або програмного засобу за один цикл пов'язана з великими труднощами;
- 3) при необхідності швидко поставити на ринок продукт, що має базові функціональні властивості;
- 4) при розробці проектів з низькою або середнім ступенем ризиків;
- 5) при виконанні проекту з застосуванням нових технологій.

Еволюційна методологія

Еволюційна методологія являє собою багаторазовий прохід етапів розробки. Дана методологія базується на частковому визначенні вимог щодо розроблюваного програмного засобу або системи на початку процесу розробки. Вимоги поступово уточнюються в послідовних циклах розробки. Результат кожного циклу розробки зазвичай являє собою чергову версію програмного засобу або системи.

Слід зазначити, що в загальному випадку для еволюційної методології характерним є істотно менша кількість циклів розробки при більшій тривалості циклу в порівнянні з інкрементною методологією. При цьому результат кожного циклу розробки (чергова версія програмного засобу значно відрізняється від результату попереднього циклу.

Г.2 Базові методології створення ПЗ.

Використання даної методології найбільш ефективно в наступних випадках:

- 1) при розробці проектів, для яких вимоги занадто складні, невідомі заздалегідь, непостійні або вимагають уточнення;
- 2) при розробці складних проектів, в тому числі:
 - великих довгострокових проектів;
 - проектів зі створення нових, які не мають аналогів ПЗ або систем;
 - проектів зі середнім і високим ступенем ризиків;
 - проектів, для яких потрібна перевірка концепції, демонстрація технічної здійсненності або проміжних продуктів;
- 3) при розробці проектів, що використовують нові технології.

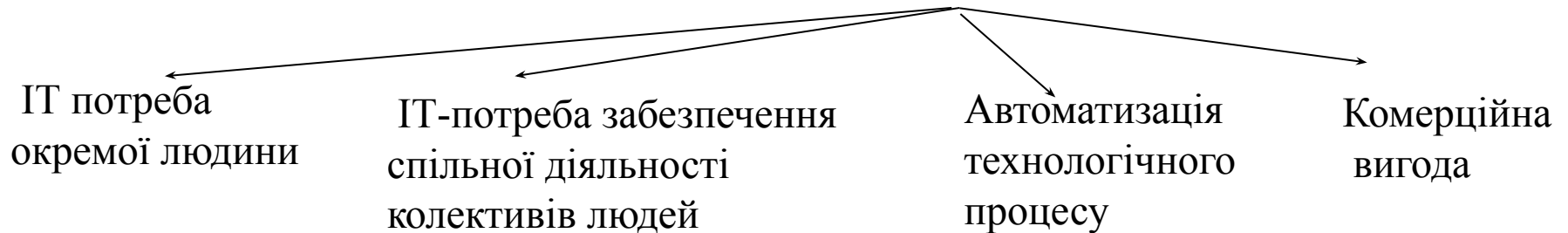
1.3 Процеси створення програмних продуктів невеликої складності: постановка задачі.

Аналіз передумов (контексту) розробки ПЗ

Ключові питання що формують передумови розробки ПЗ:

КП1. Що є причиною потреби в ПЗ?

Альтернативи рішень КП1



КП2. Спосіб задоволення ІТ-потреби?

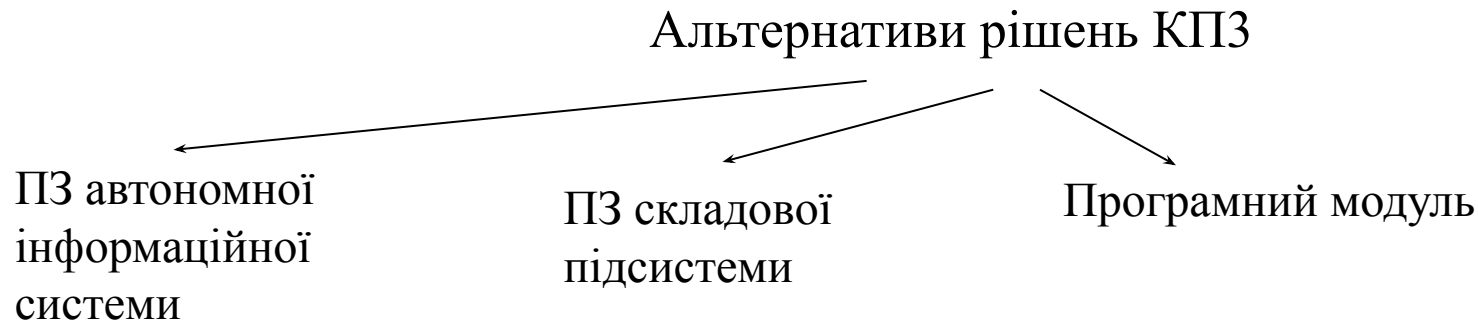
Альтернативи рішень КП2



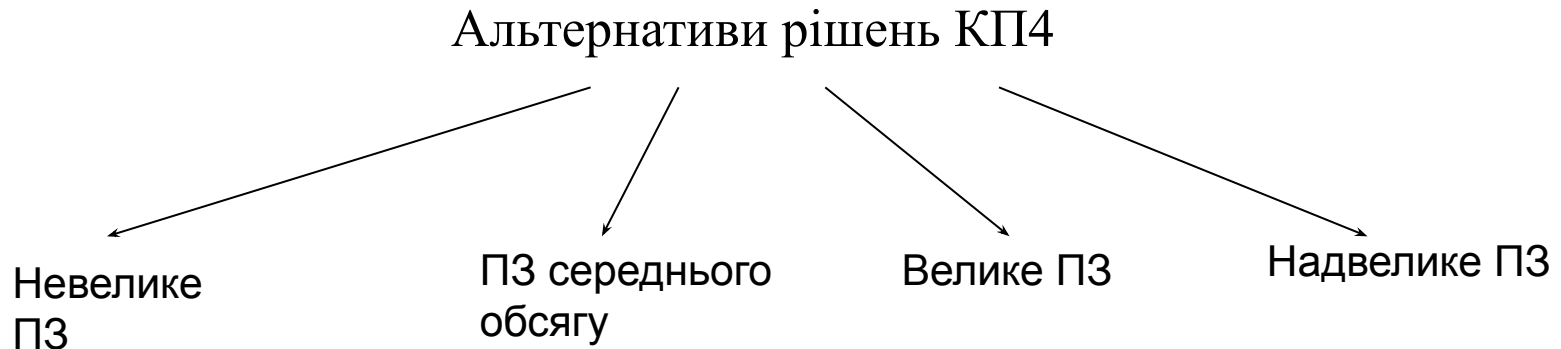
Аналіз передумов (контексту) розробки ПЗ

Ключові питання що формують передумови розробки ПЗ :

КПЗ. Як позиціонується програмний продукт, вимоги до якого розробляються, в ієрархії вкладених систем?



КП4. До якої категорії належить ПЗ за обсягом трикутника параметрів розробки: «час-ресурси-якість»?



Аналіз передумов (контексту) розробки ПЗ

Ключові питання що формують передумови розробки ПЗ :

КП5. Які можливості залучення до процесів створення ПЗ розробників?

Альтернативи рішень КП5

Індивідуальна розробка

Невеликі групи

Проектні команди

Софт-підприємства

КП6. Жорсткість вимог до якості і надійності?

Альтернативи рішень КП6

«М'які»
вимоги

Висока якість

Висока
надійність

«Жорсткі»
вимоги

КП7. Яким є рівень невизначеності щодо необхідних властивостей ПЗ?

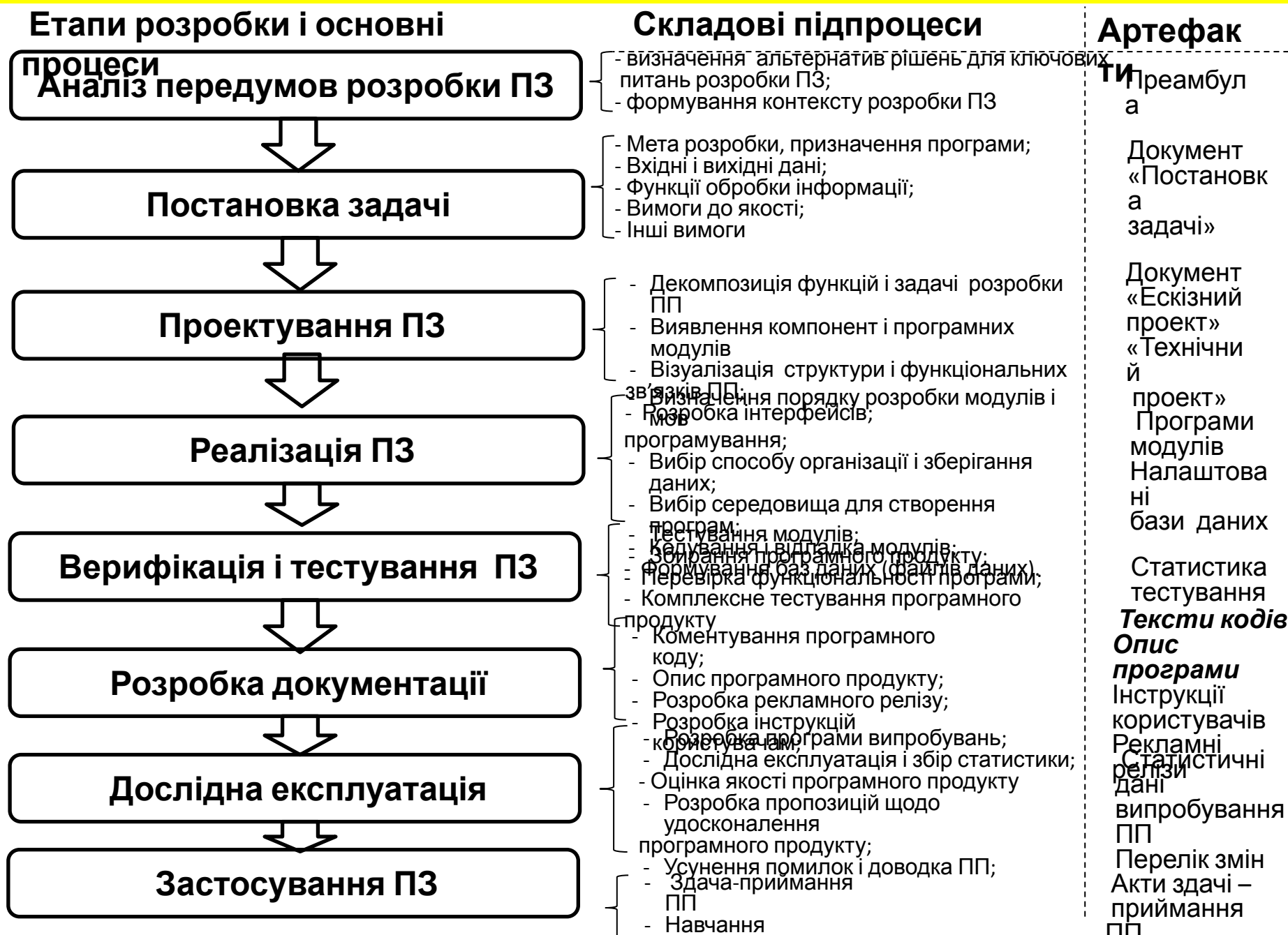
Альтернативи рішень КП7

Відсутня невизначеність

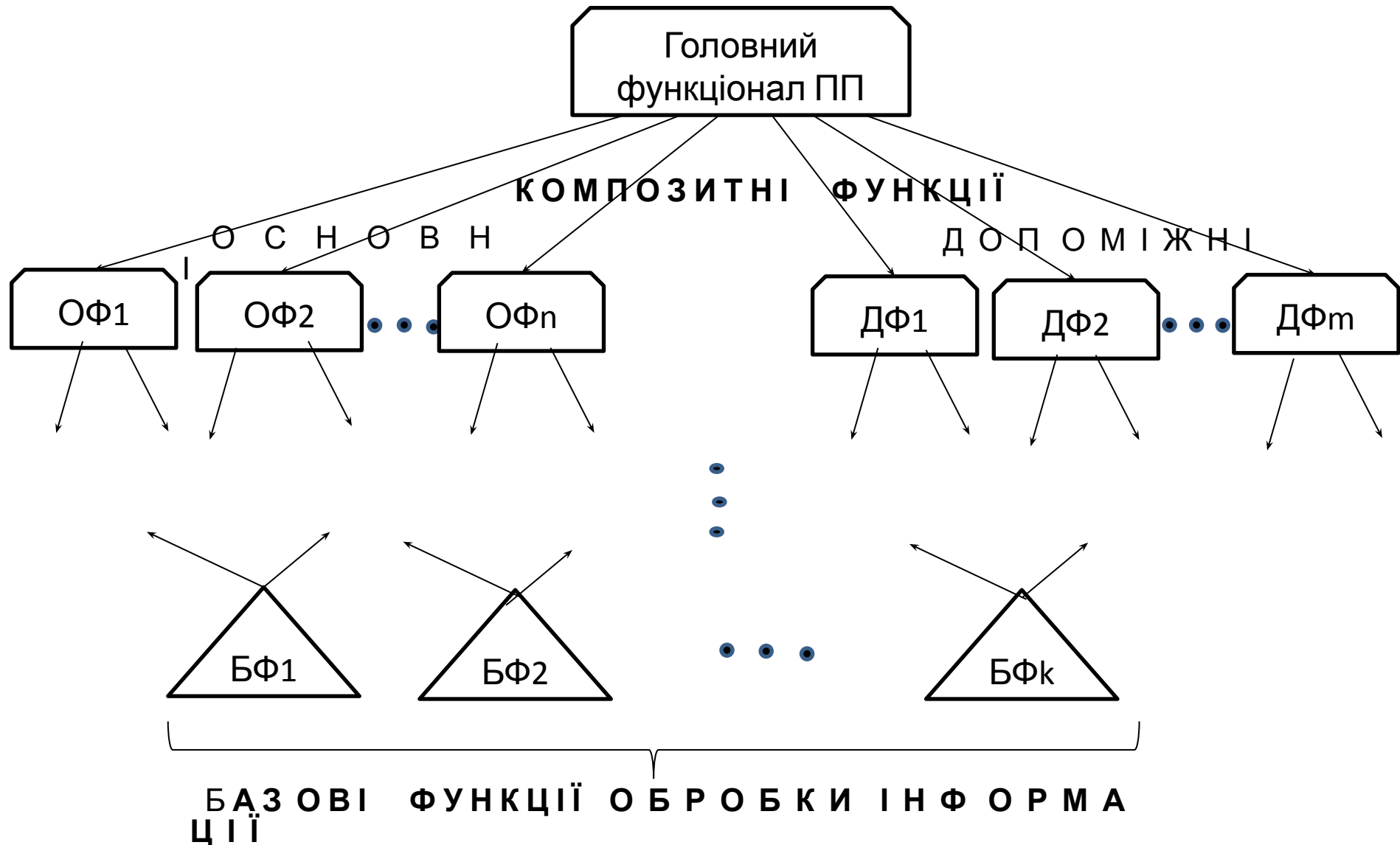
Невизначеність з другорядних вимог

Значна невизначеність, в т.ч. і з основних вимог

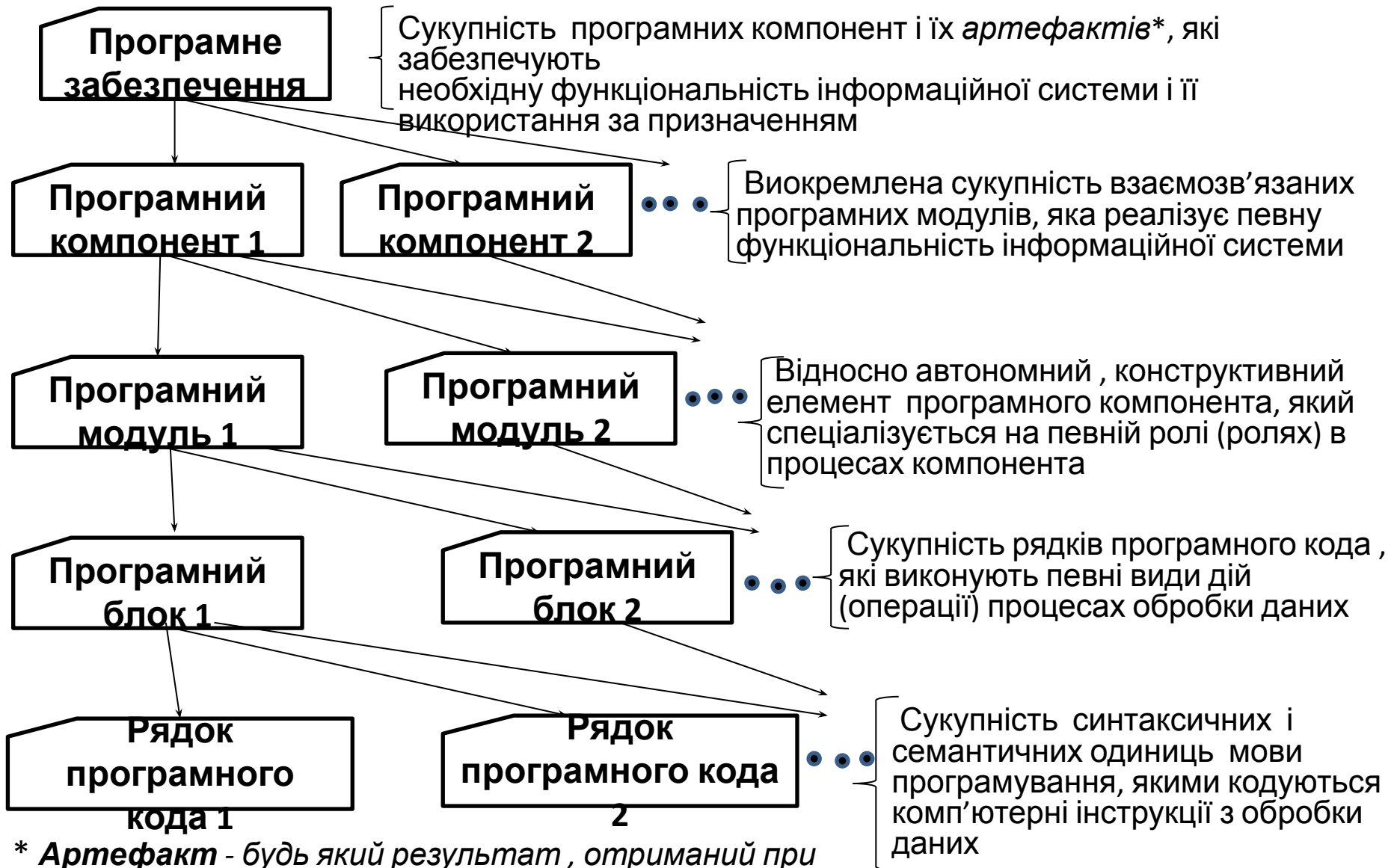
Базові процеси створення програмних продуктів невеликої складності ПП НС



Функціональна декомпозиція ПП



Ієрархія програмних засобів



* *Артефакт* - будь який результат, отриманий при виконанні процесів розробки ПЗ і який має споживчу цінність для процесів розробки і застосування програмних продуктів

Постановка задачі

Щоб приступити до вирішення задачі розробки ПЗ необхідно точно її сформулювати. В першу чергу, це означає визначення вхідних і вихідних даних, т. е. Відповіді на питання: а) що дано; б) що потрібно знайти. Подальша деталізація постановки задачі є відповіді на серію питань такого роду:

- як визначити рішення;
- яких даних не вистачає і чи всі вони потрібні;
- які зроблені припущення та т. п.

Таким чином, коротко можна сказати, що на етапі постановки задачі необхідно:

- опис вихідних даних і результату;
- формалізація завдання;
- опис поведінки програми в особливих випадках (якщо такі є).

В ході цієї роботи виявляються властивості, якими повинна володіти система в кінцевому вигляді (задум), описуються функції системи, характеристики інтерфейсу.

Документ «Постановка задачі»

Пропозиції в постановці завдання пишуться на природній мові в термінах зрозумілих і користувачеві і розробнику програмного забезпечення і повинні висловлювати однозначний сенс. Неправильне тлумачення пропозиції призводить до створення програмного продукту, який правильно вирішує невірні сформульоване завдання.

Документ «Постановка завдання» може містити розділи:

1. **Тема до програми.**
2. **Умова задачі.** Формулюється умова завдання, короткий опис програми, що розробляється і її призначення.
3. **Початок / закінчення роботи.** Вказується місяць і рік початку / закінчення розробки програмного продукту.
4. **Підстава для розробки.** Підставою для розробки програмного забезпечення може бути замовлення користувача, завдання адміністрації навчального закладу, контракт навчального закладу з іншою організацією та ін.
5. **Користувач (замовник).** Вказуються замовники програмного продукту, і пояснюється, чому він їм необхідний.
6. **Мета і призначення програми.**
7. **Основні вимоги.** Перераховуються вимоги користувача до розробляється програмного продукту. Тут же з точки зору користувача слід перерахувати функції програмного продукту.
8. **Вхідна інформація.** Описуються всі вхідні дані програмного забезпечення з точки зору їх змісту і призначення - звіти, файли, записи, поля даних, таблиці. Їх можливі джерела, носії і засоби відображення інформації і т.д.
9. **Вихідна інформація.** Описуються вихідні дані так само, як в пункті 9. Споживачі вихідних Даних.
10. **Зв'язок з іншими програмами**
11. **Вимоги до апаратного та програмного забезпечення.** Конфігурація апаратури і програмного забезпечення, в яких розробляється система може працювати і інші програмні продукти, від яких вона залежить.
12. **Зовнішні обмеження.**
13. **Ефективність.** Цілі продуктивності, такі, як тимчасові характеристики, пропускна здатність, використання ресурсів, а також необхідні засоби вимірювання продуктивності і засоби настройки.
14. **Безпека даних від несанкціонованого доступу.**
15. **Ергономічні характеристики.** Ергономічними характеристиками виробу є такі властивості, які забезпечують надійність, комфорт і продуктивність роботи користувачів і операторів. Ергономіка (грец.) - праця + закон - галузь знання, що вивчає трудові процеси з метою створення найкращих умов праці.
16. **Мобільність.** Описуються вимоги і цілі забезпечення перенесення програмного продукту з одних робочих умов в інші.
17. **Окупність капіталовкладень.** Визначається прибуток, яку дасть створення програмного продукту в поняттях, відповідних цільовим призначенням організації.
18. **Інші угоди сторін.**
19. **Термінологія.** Чітко визначається вся термінологія, яка може виявитися специфічної для даної розробки

Критерії оцінки якості і повноти документа «Постановка задачі»

Постановка задачі є коректною, якщо вона дає відповіді на питання:

- 1) «Чи існують методи або способи вирішення задання без використання комп'ютерної програми, чи потрібно її автоматизувати?»
- 2) «Якого позитивного результату слід очікувати від автоматизованого розв'язання задачі, що буде покращено, прискорене, оптимізовано, зекономлено при використанні комп'ютерної програми?»;
- 3) «Яке апаратне забезпечення необхідно для автоматизованого вирішення задачі, яка комп'ютерна платформа необхідна, яка операційна система і яке додаткове програмне забезпечення необхідно?»
- 4) «Що являють собою інформаційні процеси прикладної предметної області, де буде застосовуватися програма?»
- 5) «Який формат вихідних (вхідних) даних, які дані і в якій формі необхідні для розв'язання задачі?»
- 6) «Який формат проміжних і вихідних даних, які дані і в якій формі необхідно отримати?»
- 7) «Які функції обробки інформації повинна забезпечити програма?»
- 8) «Які режими обробки інформації повинна забезпечити програма?»
- 9) «Який інтерфейс користувача програми потрібно забезпечити, який інтерфейс з іншими програмами необхідний?»
- 10) «Якою повинна бути «глибина» опрацювання користувацької, інженерної, програмної і конструкторської документації, експлуатаційних документів, методичних посібників і керівництв по використанню програми, хто і як буде застосовувати результати роботи програми?».