

Якість програмного забезпечення та тестування

Модуль 1. Методи побудови тестів.

Методи пошуку помилок в програмах.

Поняття і визначення, пов'язані з процесом тестування

- **Тестування** – процес виконання програми з метою виявлення помилки.
- **Тестові дані** – входи, які використовуються для перевірки програми.
- **Тестовий випадок** (test-case) - це набір умов і/або змінних, за допомогою яких тестувальник визначатиме, наскільки повно і правильно об'єкт тестування задовольняє вимоги до нього.

Поняття і визначення, пов'язані з процесом тестування

- **Хороший тестовий випадок** – це випадок з великою вірогідністю виявлення поки що не виявленої помилки.
- **Вдалий тест** – такий, що виявив поки що не виявлену помилку.

Поняття і визначення, пов'язані з процесом тестування

- **Помилка** – дія програміста на етапі розроблення, яка приводить до того, що в програмному забезпеченні виникає внутрішній **дефект**, який може призвести до неправильного результату роботи програми.
- **Відмова** – непередбачувана поведінка системи, що приводить до неочікуваного результату, яка могла бути спричинене дефектами, що містяться в ній.

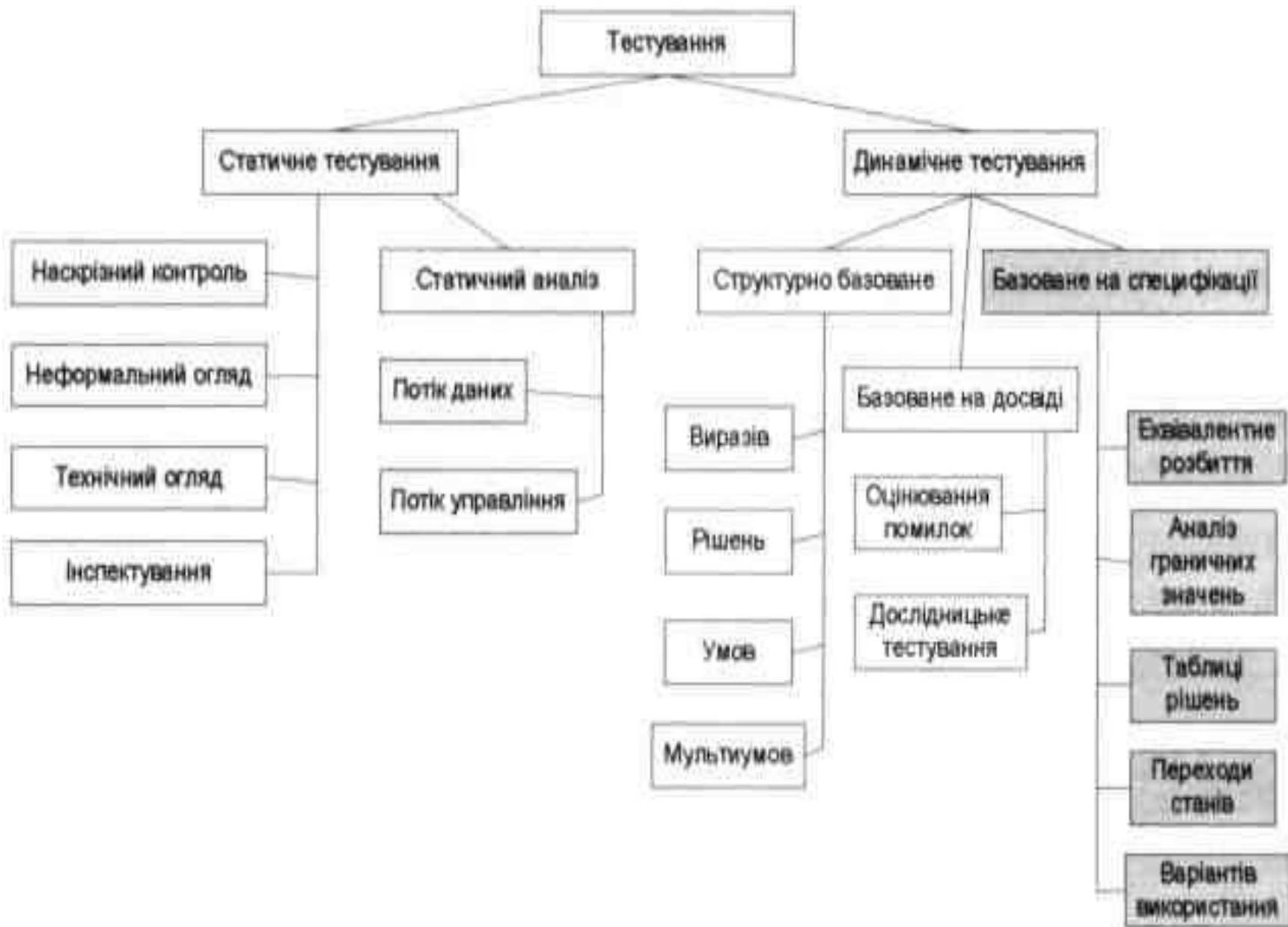
В процесі тестування програмного забезпечення перевіряють таке:

- програмне забезпечення відповідає вимогам;**
- у ситуаціях, не відображених у вимогах, програмне забезпечення поводить себе адекватно, тобто не відбувається відмови системи;**
- наявність типових помилок, яких припускають розробники.**

Тестування/ верифікація / валідація

- **Верифікація (verification)** забезпечує відповідність результатів конкретної фази процесу розроблення вимогам поточної і попередньої стадій.
- **Валідація (validation)** - гарантування того, що програмний продукт задовольняє системні вимоги.
- **Тестування** забезпечує виявлення (констатацію наявності) фактів розбіжностей з вимогами (помилки).

Методи тестування



Методи "чорної скриньки"

- **Всі матеріали, що доступні тестувальнику:**
 - ✓ **вимоги до системи, що описують її поведінку;**
 - ✓ **сама система, працювати з якою він може, тільки подаючи на її входи деякі зовнішні дії і спостерігаючи на виходах деякий результат.**
- **Всі внутрішні особливості реалізації системи приховані від тестувальника - система є "чорною скринькою", правильність поведінки якої згідно з вимогами належить перевірити.**

Методи "чорної скриньки"

Завдання тестувальника:

- ✓ **перевірка відповідності поведінки системи вимогам;**
- ✓ **перевірка роботи системи в критичних ситуаціях.**

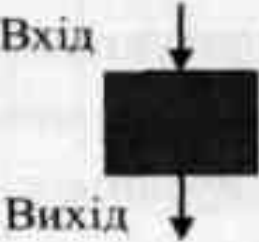
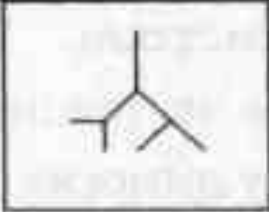
Два типи проблем системи:

- 1) Невідповідність поведінки системи вимогам.**
- 2) Неадекватна поведінка системи в ситуаціях, не передбачених вимогами.**

Методи "білої (скляної) скриньки"

- тестувальник має доступ не тільки до вимог до системи, її входів і виходів, але і до її внутрішньої структури - бачить її програмний код**
- може бачити відповідність вимог ділянкам програмного коду**
- може аналізувати ділянки, що зумовлюють проблеми**

Методи тестування при різних стратегіях

Тестова стратегія	Погляд тестувальника	Джерела знань	Методи
Чорна скринька	Вхід  Вихід	Вимоги Специфікації Знання предметної області Дані аналізу дефектів	Розбиття на класи еквівалентності Аналіз граничних значень Тестування переходу станів Відображення причин-наслідків Припущення про помилки
Біла скринька		Високорівневі проекти Деталізовані проекти Графи потоку керування Цикломатична складність	Тестування виразів Тестування гілок Тестування шляхів Тестування потоку даних Мутаційне тестування

**Методи пошуку
помилوک у програмах.**

Що шукаємо?

Типи програмних інцидентів

- **Помилка (error) - стан програми, за якого видаються неправильні результати, причиною яких є вади в операторах програми або в технологічному процесі її розроблення, що приводить до неправильної інтерпретації початкової інформації, а отже, і до неправильного рішення.**
- **Дефект (fault) у програмі - наслідок помилок розробника на будь-якому з етапів розроблення, який може міститися в початкових або проектних специфікаціях, текстах кодів програм, експлуатаційній документації тощо.**

Типи програмних інцидентів

- **Відмова (failure) - це відхилення програми від функціонування або неможливість програми виконувати функції, визначені вимогами і обмеженнями, що розглядається як подія переходу програми в непрацездатний стан через помилки, приховані в ній дефекти або збої в середовищі функціонування.**

Причини відмов:

- помилкова специфікація або пропущена вимога, яка означає, що специфікація точно не відображає того, що припускав користувач;
- специфікація може містити вимогу, яку неможливо виконати на цій апаратурі і програмному забезпеченні;
- проект програми може містити помилки (наприклад, база даних спроектована без засобів захисту від несанкціонованого доступу користувача, а захист є потрібним);
- програма може бути неправильною - виконувати неправильний алгоритм або не повністю реалізований.

Помилки на етапах процесу розроблення

Джерела виникнення :

- **ненавмисне відхилення розробників від робочих стандартів або планів реалізації;**
- **специфікації функціональних і інтерфейсних вимог виконані без дотримання стандартів розроблення, що приводить до порушення функціонування програм;**
- **організація процесу розроблення - недосконала або недостатнє управління керівником проекту ресурсами (людськими, технічними, програмними і так далі) та питаннями тестування і інтеграції елементів проекту.**

В процесі розроблення вимог

Помилки аналітиків під час специфікації верхнього рівня системи і побудові концептуальної моделі предметної області;

- неадекватність специфікації вимог кінцевим користувачам;
- некоректність специфікації взаємодії ПЗ з середовищем функціонування або з користувачами;
- невідповідність вимог замовника окремим і загальним властивостям ПЗ;
- некоректність опису функціональних характеристик;
-

В процесі проектування.

Можуть виникати при описі алгоритмів, логіки управління, структур даних, інтерфейсів, логіки моделювання потоків даних, форматів вводу-виводу. Пов'язані з:

- визначенням інтерфейсу користувача з середовищем;**
- описом функцій (неадекватність цілей і завдань компонентів, які виявляються під час перевірки комплексу компонентів);**
- визначенням процесу обробки інформації і взаємодії між процесами (результат некоректного визначення взаємозв'язків компонентів і процесів);**

Пов'язані з:

- **некоректним завданням даних і їх структур при описі окремих компонентів і ПЗ загалом;**
- **некоректним описом алгоритмів модулів;**
- **визначенням умов виникнення можливих помилок в програмі;**
- **порушенням прийнятих для проекту стандартів і технологій.**

На етапі кодування.

виникають помилки внаслідок дефектів проектування, помилок програмістів і менеджерів у процесі розроблення і відлагодження системи.

Причиною помилок є:

- безконтрольність значень вхідних параметрів, індексів масивів, параметрів циклів, вихідних результатів, та ін.;**
- неправильна обробка нерегулярних ситуацій при аналізі кодів повернення від підпрограм, функцій і т.п.;**
- порушення стандартів кодування (погані коментарі, нераціональне виділення модулів і компонент та ін.);**
- використання одного імені для позначення різних об'єктів або різних імен одного об'єкта, погана мнемоніка імен;**
- неузгоджене внесення змін до програми різними розробниками та ін.**

В процесі тестування.

помилки допускаються програмістами і тестувальниками під час виконання технології складання і тестування, вибору тестових наборів і сценаріїв тестування та ін.

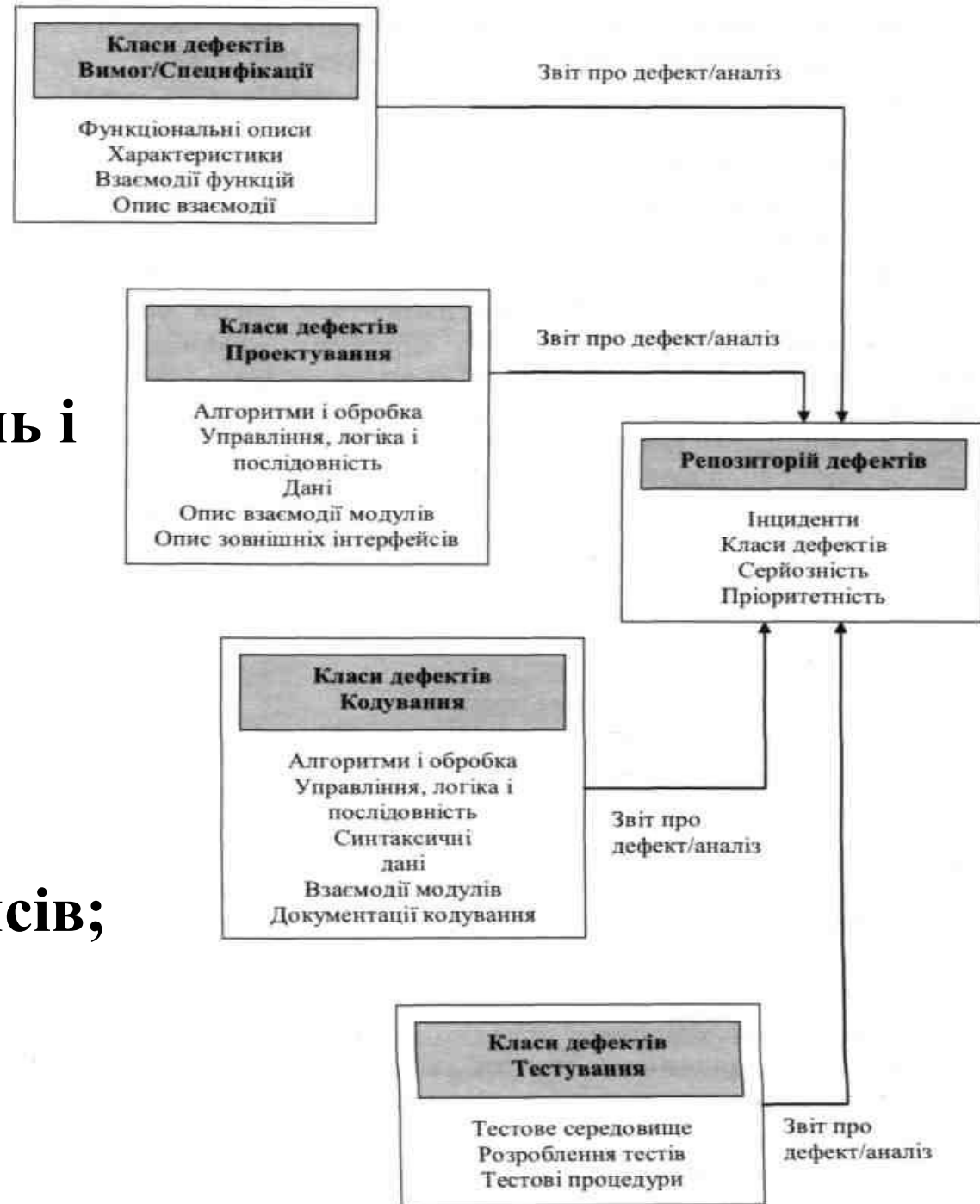
В процесі супроводу

причиною помилок є недоліки і дефекти експлуатаційної документації, недостатні показники модифікованості і легкості для читання, а також некомпетентність осіб, відповідальних за супровід і/або удосконалення ПЗ.

Можуть виникати будь-які помилки, аналогічні помилкам на попередніх етапах

Класи помилок:

- логічні і функціональні помилки;
- помилки обчислень і часу виконання;
- помилки вводу/виводу і маніпулювання даними;
- помилки інтерфейсів;
- помилки об'єму даних;
-



Боротьба з помилками

вдосконалення організації процесу розроблення, тестування і супроводу ПЗ.

Аналіз відмов і помилок для визначення методів і засобів тестування і виявлення помилок певних класів, а також критеріїв завершення тестування:

- ідентифікація недоліків в технологіях проектування і програмування;
- взаємозв'язок вад процесу проектування і помилок, які допускаються людиною;
- класифікація відмов, недоліків і можливих помилок, а також дефектів на кожному етапі розроблення;

Аналіз відмов і помилок

- **зіставлення помилок людини, що допускаються на певному процесі розроблення, і дефектів в об'єкті як наслідків помилок специфікації проекту чи моделей програм;**
- **перевірка і захист від помилок на всіх етапах ЖЦ, а також виявлення дефектів на кожному етапі розроблення;**
- **зіставлення дефектів і відмов в ПЗ для розроблення системи взаємозв'язків і методики локалізації, збору і аналізу інформації про відмови і дефекти;**
- **розроблення підходів до процесів документування і випробування ПЗ.**

Класифікація типів відмов:

- апаратний, при якому загальносистемне ПЗ непрацездатне;**
- інформаційний, спричинений помилками у вхідних даних і передавання даних по каналах зв'язку, а також у разі збою пристроїв вводу (наслідок апаратних відмов);**
- ергономічний, спричинений помилками оператора при його взаємодії з машиною (ця відмова - вторинна відмова, може привести до інформаційної або функціональної відмов);**
- програмний, за наявності помилок в компонентах та ін.**

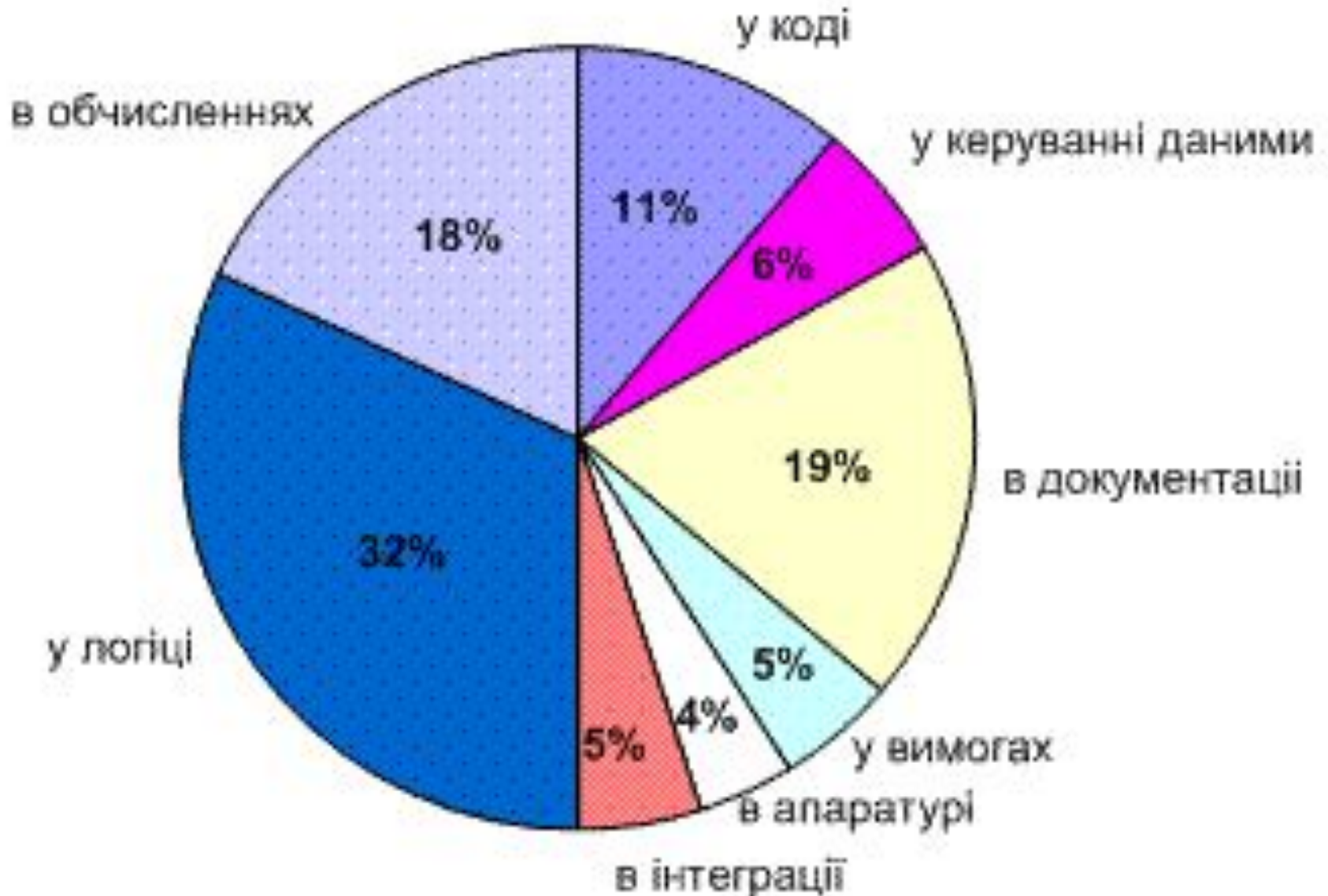
ІВМ: Ортогональна класифікація дефектів

Контекст помилки	Класифікація дефектів
Функція	Помилки інтерфейсів кінцевих користувачів ПЗ, спричинені апаратурою або пов'язані з глобальними структурами даних
Інтерфейс	Помилки у взаємодії з іншими компонентами, у викликах, макросах, керуючих блоках або в списку параметрів
Логіка	Помилки в програмній логіці, не охопленій валідацією, а також у використанні значень змінних
Присвоєння	Помилки в структурі даних або в ініціалізації змінних окремих частин програми
Зациклення	Помилки, викликані ресурсом часу, реальним часом або розподілом часу
Середовище	Помилки в репозиторії, в управлінні змінами або в контрольованих версіях проекту
Алгоритм	Помилки, пов'язані із забезпеченням ефективності, коректності алгоритмів або структур даних системи
Документація	Помилки в записах документів супроводу або в публікаціях

ІВМ:

- вартість аналізу і формування вимог, внесення в них змін $\approx 10\%$;
- вартість специфікації продукту $\approx 10\%$;
- вартість кодування $\geq 20\%$,
- вартість тестування $\geq 45\%$ від його загальної вартості.

Hewlett-Packard: *Процентне співвідношення помилок під час розроблення ПЗ*

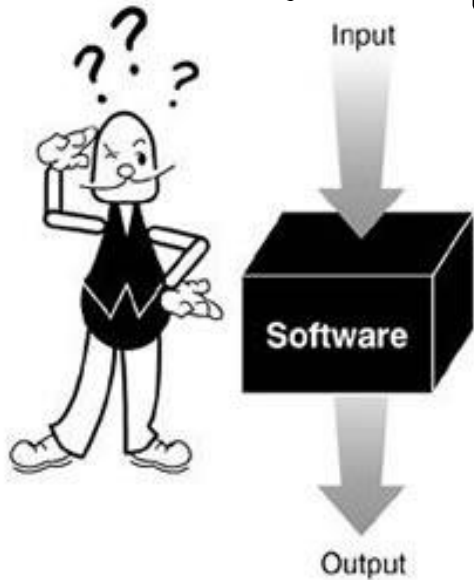


Контрольні питання:

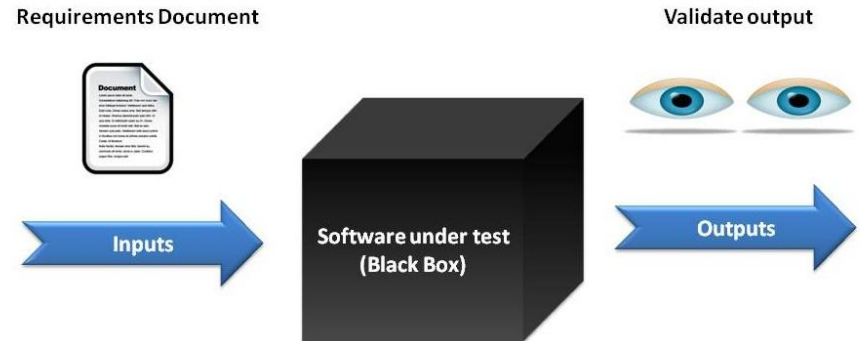
- **Дайте визначення процесу тестування.**
- **Які методи тестування ви знаєте?**
- **У чому полягає статичне тестування?**
- **Що таке програмна помилка?**
- **Які помилки виникають у процесі формування вимог?**
- **Які помилки виникають у процесі проектування?**
- **Які помилки виникають у процесі кодування?**
- **Які є джерела програмних помилок?**
- **Який відсоток у загальних витратах становлять витрати на тестування?**
- **Яку класифікацію помилок ви знаєте?**

Наступне

- Методологія розробки тестових випадків (чорної скриньки).



Black-Box Testing



www.SoftwareTestingSoftware.com