

Meranie v softvérovom projekte

Meranie a metriky

Oblasti uplatnenia softvérových metrik

Metódy a prostriedky merania

Typy meraní

Meranie a ISO 9000 (resp. ISO 90003)

Často používané metriky

Metriky pre výrobok

Metriky pre proces

Metriky pre zdroje

Metriky štýlu programovania

Príklad pre programy v jazyku C

Metódy merania

Faktory kvality softvéru

Metóda riadená cieľom

Meranie v softvérovom projekte

„If you cannot measure what you are doing then you have no control. All you have left is hysterical optimism.“ (Tom DeMacro)

„To measure is to know!“ (James Clark)

„Invisible targets are usually hard to hit.“ (Tom Gilb)

„If you don't know where you are going, how do you know if you have arrived?“ (Alka Jarvis, Vern Crandall)

“When you can measure what you are speaking about, and can express it in numbers, you know something about it...” (Lord Kevin)

- ▶▶ Ak nemožno merať, nemožno ani riadiť
ale tiež
Dostaneme to, čo meriame
- ▶▶ Pre úspech projektu je dôležitá dohoda na kritériách akceptácie projektu (rozsah projektu); tieto musia byť merateľné:
 - “Používateľské rozhranie musí byť priateľské”
a
 - “Operátori musia byť schopní začať pracovať s ľubovoľnou funkciou do 30 sekúnd odkedy si sadli za terminál”

Meranie a metriky

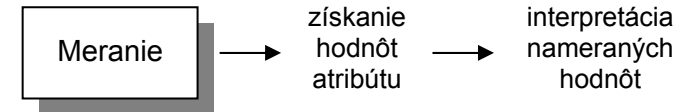
Metriky možno definovať ako kritériá určujúce atribúty v softvérovom projekte; umožňujú hodnotenie vytvoreného produktu a procesu, ktorý sa použil pri jeho tvorbe.

Meranie je proces objektívneho priradovania symbolov entitám (obyčajne čísel) za účelom opisu alebo charakterizácie určitého atribútu danej entity.

Entita → objekt alebo udalosť, ktorej charakteristika sa meria

Atribút → črta, charakteristika, ktorá nás zaujíma

Objektivita → skutočnosť, že vlastný proces merania musí vychádzať z presne definovaného pravidla, t.j. meranie nezávisí od toho, kto ho vykonáva



- rôzne interpretácie nameraných hodnôt:
 - počet chýb nájdených za jednotku času reprezentuje kvalitu testovania alebo (ne)spoľahlivosť programu

Meranie:

- *priame*: priame získanie hodnoty sledovaného atribútu (napr. počet riadkov programu)
- *nepriame*: odvodenie z iných atribútov, ktoré možno merať priamo (napr. udržateľnosť možno merať časom potrebným na odstránenie chyby)

Oblasti uplatnenia softvérových metrik

- definícia používateľských požiadaviek vo forme merateľných atribútov produktu umožňuje predchádzať potenciálnym nedorozumeniam
- poskytovanie kvantifikovateľného prehľadu pre podporu rozhodovania (napr. hodnotenie efektívnosti stratégií testovania, identifikácia potenciálne kritických a chybových modulov)
- vytváranie presnejších odhadov (najmä nákladov a dĺžky trvania budúcich projektov)
- sledovanie kvality softvéru aj procesu jeho tvorby
- prehľadnejší vývoj softvéru a skoršia identifikácia potenciálnych problémov na základe nameraných hodnôt vybraných atribútov charakterizujúcich projekt (výrobok, proces)
- vyhodnocovanie vplyvu nových metód a podporných prostriedkov na produktivitu práce a kvalitu
- zlepšovanie procesu aj výrobku

Metódy a prostriedky merania

- závisia od týchto charakteristík:
 1. aké entity sa merajú (napr. výrobky, procesy)
 2. prečo sa merajú (napr. vyhodnotenie, riadenie, predpoveď, zlepšenie)
 3. kto sa zaujíma o výsledky merania (napr. návrhár, manažér)
 4. ktoré vlastnosti sa už merajú a
 5. v akom prostredí sa merajú (ľudia, technológia, ďalšie zdroje)

Typy meraní

Aby sme mohli výsledok merania spracovať, treba výsledok merania reprezentovať na stupnici.

V praxi sa používa 5 rôznych typov meraní:

- **nominálne** (klasifikácia)
možno určiť iba, či sa merané veličiny “rovnajú” alebo “nerovnajú” (patria do rovnakej triedy), napr. typ programovacieho jazyka
- **ordinálne** (+ usporiadanie)
usporiadanie nad hodnotami umožňuje odmerať úroveň atribútu a porovnávanie, napr. “menej modulárny program”
- **intervalové** (+ vzdialenosti medzi prvkami stupnice)
uvažujú sa aj vzdialenosti medzi hodnotami, umožňuje výpočet strednej hodnoty, napr. rozdiel v modularite medzi programom X a Y je rovnaký ako medzi programami Y a Z
- **pomerové** (+ absolútna nula)
umožňuje porovnávanie pomerov, napr. program X je dvakrát tak modulárny ako program Y
- **absolútne**
všetky matematické operácie sú zmysluplné, napr. počet riadkov textu programu, počet chýb

Príklad metriky

Metrika	ukázkový príklad
Entita	zdrojový text programu
Atribút	modularita
Pravidlo	1: nízka – málo, veľkých komponentov 2: stredná – procedurálne komponenty, ktoré zahŕňajú základnú funkcionálnu 3: vysoká – funkcionálna dekompozícia, použitie abstraktných typov údajov
Typ	ordinálne

Platnosť metriky

Metrika je platná, ak má v matematickom svete také isté správanie, aké možno pozorovať v empirickom svete. Napr. ak H je metrika výšky, tak ak A je vyšší ako B, tak $H(A) > H(B)$.

Meranie a ISO 9000 (resp. ISO 90003)

- bod 8.2 Monitorovanie a meranie (pozri kapitolu Manažment kvality)
- požadujú sa metriky výrobku - 8.2.4 Monitorovanie a meranie produktu
„Organizácia musí monitorovať a merať charakteristiky produktu, aby si overila, či spĺňa naň kladené požiadavky. Musí sa to vykonávať v primeraných etapách procesu realizácie produktu v súlade s plánovanými opatreniami..“
- aj procesu - 8.2.3 Monitorovanie a meranie procesov
„Organizácia musí používať vhodné metódy na monitorovanie a ak treba, meranie procesov systému manažérstva kvality. Tieto metódy musia preukázať schopnosť procesov dosahovať plánované výsledky. Keď sa plánované výsledky nedosiahnu, musí sa podľa potreby urobiť náprava a nápravná činnosť na zaistenie zhody produktu.“
- nepredpisujú sa konkrétne metriky

Často používané metriky

- *teoreticky podložené metriky*, napr. Halsteadova teória (pokúšajú sa modelovať základné vlastnosti sledovaného objektu, najčastejšie programu)
- *empirické metriky*

Metriky pre výrobok

- **veľkosť**, **rozsah** (používa sa aj na odhad času a nákladov, meranie produktivity)
 - počet riadkov textu programu (LOC)
 - počet funkčných bodov (FP)
 - počet modulov
 - priemerný počet LOC na modul
 - dĺžka dokumentácie (počet strán, slov)
 - rozsah dokumentácie (počet strán/veľkosť programu)
- **zložitosť**
 - počet súborov
 - veľkosť programu (počet príkazov, riadkov,...)
 - počet vetvení (príkazy IF)
 - hĺbka vnorenia riadiacich štruktúr
 - počet cyklov
 - cyklomatické číslo (McCabe; zakladá sa na počte nezávislých ciest v grafe riadenia programu) - počet rozhodovacích príkazov +1
 - pamäťové nároky, nároky na procesor
- **chyby**
 - počet chýb a nedostatkov (v dokumentácii aj v programe)
 - chybovosť = počet chýb/FP
 - klasifikácia chýb a nedostatkov a frekvencia ich výskytu

- **modularita**
 - zviazanosť modulov (počet tokov údajov a riadenia medzi modulmi a počet globálnych štruktúr údajov)
 - súdržnosť modulov
- **spoľahlivosť**
 - priemerná doba medzi výpadkami systému (MTBF)
 $MTBF = MTTF + MTTR$
(MTTF - priemerná doba do nasledujúceho výpadku, MTTR - priemerná doba opravy)
- **dostupnosť**
 - pravdepodobnosť, že v danom čase program pracuje správne podľa špecifikácie
 $dostupnosť = 100 * MTTF/MTBF [\%]$
- **udržovateľnosť**
 - stredná doba potrebná na opravu chyby
 - stredná doba na pochopenie logiky modulu
 - stredná doba na nájdenie príslušnej informácie v dokumentácii

Metriky pre proces

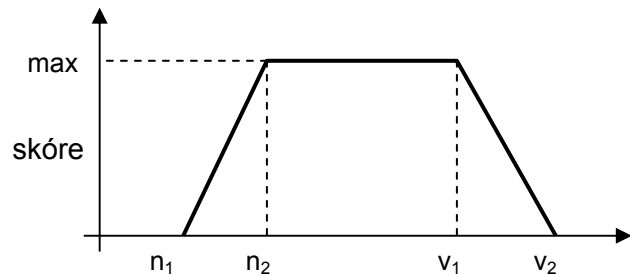
- **úsilie**
 - čas vynaložený na vývoj systému (človeko-mesiace)
- **zmeny požiadaviek** (odráža kvalitu špecifikácie požiadaviek)
 - počet zmien požiadaviek
 - stredný čas od ukončenia špecifikácie príslušnej časti do zapracovania požiadavky na zmenu
- **náklady a čas**
 - začiatky a konce činností
 - trvanie činností
 - náklady na vykonanie jednotlivých činností

Metriky pre zdroje

- **charakteristiky personálu**
 - produktivita
 - veľkosť tímu, rozsah komunikácie
 - skúsenosti
- **charakteristiky výpočtových systémov a softvérových prostriedkov**
 - kapacita pamätí, taktovacia frekvencia procesora
 - dostupnosť, použiteľnosť softvérových prostriedkov,...

Metriky štýlu programovania

- štýl programovania sa nedá merať priamo
- súčet vážených charakteristík programu
- každá charakteristika má pridelené určité bodové rozpätie, pričom čím viac bodov, tým lepší výsledok (R.E. Berry, B.A. Meeking)
- veľmi vysoké a veľmi nízke hodnoty sa považujú za nevhodné



Príklad pre programy v jazyku C

Sledujú sa tieto charakteristiky

(pozri aj <http://www.springer-ny.com/supplements/jalote/>):

1. *Dĺžka modulu*: priemerná dĺžka modulov v programe (merajú sa nekomentované, neprázdne riadky)
2. *Dĺžka identifikátora*: priemerná dĺžka v znakoch (počítajú sa programátorom definované identifikátory)
3. *Komentáre*: percento komentárových riadkov z celkového počtu riadkov programu
4. *Odsadenie*: pomer medzier na začiatku riadku k celkovému počtu znakov programu
5. *Prázdne riadky*: percento prázdnych riadkov (prázdne riadky sa používajú na zvýraznenie štruktúry programu)
6. *Dĺžka riadku*: priemerný počet znakov v riadku iných ako biele znaky (medzera, tabulátor)
7. *Vnorené medzery*: priemerný počet medzier v riadku (signalizujú čitateľnosť, napr. pri aritmetických a logických výrazoch)
8. *Definície konštánt*: percento identifikátorov, ktoré programátor definoval ako konštanty
9. *Rezervované slová*: počet rezervovaných slov (a štandardných knižničných funkcií), ktoré sa použili v programe
10. *Súbory Include*: počet (vyjadruje dekompozíciu programu do súborov)
11. *GoTo*: počet príkazov skoku (0 je najlepšie)

	max	n ₁	n ₂	v ₁	v ₂
1.	15	4	10	25	35
2.	14	4	5	10	14
3.	12	8	15	25	35
4.	12	8	24	48	60
5.	11	8	15	30	35
6.	9	8	12	25	30
7.	8	1	4	10	12
8.	8	10	15	25	30
9.	6	4	16	30	36
10.	5	0	3	3	4
11.	-5	1	3	99	99

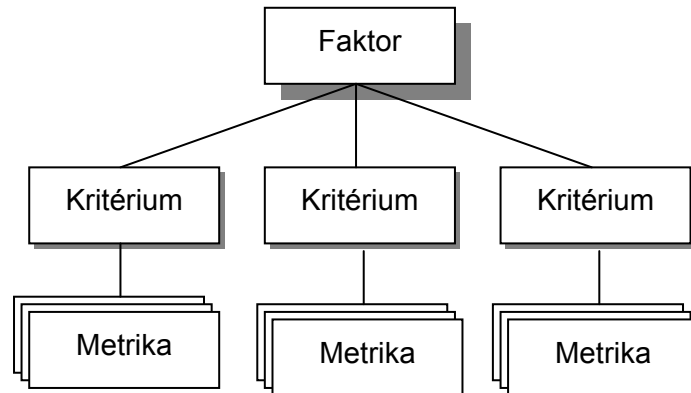
dobrý štýl:
skóre > 60

zlý štýl:
skóre < 20

Metódy merania

Faktory kvality softvéru

- autor: McCall a kol., 70-te roky
- získanie kvantitatívneho pohľadu na kvalitu
- všetky aspekty kvality nemusia byť rovnako dôležité



- vyhodnotenie faktora

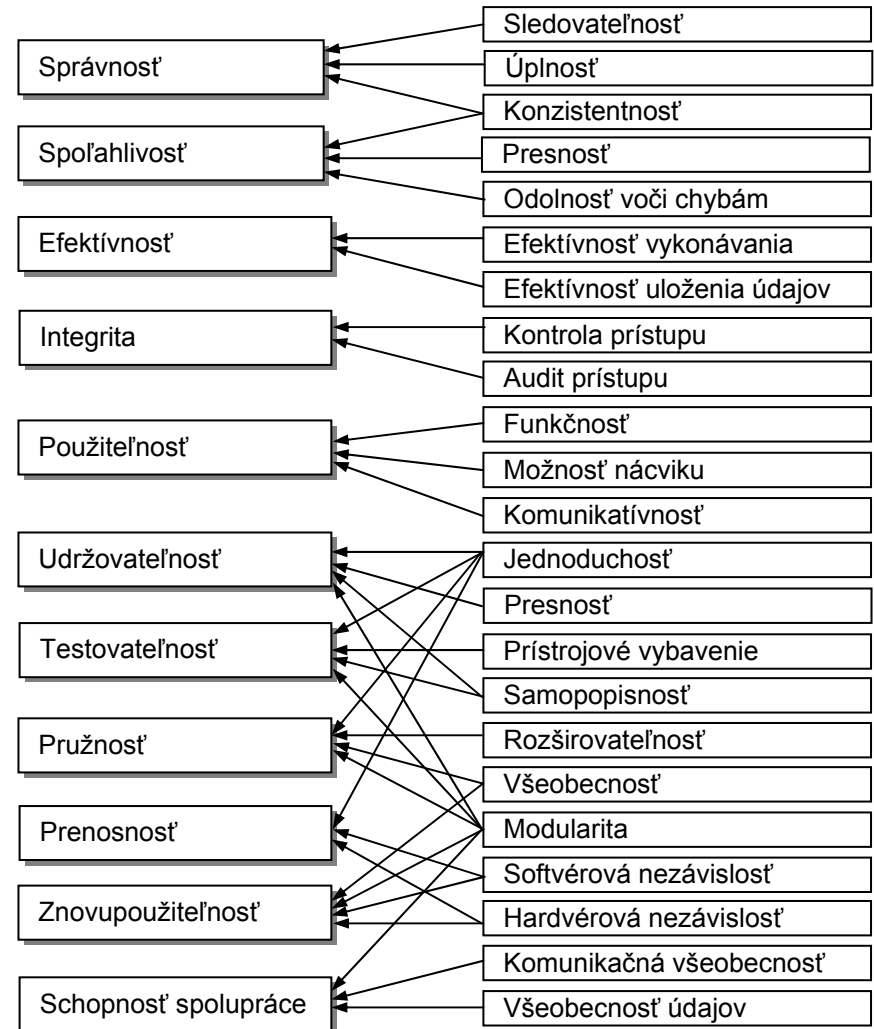
$$F_q = c_1 * m_1 + \dots + c_n * m_n$$

c_i ... preddefinované koeficienty

m_i ... „subjektívne“ merania kritérií, ktoré určujú daný faktor kvality (0-10 b.)

- preddefinované vzťahy medzi faktormi, kritériami a metrikami (problémy pri posudzovaní úplne nedokončených produktov)
- potreba dopĺňania o nové faktory (napr. bezpečnosť)

McCullove faktory kvality



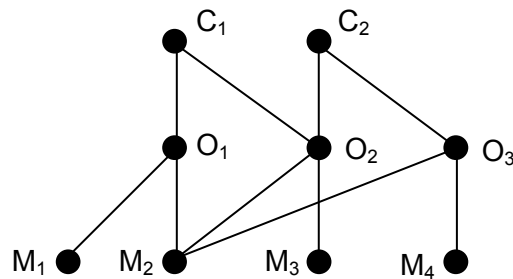
Metóda riadená cieľom

- vždy musíme mať dôvod pre meranie
- prvordá otázka je „**Čo je cieľom merania?**“
a nie „**Čo budeme merať?**“

Postup metódy riadenej cieľom (Cieľ-Otázka-Metrika):

1. Identifikácia cieľov

- definícia cieľov
- polozenie otázok
- odvodenie metrík



2. Plánovanie procesu merania

- preformulovanie otázok do hypotéz
- identifikácia tých častí hierarchie Cieľ-Otázka-Metrika, ktoré sa považujú za najdôležitejšie

3. Vykonalenie merania

- zbieranie údajov (automaticky, manuálne)
- validácia údajov

4. Analýza a interpretácia