

Dr. Mileff Péter

## Integrált szoftver rendszerek és minőségbiztosításuk

A szoftver mérése és metrikái  
Miskolci Egyetem  
Általános Informatikai Tanszék

## A szoftver mérése

- **Szoftvermérés:** hogyan lehet a szoftvertermék vagy a szoftverfolyamat valamely jellemzőjéből numerikus értéket előállítani.
  - az értékeket összegyűjtik, tárolják
  - egymással és az egész szervezetre alkalmazott szabványokkal összehasonlíthatók
  - levonhatók bizonyos következtetések a szoftver vagy a szoftverfolyamat minőségére vonatkozóan.

2

## A mérés területei

Két terület, ahol a mérések használhatók:

- **Általános előrejelzés készítése a rendszerről:**
  - A rendszerkomponensek jellemzőinek mérése, és azok összegzése
  - Így becslést adhatunk a rendszer jellemzőire
    - például a rendszerben előforduló hibákra
- **A rendellenes komponensek azonosítása:**
  - Bizonyos egyedi komponensek jellemzői megsérthetnek bizonyos normákat.
  - Pl.: mérhetjük a komponenseket azért, hogy felfedezzük a legösszetettebbeket
    - ezután a felülvizsgálati folyamatban ezekre koncentrálnunk.
    - Feltételezzük, hogy a hibák többsége itt lesz.

3

## SZOFTVERMETRIKÁK...

4

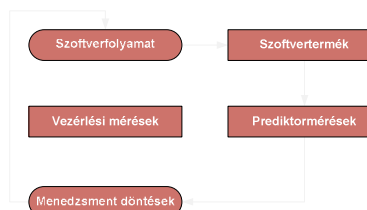
## A metrika

- **Szoftvermetrika:**
  - Minden mérés, amely egy szoftverrendszerhez, szoftverfolyamathoz vagy az ezekhez kapcsolódó dokumentációhoz fűződik.
  - Pl.: a termék méretének kifejezése a kódsorok számában
- A szoftveriparban igen kis mértékben használnak csak metrikákat
  - A minőségfejlesztés fontos szerepe miatt egyre inkább igény van rá.
  - A középpontban leginkább:
    - a programhibák, a verifikációs és validációs eljárásokhoz tartozó mérőszámok.

5

## A metrikák típusai

- Vezérlési metrikák (Control metrics)
- Prediktor metrikák (Predictor metrics)



Metrikák hatása a vezetői döntéshozatalra

6

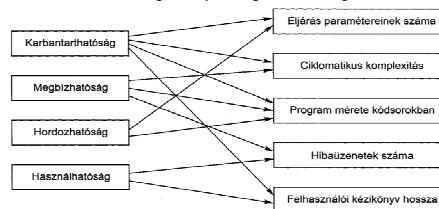
## A metrikák típusai

- A **vezérlési metrikák** általában a szoftverfolyamattal kapcsolatosak.
  - Pl.: a feljegyzett hibák kijavításához szükséges átlagos idő és munka
- A **prediktormetrikák** inkább a szoftvertermékre koncentrálnak.
  - a modul ciklomatikus komplexitása,
  - a program azonosítóinak átlagos hossza,
  - a tervezett objektumok műveleteiben szereplő attribútumok száma.

7

## A minőségi jellemzők

- A szoftver minőségi jellemzőit gyakran lehetetlen közvetlenül lemérni.
  - A külső jellemzők arra vonatkoznak, hogy a fejlesztők és a felhasználók hogyan látják a szoftvert.
    - Pl.: a karbantarthatóság, a bonyolultság és érthetőség,



Kitchenham (1990)

8

## A metrikák típusai

- Az ábra feltünteti a külső és belső jellemzők közötti kapcsolatot.
  - a kapcsolat milyenségéről nem mond semmit.
- A belső tulajdonságok mérése akkor hasznos predikátora a szoftver külső karakterisztikájának:
  1. A belső jellemzőnek pontosan mérhetőnek kell lennie.
  2. Léteznie kell valamilyen kapcsolatnak a mérhető és a bennünket érdeklő külső viselkedésű jellemzők között.
  3. A kapcsolat ismert, validált és kifejezhető képlet vagy modell segítségével.

9

## A MÉRÉS FOLYAMATA...

10

## A mérés folyamata



11

## A mérés folyamata

- A szoftverek mérésénél a rendszer minden egyes komponensét külön elemzik
  - a metrika különböző értékeit összehasonlítják egymással,
    - Esetleg a korábbi projektekből összegyűjtött mérési adatokkal is.
- A rendellenes mérési eredmények:
  - Általában a problémás komponensek esetén
  - Mutatja, hogy nagyobb odafigyelést kell szentelni ezekre

12

## A mérés folyamata

- **1. Az alkalmazandó mérések kiválasztása**
  - Meg kell fogalmazni, hogy mit szeretnénk mérni
    - amelyre a mérés fogja megadni a választ,
    - meg kell határozni a kérdések megválaszolásához szükséges méréseket.
- **2. A mérni kívánt komponensek kiválasztása**
  - Nem biztos, hogy egy szoftverrendszer minden komponensét szükséges mérni.
  - A komponenseknek csak egy reprezentatív csoportját jelölik ki mérésre.

13

## A mérés folyamata

- **3. A komponensek jellemzőinek mérése**
  - Lemérik a kiválasztott komponenseket, kiszámítják a metrikus értékeket
    - magában foglalja a komponens reprezentációjának (terv, kód stb.) feldolgozását
    - automatizált adatgyűjtő eszköz felhasználása
- **4. A rendellenes mérések azonosítása**
  - A mért értékeket összehasonlíthatók
    - egymással és a mérési adatbázisban feljegyzett korábbi mérésekkel.
  - A metrikáknál meg kell keresni a szokatlanul magas vagy alacsony értékeket.

14

## A mérés folyamata

- **5. A rendellenes komponensek elemzése**
  - A rendellenes értéket mutató komponensek vizsgálata
  - El kell dönteni, hogy a rendellenes metrikus értékek veszélyeztetik-e a komponens minőségét.
- Az összegyűjtött adatokat célszerű tárolni
  - elég nagy mérési adatbázis felállítása után összehasonlíthatók a projektek,
  - olyan információk nyerhető ki, amelyek segíthetnek a minőség növelésében.

15

**TERMÉK METRIKÁK...**

16

## Termék metrikák

- **Termékmétrikák:** magának a szoftvernek a jellemzőihez kapcsolódnak
  - Vannak belső és külső jellemzők
    - Köztük kapcsolat
  - A kapcsolatok felderítéséhez és validálásához a létező rendszerről sok adatot kell gyűjteni.
  - Felhasználható arra, hogy kiderítsük, hogyan kapcsolódnak a külső minőségek a szoftvertémék jellemzőihez.
  - A termékmétrikák két osztályba sorolhatók:
    - Dinamikus metrikák
    - Statikus metrikák

17

## Dinamikus termék metrikák

- **Fontosabb jellemzőik:**
  - Egy program futása közben készített mérések állítják elő
  - Segítenek a program hatékonyságának és megbízhatóságának megállapításában.
  - Általában szoros kapcsolatban állnak a szoftver minőségi jellemzőivel.
  - Pl.: viszonylag könnyen mérhető az egyes funkciók végrehajtásához és a rendszer elindításához szükséges idő,
    - Ez összefügg a rendszer hatékonyságával.
  - Pl.: a rendszer meghibásodásainak száma és a meghibásodás típusa is naplózható
    - Ez összefügg a szoftver megbízhatóságával.

18

## Statikus termék metrikák

- **Fontosabb jellemzőik:**
  - A rendszer reprezentációjának mérései gyűjtik össze.
    - például terv, program vagy dokumentáció
  - A szoftverrendszer komplexitásának, érthetőségének és karbantarthatóságának felmérésében segítenek.
  - A statikus metrikák és a minőségi jellemzők között csak közvetett kapcsolat létezik.
    - Kutatások folynak a közvetett kapcsolat felismerésében napjainkban is

19

## PÉLDÁK STATIKUS METRIKÁKRA...

20

## Statikus termék metrikák

- **Fan-in/fan-out**

- A fan-in azon függvények száma, amelyek valamilyen más függvényt (nevezzük X-nek) hívnak meg.
- A fan-out az X függvény által meghívott függvények száma.
- **A fan-in magas értéke:**
  - azt jelenti, hogy az X szorosan kapcsolódik a terv többi részéhez, és megváltoztatása sok helyen ütközéshez vezethet.
- **A fan-out magas értéke:**
  - Jelzi hogy a meghívott komponensek koordinálásához szükséges vezérlőlogika komplexitása megnövelheti az X általános bonyolultságát.

21

## Statikus termék metrikák

- **A kód hossza**

- A program méretének mértéke.
- Ha a kód mérete nagy
  - annál valószínűbb, hogy a komponens bonyolultabb és hibára hajlamos.
- **A kód hossza az egyik legmegbízhatóbb mérték annak előrejelzésére, hogy a komponensben hibák lehetnek.**

22

## Ciklomatikus komplexitás

- **Ciklomatikus komplexitás**

- Thomas J. McCabe publikált 1976-ban
- A program vezérlésének bonyolultságát méri.
- Eredménye egy egész szám
  - alacsony értéke a program érthetőségével függ össze.
- **A komplexitás számítása a gráfelméletre alapul**
  - A forráskódban az elágazásokból felépülő vezérfolyam-gráf pontjai, és a köztük lévő élek alapján számítható.
  - A ciklomatikus (McCabe) komplexitás értéke:

$$M = E - N + 2P$$

- A ciklomatikus szám:

$$M = E - N + P$$

23

## Ciklomatikus komplexitás

E: A gráf éleinek száma

N: A gráfban lévő csúcsok száma

P: Az összefüggő komponensek száma

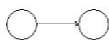
- **Szoftverek esetében:**

- a gráfot az adott függvényben lévő utasítások alkotják
- él akkor van 2 utasítás között, ha az egyik után a másik azonnal végrehajtható
- A metrika közvetlenül számolja a lineárisan független útvonalakat a forráskódon keresztül.

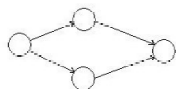
24

## Ciklomatikus komplexitás

szekvencia ( $v - 1 - 2 + 2 - 1$ ):



Az if-then-else szerkezet ( $v - 4 - 4 + 2 - 2$ ):



A while vezérlőszerkezet ( $v - 3 - 3 + 2 - 2$ ):

25

## Ciklomatikus komplexitás

Ciklomatikus komplexitás	Kockázati szint
1-10	Egyszerű program, kevés kockázat
11-20	Mérsékelt bonyolult program, mérsékelt kockázat
21-50	Bonyolult program, magas kockázat
50+	Nem tesztelhető program, nagyon magas kockázat

26

## Ciklomatikus komplexitás

- **Két hátránya:**

- **1.** A program döntési komplexitását méri
  - feltételes utasításokkal, előfeltételekkel van meghatározva.
- Ha a program adatorientált, akkor a ciklomatikus komplexitásra alacsony érték adódik,
  - viszont ettől még mindig elég komplex és nehezen érthető.
- **2.** Ugyanolyan súlyal szerepelnek a beágyazott és nem beágyazott ciklusok.
  - Azonban a mélyen beágyazott feltételes struktúrákat nehezebb megérteni.

27

## Statikus termék metrikák

- **Azonosítók hossza:**

- a program különböző azonosítóinak átlagos hosszát méri.
- Minél hosszabb egy azonosító, annál nagyobb az érthetőségének a valószínűsége
  - a program érthetősége egyenes arányban nő az azonosítók hosszával.

- **A feltételek egymásba ágyazásának mélysége:**

- méri, hogy a program *if* utasításai milyen mélyen vannak egymásba ágyazva.
  - A mélyen egymásba ágyazott *if* utasítások nehezen érthetők, és esetleg hibára hajlamosak is.

28

## Statikus termék metrikák

- **Fog index:**

- A dokumentum szavainak és mondatainak átlagos hosszát méri.
- Minél magasabb a Fog index értéke, annál nehezebb megérteni a dokumentumot.

29

## OO STATIKUS METRIKÁK...

30

## Statikus termék metrikák

- **Öröklődési fa mélysége:**

- a különálló szintek száma az öröklődési fában
- az alosztályok tulajdonságokat és műveleteket (módszereket) örökölnek a szuperosztálytól.
- Minél mélyebb az öröklődési fa, annál összetettebb a terv:
  - a fa leveleinél elhelyezkedő objektumosztályok megértéséhez sok különböző objektumosztállyal kell megismerkedni.

- **Módszer fan-in/fan-out:**

- közvetlenül kapcsolódik a fan-in/fan-out-hoz, jelentésük is megegyezik
- Itt nem árt megkülönböztetni az objektum más módszereiből és a külső módszerekből érkező hívásokat.

31

## Statikus termék metrikák

- **Osztályonkénti súlyozott metódusok:**

- az osztály módszereinek száma, súlyozva az egyes módszerek bonyolultságával.
- Az egyszerű módszerek bonyolultsága így 1, a nagyobb és összetettebb módszerek bonyolultsága ennél magasabb lesz
- Minél magasabb a metrika értéke, annál bonyolultabb az objektumosztály.
- A bonyolult osztályok nehezebben érthetők, de logikailag sem biztos, hogy összefüggnek
  - ezért szuperosztályként nem lehet őket ténylegesen újrafelhasználni az öröklődési fában.

32



## Statikus termék metrikák

- **Túlterhelt műveletek száma:**
  - a szuperosztály azon műveleteinek a száma, amelyeket a szuperosztály valamelyik alosztálya túlterhel.
- **A metrika magas értéke:**
  - azt jelenti, hogy a használt szuperosztály lehet, hogy nem alkalmas az alosztály szülőjének.

33

**Köszönöm a figyelmet!**

34