

## Лекція 8. Моделі і метрики якості програмних систем

Атрибути програмної системи, які характеризують її якість, вимірюються за допомогою *метрик якості*.

**Метрика** – це комбінація конкретного методу вимірювання атрибута і шкали вимірювання.

Метрика визначає *міру* – змінну, котрій присвоюється значення в результаті вимірювання.

Стандарт ISO 9126-2 визначає метрику якості програмної системи так: «модель вимірювання атрибута, який пов'язаний з деякою характеристикою якості ПС. Є індикатором одного або декількох атрибутів. Метрику можна побачити в лівій частині рівнянь типу  $X=A*B$ »

**A і B – базові метрики.**

## Приклади метрик:

**1. Повнота функціональної реалізації:**  $X=1-A/V$  ( $0 \leq X \leq 1$ ).

A – число нереалізованих функцій,

V – число функцій, описаних в специфікації вимог.

**2. Точність обчислювання даних:**  $X=A/V$ .

A – число елементів даних, для яких забезпечується встановлений рівень точності.

V – число елементів даних, для котрих в специфікації встановлені рівні точності.

**3. Інтенсивність виявлення дефектів.**  $F = x_1/M(x)$ .

$x_1$  – кількість знайдених дефектів.

$M(x)$  – очікувана кількість дефектів.

## Стандарт ISO 9126-2 визначає 5 видів шкали вимірювань:

- Номінальна шкала (класифікаційна). Якісна шкала
- Порядкова шкала (впорядковує характеристики). Якісна шкала
- Інтервальна шкала
- Відносна шкала
- Абсолютна шкала

## Класифікація мір якості:

- Міри розміру
  - Функціональний розмір
  - Розмір програми
- Об'єм ресурсів, які використовує програма

- ◎ Міри часу
  - Час функціонування системи
  - Час виконання задачі
  - Час використання
- ◎ Міри зусиль
  - Продуктивність праці
  - Трудомісткість
- ◎ Міри інтервалів між подіями (наприклад час між відмовами)

- ◎ Розрахункові міри
  - Кількість знайдених помилок
  - Структурна складність програми (кількість програмних шляхів, циклічна складність)
  - Число несумісних елементів
  - Число змін
  - Число знайдених відмов
  - Ергономічні лічильники
  - Лічильники-оцінки (очки, бали)
- ◎ Базові міри розміру, часу і розрахункові можуть комбінуватись.

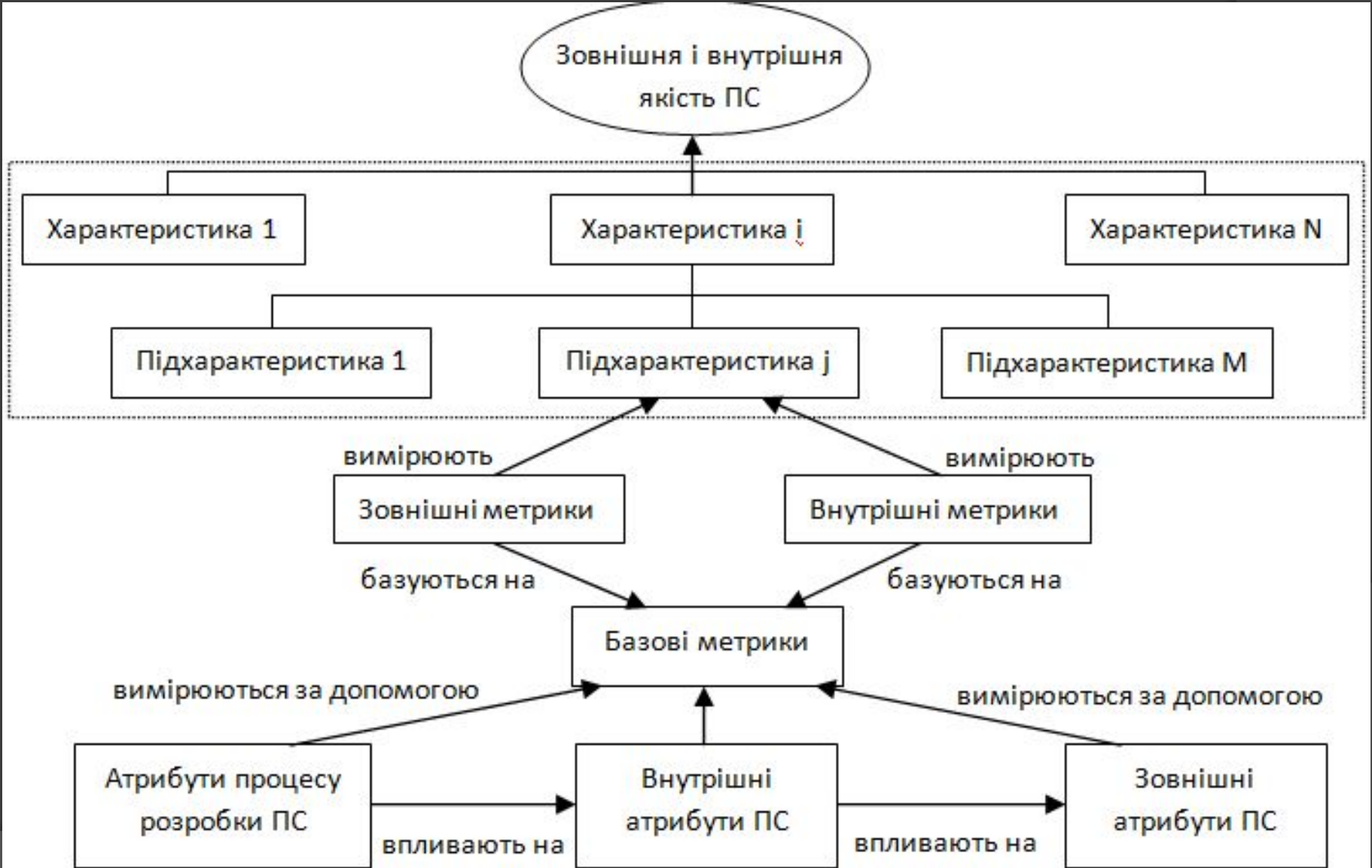
## Класифікація метрик якості:

- ◎ Об'єктивні/суб'єктивні
- ◎ Примітивні/обчислювані
- ◎ Динамічні/статичні
- ◎ Передбачаючі/пояснюючі

По відношенню до об'єкту вимірювань (ПС) метрики діляться на зовнішні, внутрішні і метрики використання ПС.

- ◎ **Зовнішні метрики** використовують міри працюючого на комп'ютері програмного продукту, отримані в результаті вимірювання його поведінки в ході тестування і функціонування.
- ◎ **Внутрішні метрики** забезпечують можливість оцінювати якість проміжних і кінцевих продуктів ПС безпосередньо по їх властивостям, без виконання на комп'ютері.
- ◎ **Метрики якості у використанні** вимірюють ступінь, з яким програмний продукт, який експлуатується в певному середовищі, задовольняє потреби користувача.

## УЗАГАЛЬНЕНА МОДЕЛЬ ЯКОСТІ



## Ієрархічні моделі якості ПС

- ◎ Модель МакКола (1977 р)
- ◎ Модель Боема (1978 р)
- ◎ Модель Ваттса (1987 р)
- ◎ Модель Дьюча та Вілліса (1988 р)
- ◎ ISO 9126 (1991 р)
- ◎ COQUALMO (Constructive Quality Model) – запропонована Боемом для визначення трудомісткості та вартості розробки ПС.



## Неієрархічні моделі якості ПС

1999 р, А. Абран і Л. Бугліоне запропонували модель QEST (Quality Factor + Economic, Social and Technical Dimensions).

Три виміри моделі:

E – економічний аспект (точка зору менеджерів на вартість та графік розробки).

T – технічний аспект (точка зору розробників на досягнення якості).

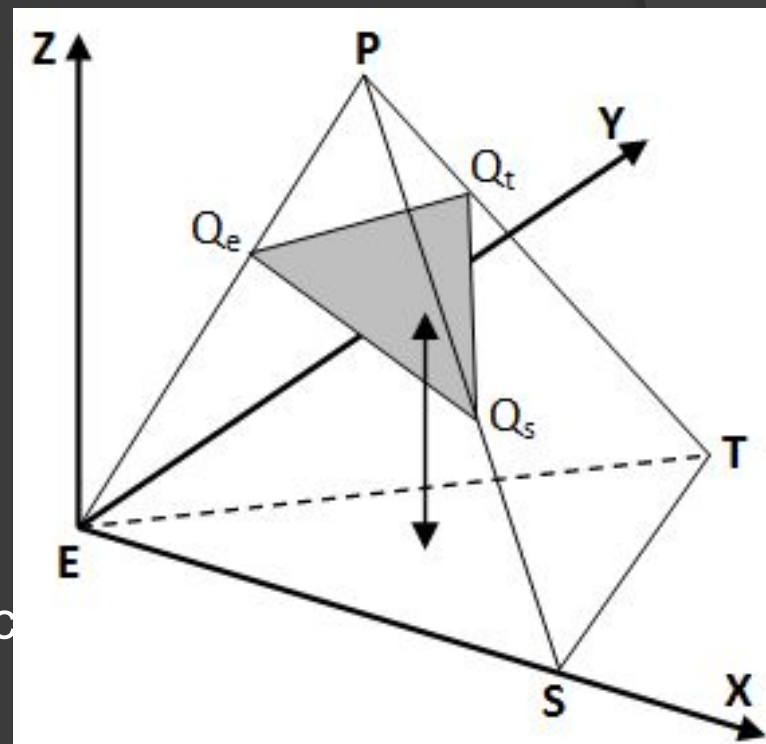
S – соціальний аспект (точка зору користувачів).

Вершина P – максимальний рівень якості.

Значення по кожному виміру являють

собою нормовану суму зважених оцінок експертів. Отриманий переріз

відокремлює «кращі» та «гірші» рівні якості. Характеристика якості – відстань до площини основи або площа перерізу.



В 2001 р Абраном та Кецесі запропонована схема GDQA (Graphical Dynamic Quality Assessment).

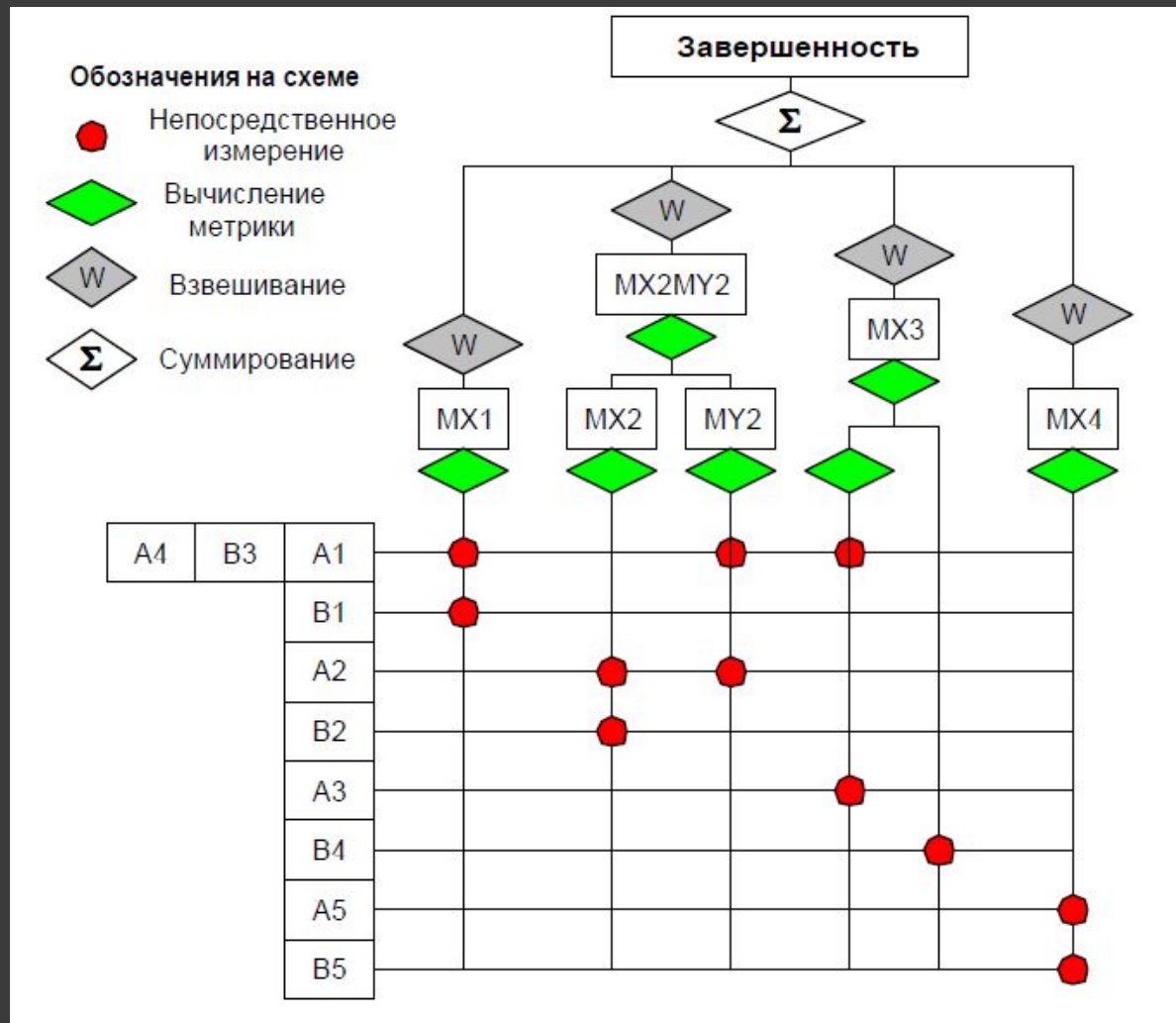


Таблица 3.6. Внутренние метрики «завершенности» ПС

Название метрики	Обозначение	Формула	Собираемые данные	
			Имя	Описание
Интенсивность обнаружения ошибок при проверке	MX1	$MX1 = A1/B1$	A1	Число ошибок, обнаруженных при проверке
			B1	Ожидаемое число ошибок, которые будут обнаружены при проверке
Количество устраненных дефектов	MX2Y2	$MX2Y2 = MX2/MY2$ $MX2 = A2/B2$ $MY2 = A2/B3$	A2	Число откорректированных дефектов в проекте, коде...
			B2	Оцениваемое число дефектов, которые будут обнаружены в ПС (при инспекции, тестировании)
			B3	Число ошибок, обнаруженных при проверке
Плотность оставшихся дефектов	MX3	$MX3 = (A3 - A4) / B4$	A3	Ожидаемое число ошибок, которые будут обнаружены при проверке
			A4	Число ошибок, обнаруженных при проверке
			B4	Оцениваемый размер ПС
Адекватность тестирования	MX4	$MX4 = A5/B4$	A5	Число тестовых ситуаций в плане тестирования
			B4	Число тестовых ситуаций, которые должны быть предусмотрены для обеспечения адекватности тестового покрытия

## БАЄСІВСЬКИЙ ПІДХІД.

Часто основне мірило якості – кількість знайдених дефектів.

Очікувану кількість дефектів необхідно прогнозувати. Для прийняття рішень в умовах невизначеності використовується формула повної ймовірності і формула Баєса.

Простий приклад.

Нехай  $A_1$  – розроблено програмний продукт високої якості,  $A_2$  – низької якості;

$B_1$  – розробку виконав спеціаліст високої кваліфікації,  $B_2$  – низької.

Апріорні ймовірності:  $P(B_1)=0.3$ ,  $P(B_2) = 0.7$ .

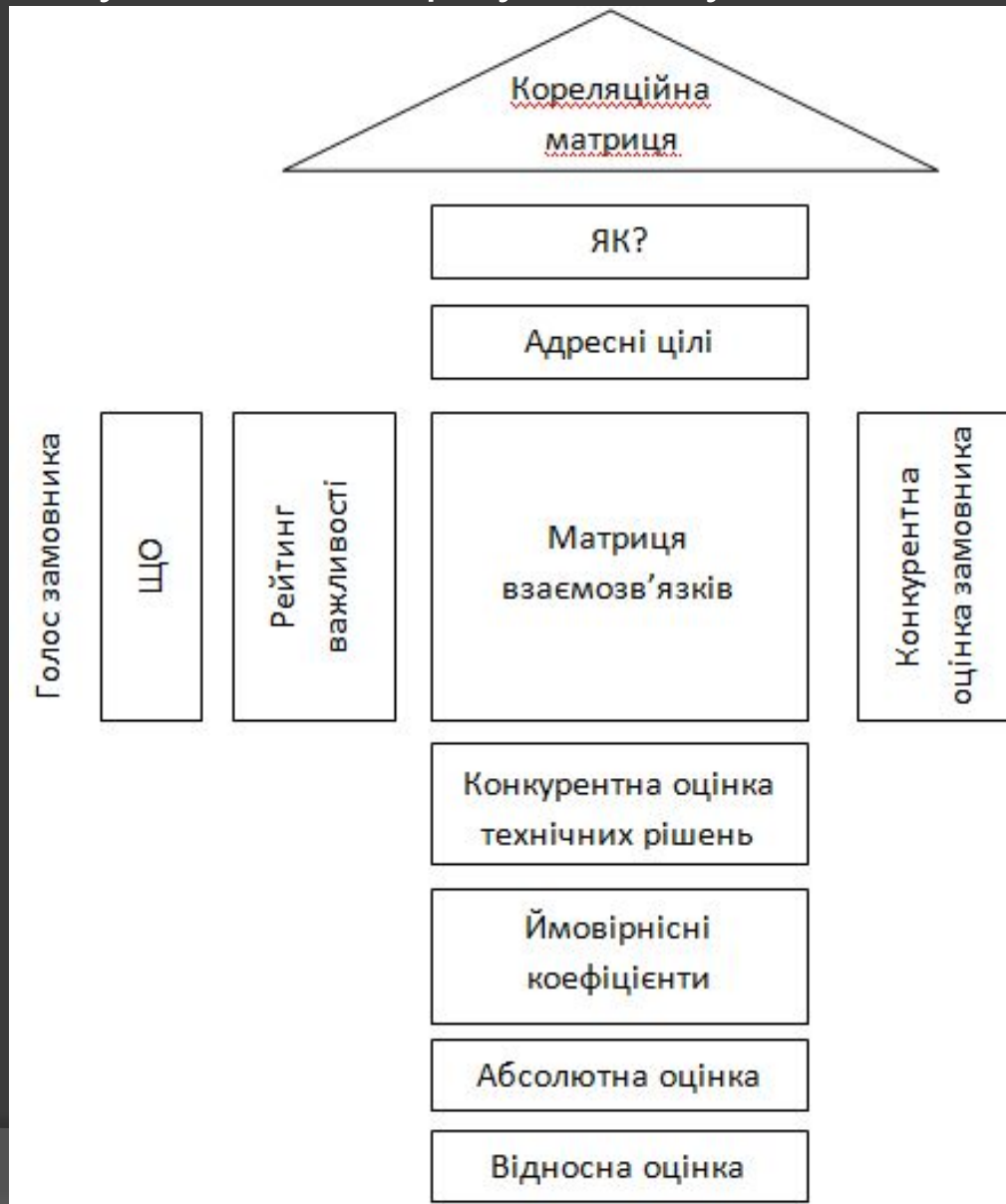
Умовні ймовірності:  $P(A_1|B_1) = 0.6$ ,  $P(A_1|B_2) = 0.2$

Тоді за формулою повної ймовірності:

$$P(A_1) = \sum_{i=1}^2 P(A_1|B_i) \cdot P(B_i) = 0.6 * 0.3 + 0.2 * 0.7 = 0.32$$

# ПРОЦЕСИ ТА СИСТЕМИ ПІДТРИМКИ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

## QFD – Quality Function Deployment. Будинок якості



## Вимірювання як процес життєвого циклу

Процес вимірювання входить в групу організаційних процесів ЖЦ і пов'язаний з процесом верхнього рівня – процесом керування.

4 основні причини, які обумовлюють вимірювання процесів, продуктів і ресурсів в ЖЦ це необхідність їх:

- Охарактеризувати
- Оцінити
- Передбачити
- Вдосконалити

Визначення Н. Фентона: Вимірювання - це процес, в ході якого атрибутам сутностей реального світу присвоюються числові або символічні значення, що дозволяє охарактеризувати атрибути за допомогою ясно визначених правил.

Види сутностей, які представляють інтерес з позиції вимірювання:  
*продукти, процеси, ресурси, артефакти, дії, агенти, організації, середовища, обмеження.*



**Мистецтво вимірювання** полягає в прийнятті рішень про те, які атрибути необхідно використати для того, щоб дати правильне представлення про відповідні сутності.

Приклади атрибутів для класу сутностей:

- ◎ **Система:** розмір, щільність дефектів
- ◎ **Компонент:** довжина (число рядків коду, число операторів), об'єм повторного використання
- ◎ **Рядок коду:** тип оператора, мова програмування
- ◎ **Дефект:** тип, місце розташування, серйозність, зусилля на усунення, вік (кількість часу від моменту відкриття дефекту)
- ◎ **Процес розробки:** час розробки, контрольні строки, зусилля на розробку, відповідність стандартам, ефективність.

Між суб'єктивними та об'єктивними слід вибирати *об'єктивні* вимірювання (і відповідні їм метрики)

## ПРОЦЕСИ ТА СИСТЕМИ ПІДТРИМКИ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

Існують різні моделі і механізми трансформації інформаційних потреб у вимірювані атрибути і множину придатних до вимірювання мір і метрик.

Найбільш відомі – QFD,

GQM (Goal-Question-Measure) парадигма **«ціль-питання-метрика»**.

Керований цілями процес вимірювання базується на 3 принципах і включає 10 етапів робіт.

Три принципу вимірювань:

- ◎ Цілі вимірювання отримуються із ділових цілей
- ◎ Контекст вимірювання забезпечується побудовою ментальних моделей
- ◎ Неформальні цілі перетворюються в інформаційні структури вимірювань.



## Етапи процесу вимірювання:

- Визначити ділові цілі
- Ідентифікувати об'єкти дослідження
- Визначити підцілі
- Ідентифікувати сутності і атрибути, пов'язані з підцілями
- Формалізувати цілі вимірювання
- Сформулювати питання, які вимагають відповіді у кількісній формі і визначити наглядні засоби відображення залежностей (діаграми), які допоможуть досягти цілей вимірювання
- Визначити елементи даних, які збираються для конструювання діаграм
- Дати визначення використаних мір і зробити їх основаними на конкретних метриках
- Ідентифікувати дії, які будуть зроблені для виконання вимірювань (визначення мір)
- Підготувати план реалізації вимірювань

## ПРОЦЕСИ ТА СИСТЕМИ ПІДТРИМКИ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

Не дивлячись на існування різноманітних методологій проведення вимірювань, статистика показує, що біля 80% програм вимірювань були невдалими, а деякі навіть шкідливі із-за зловживання вимірюваннями.

### **Загальні рекомендації по впровадженню вимірювань:**

- Необхідно починати з малої кількості самих простих метрик, поступово збільшуючи об'єм по мірі накопичення досвіду в організації
- Встановлювати чіткі цілі і плани вимірювань. Програму вимірювань оформити у вигляді самостійного проекту по вдосконаленню процесу вимірювань
- Створити середовище, яке забезпечує безпеку процесу збору даних і правильність самих даних. Розробники повинні мати стимули збирати коректні дані
- Вповноважити і запропонувати розробникам використовувати інформацію вимірювань для аналізу власних дій
- Своєчасно доводити інформацію до усіх зацікавлених сторін, які беруть участь в програмі вимірювань.