



ЛЕКЦІЯ 11

Визначення вимог до програмного забезпечення і вихідних даних для його проектування

- **Етап постановки завдання** - один з найбільш відповідальних етапів створення програмного продукту.
- На цьому етапі формулюють основні вимоги до розробляється програмного забезпечення.
- Від того, наскільки повно визначені функції та експлуатаційні вимоги, наскільки правильно прийняті принципові рішення, що визначають процес проектування, багато в чому залежить вартість розробки і її якість.

Класифікація програмних продуктів за призначенням



ОСНОВНІ ЕКСПЛУАТАЦІЙНІ ВИМОГИ ДО ПРОГРАМНИХ ПРОДУКТІВ

- Експлуатаційні вимоги визначають деякі характеристики розроблюваного програмного забезпечення, що виявляються в процесі його функціонування.
- До таких характеристик відносять:
- **правильність** - функціонування відповідно до технічного завдання;
- **універсальність** - забезпечення належного функціонування при будь-яких припустимих даних і захисту від неправильних даних;
- **надійність (перешкодозахищеність)** - забезпечення повної повторюваності результатів, тобто забезпечення їх правильності при наявності різного роду збоїв;

- **перевіряємість** - можливість перевірки отриманих результатів;
- **точність результатів** - забезпечення похибки результатів не вище заданої;
- **захищеність** - забезпечення конфіденційності інформації;
- **програмна сумісність** - можливість спільного функціонування з іншим програмним забезпеченням;
- **апаратна сумісність** - можливість спільного функціонування з деяким обладнанням;

- **ефективність** - використання мінімально можливої кількості ресурсів технічних засобів, наприклад часу мікропроцесора або обсягу оперативної пам'яті;
- **здатність до адаптації** - можливість швидкої модифікації з метою пристосування до умов, що змінюються функціонування;
- **повторна входимість** - можливість повторного виконання без перезавантаження з диска;
- **ресентерабельним** - можливість «паралельного» використання декількома процесами.

ОСНОВНІ ЕКСПЛУАТАЦІЙНІ ВИМОГИ ДО ПРОГРАМНИХ ПРОДУКТІВ

- Правильність є обов'язковою вимогою для будь-якого програмного забезпечення: все, що зазначено в технічному завданні, неодмінно має бути реалізовано.
- Однак слід розуміти, що ні тестування, ні верифікація не доводять правильності створеного програмного продукту.
- У зв'язку з цим зазвичай говорять про певну ймовірність наявності помилок.
- Природно, чим більша відповідальність перекладається на комп'ютерну систему, тим менше повинна бути ймовірність як програмного, так і апаратного збою.
- Вимоги універсальності також зазвичай входить в групу обов'язкових. Нічого хорошого немає в тому, що розроблена система видасть результат для некоректних даних або аварійно завершує свою роботу на деяких наборах даних.

ОСНОВНІ ЕКСПЛУАТАЦІЙНІ ВИМОГИ ДО ПРОГРАМНИХ ПРОДУКТІВ

- Практично, чим вище вимоги до правильності і універсальності програмного забезпечення, тим вище і вимоги до його надійності. Джерелами перешкод можуть бути всі учасники обчислювального процесу:
 - технічні засоби (схильні до збоїв),
 - програмні засоби (може містити помилки),
 - люди (можуть помилятися при введенні вихідних даних).
- Сучасні обчислювальні пристрої вже досить надійні.
- Збої технічних засобів, як правило, реєструються апаратно, відповідно результати обчислень в цьому випадку відновлюються. У разі тривалих обчислень, як правило, проміжні результати зберігають (прийом отримав назву «створення контрольних точок»), що дозволяє при виникненні збою продовжити обчислення з даними, записаними в останній контрольній точці.

ОСНОВНІ ЕКСПЛУАТАЦІЙНІ ВИМОГИ ДО ПРОГРАМНИХ ПРОДУКТІВ

- Передача інформації по мережах також апаратно контролюється, крім того, зазвичай застосовується спеціальне заводозахисні кодування, яке дозволяє знаходити і виправляти помилки передачі даних.
- Однак повністю виключити помилки технічних засобів неможливо, тому в тих випадках, коли вимоги до надійності високі, зазвичай використовують дублювання систем, при якому дві системи вирішують одну і ту ж задачу паралельно, періодично звіряючи отримані результати.
- Часто самим «ненадійним елементом» сучасних систем є люди. Вже добре відомо, що в умовах монотонної роботи за пультом обчислювальної установки оператори допускають велику кількість помилок.
- Підвищені вимоги до надійності пред'являють при розробці систем управління, що функціонують в режимі реального часу, коли обчислення виконуються паралельно з технологічними процесами.

ОСНОВНІ ЕКСПЛУАТАЦІЙНІ ВИМОГИ ДО ПРОГРАМНИХ ПРОДУКТІВ

- Для забезпечення можливості перевірки слід документально **фіксувати вихідні дані**, встановлені режими та іншу інформацію, яка впливає на одержувані результати. Особливо це важливо у випадках, коли дані надходять безпосередньо від датчиків. Якщо такі дані не виводити разом з результатами, то останні не можна буде перевірити.
- Точність або величина похибки результатів залежить від точності вихідних даних, ступеня адекватності використовуваної моделі, точності обраного методу і похибки виконання операцій в комп'ютері.
- Вимоги до точності результатів зазвичай найбільш жорсткі для систем управління технологічними процесами (наприклад, хімічними) і систем навігації (наприклад, система управління стикуванням космічних апаратів).

- Забезпечення захищеності (конфіденційності) інформації, використовуваної проектованої системою, окрема і в умовах наявності мереж досить складне завдання.

Крім чисто програмних засобів захисту, таких як кодування інформації та ідентифікація користувача, для забезпечення захищеності використовують також спеціальні організаційні прийоми.

Найбільш жорсткі вимоги пред'являються до систем, в яких зберігається інформація, пов'язана з державною і комерційною таємницею.

- Вимога програмної сумісності може варіюватися від можливості спільної установки з зазначеним програмним забезпеченням до забезпечення взаємодії з ним, наприклад обміну даними і т. п.
- Вимога апаратної сумісності в основному формулюють у вигляді мінімально можливої конфігурації обладнання, на якому буде працювати програмне забезпечення.

ОСНОВНІ ЕКСПЛУАТАЦІЙНІ ВИМОГИ ДО ПРОГРАМНИХ ПРОДУКТІВ

- Ефективність системи зазвичай оцінюється окремо по кожному ресурсу обчислювальної установки.
Часто використовують такі критерії:
- час відповіді системи (зазвичай віднесене до швидкодії використовуваного обладнання) - для систем, що взаємодіють з користувачем в інтерактивному режимі, і систем реального часу;
- обсяг оперативної пам'яті - для продуктів, що працюють в системах з обмеженим обсягом оперативної пам'яті;
- обсяг зовнішньої пам'яті - для продуктів, що інтенсивно використовують зовнішню пам'ять, наприклад баз даних;
- кількість обслуговуваних зовнішніх пристроїв - для продуктів, які здійснюють інтенсивну взаємодію з зовнішніми пристроями, наприклад датчиками.

- Вимога повторної входимості зазвичай пред'являється до програмного забезпечення, що резидентно завантажується в оперативну пам'ять, наприклад драйверам.
Для забезпечення цієї вимоги необхідно так організувати програму, щоб ніякі її вихідні дані не затирились в процесі виконання або відновлювалися на початку або при завершенні кожного виклику.
- Вимога реєнтерабельним є більш жорстким, ніж повторна входимість, так як в цьому випадку вага дані, що змінюються програмою в процесі виконання, повинні бути виділені в спеціальний блок, копія якого створиться для кожного процесу при виклику програми.
- *Складність багатьох програмних систем не дозволяє відразу сформулювати чіткі вимоги до них. Зазвичай для переходу від ідеї створення деякого програмного забезпечення до чіткому формулюванню вимог, які можуть бути занесені в технічне завдання, необхідно виконати передпроектні дослідження в області розробки.*

- **Метою передпроектних досліджень** є перетворення загальних нечітких знань про призначення майбутнього програмного забезпечення в порівняно точні вимоги до нього.

- Існують два варіанти невизначеності:
 1. невідомі методи вирішення сформульованого завдання - такого типу невизначеності зазвичай виникають при вирішенні науково-технічних завдань;
 2. невідома структура автоматизованих інформаційних процесів - зазвичай зустрічається при побудові автоматизованих систем керування підприємствами.

- У першому випадку під час передпроектних досліджень визначають можливість вирішення поставленого завдання і методи, що дозволяють отримати необхідний результат, що може зажадати відповідних наукових досліджень як фундаментального, так і прикладного характеру.
- У другому випадку визначають:
 1. структуру та взаємозв'язки автоматизуються інформаційних процесів;
 2. розподіл функцій між людиною і системою, а також між апаратурою і програмним забезпеченням;
 3. функції програмного забезпечення; зовнішні умови його функціонування і особливості його інтерфейсів як з користувачами, так і при необхідності - з апаратною частиною;
 4. вимоги до програмних і інформаційних компонентів, необхідні апаратні ресурси, вимоги до баз даних і фізичні характеристики програмних компонент.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- На початкових етапах процесу проектування повинні бути прийняті принципові рішення, багато в чому визначають цей процес, а також якість і трудомісткість розробки.
- До таких рішень відносять:
 - вибір архітектури програмного забезпечення;
 - вибір типу призначеного для користувача інтерфейсу і технології роботи з документами;
 - вибір підходу до розробки (структурного або об'єктного);
 - вибір мови та середовища програмування.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- Вибір архітектури програмного забезпечення.
- **Архітектурою програмного забезпечення** називають сукупність базових концепцій (принципів) його побудови.
- Архітектура програмного забезпечення визначається складністю вирішуваних завдань, ступенем універсальності розроблюваного програмного забезпечення і числом користувачів, що одночасно працюють з однією його копією.
- Розрізняють:
 - **однокористувальницьку архітектуру**, при якій програмне забезпечення розраховане на одного користувача, що працює за персональним комп'ютером;
 - **багатокористувальницьку архітектуру**, яка розрахована на роботу в локальній або глобальній мережі.

- В рамках одного користувача архітектури розрізняють:
- **Програми** (Програмою називають адресований комп'ютера набір інструкцій, точно описує послідовність дій, які необхідно виконати для вирішення конкретного завдання.);
- **пакети програм** (Пакети програм являють собою сукупність програм, що вирішують завдання деякої прикладної області. Наприклад, пакет графічних програм, пакет математичних програм.);
- **програмні комплекси** (Програмні комплекси являють собою сукупність програм, спільно забезпечують вирішення невеликого класу складних завдань однієї прикладної області. Для вирішення такого завдання може знадобитися вирішити кілька підзадач, послідовно викликаючи програми комплекс);
- **програмні системи** (Програмні системи являють собою організовану сукупність програм (підсистем), що дозволяє вирішувати широкий клас задач з деякою прикладної області. Взаємодіють через загальні дані.).

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- Розраховані на багато користувачів програмні системи на відміну від звичайних програмних систем повинні організовувати мережеве взаємодія окремих компонентів програмного забезпечення, що ще ускладнює процес його розробки.
- Для розробки подібного програмного забезпечення використовують спеціальні технології або платформи, наприклад технології CORBA, COM, Java і т. П.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- Вибір типу користувальницького інтерфейсу. Розрізняють чотири типи користувальницьких інтерфейсів:
- **примітивні** - реалізують єдиний сценарій роботи, наприклад введення даних - обробка - виведення результатів;
- **меню** - реалізують безліч сценаріїв роботи, операції яких організувати в ієрархічні структури, наприклад «вставка»: «вставка файлу», «вставка символу» і т. д .;
- **з вільною навігацією** - реалізують безліч сценаріїв, операції яких не прив'язані до рівнів ієрархії, і припускають визначення безлічі можливих операцій на конкретному етапі роботи; інтерфейси даної форми в основному використовують Windows-додатки;
- **прямого маніпулювання** - реалізують безліч сценаріїв, представлених в операціях над об'єктами, основні операції ініціюються переміщенням піктограм об'єктів мишею, дана форма реалізована в інтерфейсі самої операційної системи Windows альтернативно інтерфейсу з вільною навігацією.

- ❑ **Вибір мови програмування.** У більшості випадків ніякої проблеми вибору мови програмування реально не існує. Мова може бути визначена:
 - ❑ організацією, яка здійснює розробку; наприклад, якщо фірма володіє ліцензійним варіантом C ++ Builder, то вона буде вести розробки переважно в даному середовищі;
 - ❑ програмістом, який по можливості завжди буде використовувати добре знайомий мову;
 - ❑ усталеною думкою («вага розробки подібного роду повинні виконуватися на C ++ або на Java або на ...») і т. п.
- ❑ *Якщо ж все-таки вибір мови реально можливий, то потрібно мати на увазі, що всі існуючі мови програмування можна розділити на слід групи:*
 - ❑ універсальні мови високого рівня;
 - ❑ спеціалізовані мови розробника програмного забезпечення;
 - ❑ спеціалізовані мови користувача;
 - ❑ мови низького рівня.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- У групі універсальних мов високого рівня лідером на сьогодні все ще є мова C (разом з C ++).
- Дійсно різні версії C і C ++ мають цілий ряд дуже істотних переваг:
 1. багатоплатформенність - для всіх використовуваних в даний час платформ існують компілятори з мови C і C ++;
 2. наявність операторів, що реалізують основні структурні алгоритмічні конструкції (умовну обробку, всі види циклів);
 3. можливість програмування на низькому (системному) рівні з використанням адрес оперативної пам'яті;
 4. величезні бібліотеки підпрограм і класів.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- Однак С і С ++ мають і серйозні недоліки:
- відсутність повноцінних вбудованих структурних типів даних (наявні псевдоструктурні типи, які використовують адресну арифметику, недостатньо жорстко визначені, щоб контролювати багато операцій над цими даними, що призводить до великої кількості помилок, що виявляються тільки в процесі відладки програми);
- наявність синтаксичних неоднозначностей, які також не дозволяють компілятору контролювати правильність програми;
- обмежений контроль параметрів, що передаються в підпрограму, що також можна знайти лише в процесі налагодження програми, і т. п.

- **Спеціалізовані мови розробника** використовують для створення конкретних типів програмного забезпечення. До них відносять:
 - мови баз даних;
 - мови створення мережових додатків;
 - мови створення систем штучного інтелекту і т. д.
- **Спеціалізовані мови користувача** зазвичай є частиною професійних середовищ користувача, характеризуються вузькою спрямованістю і розробниками програмного забезпечення не використовуються.
- **Мови низького рівня** дозволяють здійснювати програмування практично на рівні машинних команд. При цьому отримують найоптимальніші, як з точки зору часу виконання, так і з точки зору обсягу необхідної пам'яті програми. Але ці мови абсолютно не годяться для створення великих програм і тим більше програмних систем.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- Вибір або формування стандартів розробки.
- Реальне застосування будь-якої технології проектування вимагає формування або вибору ряду стандартів, які повинні дотримуватися всіма учасниками проекту:
 - стандарт проектування;
 - стандарт оформлення проектної документації;
 - стандарт інтерфейсу користувача.

- **Стандарт проектування повинен визначати:**
- набір необхідних моделей (схем, діаграм) на кожній стадії проектування і ступінь їх деталізації;
- правила фіксації проектних рішень на діаграмах, в тому числі правила іменування об'єктів і угоди по термінології, набір Атрибутів для всіх об'єктів і правила їх заповнення на кожній стадії, правила оформлення діаграм, включаючи вимоги до форми і розмірів об'єктів;
- вимоги до конфігурації робочих місць розробників, включаючи налаштування операційної системи і використовуваних CASE-засобів;
- механізм забезпечення спільної роботи над проектом, в тому числі і правила інтеграції підсистем проекту і аналізу проектних рішень на несуперечність.

ПРИНЦИПОВІ РІШЕННЯ ПОЧАТКОВИХ ЕТАПІВ ПРОЕКТУВАННЯ

- **Стандарт оформлення проектної документації** повинен регламентувати:
 - комплектність, склад і структуру документації на кожній стадії;
 - вимоги до її змісту та оформлення;
 - правила підготовки, розгляду, узгодження і затвердження документів.
- **Стандарт інтерфейсу користувача** повинен визначати:
 - правила оформлення екранів (шрифти і колірну палітру), склад і розташування вікон і елементів управління;
 - правила користування клавіатурою і мишкою;
 - правила оформлення текстів допомоги;
 - перелік стандартних повідомлень;
 - правила обробки реакції користувача.

ДЯКУЮ ЗА УВАГУ!