

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Igor Rožanc

Kakovost programske opreme Merjenje programske opreme

**Študijsko gradivo za interno uporabo pri
predmetu Testiranje in kakovost (TiK)**

Ljubljana, 2017/18

Vsebina:

1

Kakovost programske opreme (Software Quality)

- kaj je kakovost programske opreme
- vidiki kakovosti PO
- modeli kakovosti
 - McCalov model, Boehmov model
 - FURPS model, CUPRIMDSO model
 - Dromeyev model
 - ISO 9126, ISO 25010
- Total Quality Management (TQM)

Merjenje programske opreme (Software Measurement)

- Praktično merjenje programske opreme (PSM)



Kakovost programske opreme je ...

2

- nič izmeta (Cosby, 1979)
- stopnja odličnosti (Oxford English Dictionary)
- skupina lastnosti izdelka ali storitve, da vplivajo na njeno možnost zadovoljevanja opredeljenih in samo po sebi umevnih potreb (ISO 8402)
- sposobnost biti uporaben za določen namen (Juran, 1988)
- izjemnost, kot ljudje (Gillies)
- kakovost programskega izdelka je določena s tem, koliko je izboljšala možnosti za zadovoljitev ciljev uporabnika (De Marco)
- zahtevam, ki jih je eksplicitno ali implicitno določil naročnik, je v celoti zadoščeno
- z vidika TPO kakovost PO meri, kako dobro ustreza načrt programskega izdelka oziroma kako dobro programski izdelek izpolnjuje pričakovanja naročnika (Wikipedia, 2016)

Dejstva o kakovosti PO

3

Vsi menimo, da kakovost razumemo, a to običajno ni res:

1. Kakovost je večdimenzionalna
2. Kakovost lahko razumemo na različnih nivojih abstrakcije
3. “Splošna” in “profesionalna” definicija kakovosti se zelo razlikujeta
 - **Splošna:** “jo prepoznam, ko jo vidim”, subjektivna, opisna – ne merljiva, povezana z ceno ali zahtevnostjo izdelka
 - **Profesionalna:** konkretna in merljiva; govori o tem, ali izdelek ustreza zahtevam iz dokumentacije, lahko jo planiramo, načrtujemo, nadzorujemo, izboljšujemo

Dokumentacija je začetek in konec: vmes so ukrepi za zagotavljanje kakovosti in merjenje le-te

Dva vidika kakovosti:

4

Ključna vloga uporabnika:

- a) kakovost z vidika razvijalca – ustreznost zahtevam
- b) kakovost z vidika uporabnika – zadovoljstvo z izdelkom

ALI

- a) “mali q” – kakovost izdelka (št.napak / KLOC, št.odpovedi / uro delovanja)
- b) “veliki Q” – kakovost izdelka, kakovost procesa in zadovoljstvo uporabnika
(% zadovoljnih pri anketah uporabnikov)

Več atributov kakovosti:

5

Celotna kakovost je sestavljena iz več vidikov:

- opis kakovosti na različnih stopnjah abstrakcije
- najvišji nivo vsebuje tako pogled uporabnika kot razvijalca
- vidik kakovosti na višjem nivoju sestavlja več atributov na nižjem
- med posameznimi atributi so povezave
- atributi so merljivi – metrike
- pri določanju rezultatov na višjih nivojih upoštevamo uteži oziroma normiranje pri izračunih

Obstaja več uveljavljenih modelov, ki so prešli v (ISO, IEEE) standarde in “širše” modele za zagotavljanje kakovosti

Modeli kakovosti:

6

Različni modeli kakovosti

- 1. McCallov model** (1976-77, General Electrics – US AF)
- 2. Boehmov model** (1978)
- 3. FURPS / FURPS+ model** (Grady, Hewlet Packard)
- 4. CUPRIMDSO model** (IBM)
- 5. Dromeyev model**
- 6. ISO/IEC 9126**
- 7. ISO/IEC 250xx**

McCallov model kakovosti:

7

- osnovni (najbolj znan) model, prednik ostalih
- Namenjen razvoju različnih sistemov
- 3 vidiki:
 - Revizija izdelkov (sposobnost za spremembe)
 - Tranzicija izdelkov (sposobnost za namestitve v novih okoljih)
 - Uporaba izdelkov (značilnosti delovanja)
- 11 faktorjev (zunanja kakovost - z vidika uporabnika) – to specify
- **23 kakovostnih kriterijev** (notranja – z vidika razvijalca) – to develop
- Več metrik (način in merska skala) – to control
- Metrike – da/ne
- Rezultat: v % glede na kakovostni kriterij, faktor ali celoten izdelek

McCallov model kakovosti:

8

Revizija (3 faktorji):

- Vzdrževalnost (mainteinability) :
 - Enostavnost (simplicity)
 - Jedratost (consiseness)
 - Samoopisljivost (seld-descreptiveness)
 - Modularnost (modularity)
- Prilagodljivost (flexibility) :
 - Samoopisljivost (self-descreptiveness)
 - Razširljivost (expandability)
 - Splošnost (generality)
- Testabilnost (testability) :
 - Enostavnost (simplicity)
 - Samoopisljivost (seldf-descreptiveness)
 - Instrumentiranost (instrumentation)
 - Modularnost (modularity)

McCallov model kakovosti:

9

Tranzicija (3 faktorji):

- Prenosljivost (portability):
 - Samoopisljivost (self-descreptiveness)
 - Programska/sistemska neodvisnost (software-system independence)
 - Strojna neodvisnost (machine independence)
- Ponovna uporabljivost (reusability):
 - Samoopisljivost (self-descreptiveness)
 - Splošnost (generality)
 - Modularnost (modularity)
 - Strojna neodvisnost (machine independence)
- Sposobnost prenosa v različna okolja (interoperability):
 - Modularnost (modularity)
 - Splošnost komuniciranja (communication commonality)
 - Splošnost podatkov

McCallov model kakovosti:

10

Uporaba (5 faktorjev):

- Pravilnost (correctness):
 - Sledljivost (traceability)
 - Popolnost (completeness)
 - Usklajenost (consistency)
- Zanesljivost (reliability):
 - Usklajenost (consistency)
 - Natančnost (accuracy)
 - Toleranca (error toleranca)
- Učinkovitost (efficiency):
 - Učinkovitost delovanja (execution efficiency)
 - Učinkovitost hranjenja (storage efficiency)

McCallov model kakovosti:

11

Uporaba (nadaljevanje):

- Skladnost (integrity):
 - Nadzor dostopa (access control)
 - Ocena dostopa (access audit)
- Uporabnost (usability):
 - Operabilnost (operability)
 - Usposabljanje (training)
 - Komunikativnost (communicativeness)

Boehmov model kakovosti:

12

- Tudi klasičen hierarhičen model, namenjen za razvoj PO
- Trije nivoji abstrakcije:
- **Visokonivojske karakteristike kakovosti** – splošna uporabnost PO v smislu Uporabnosti (as-is utility), Vzdrževalnosti (maintainability) in Prenosljivosti (portability)
- **Srednjenivojske karakteristike kakovosti** - 7 kakovostnih faktorjev:
 - Prenosljivost (portability) - P
 - Zanesljivost (reliability) - U
 - Učinkovitost (efficiency) - U
 - Uporabnost (usability) - U
 - Testabilnost (testability) - M
 - Razumljivost (understandability) - M
 - Prilagodljivost (flexibility) - M

Boehmov model kakovosti:

13

- **Osnovne karakteristike kakovosti** so merljive vrednosti:

Prenosljivost:

- Neodvisnost naprav (device independance)
- Samovsebovalnost (self containdeness)

Zanesljivost:

- Samovsebovalnost (self containdeness)
- Natančnost (accuracy)
- Popolnost (completeness)
- Robustnost (robusteness)
- Usklajenost (consistency)

Učinkovitost:

- Zapisljivost (accountability)
- Učinkovitost naprav (device efficiency)
- Dostopnost (accessibility)

Boehmov model kakovosti:

14

Uporabnost:

- Robustnost (robustness)
- Dostopnost (accessibility)
- Komunikativnost (communicativeness)

Testabilnost:

- Zapisljivost (accountability)
- Komunikativnost (communicativeness)
- Samopisljivost (self descriptiveness)
- Strukturiranost (structuredness)

Razumljivost:

- Usklajenost (consistency)
- Strukturiranost (structuredness)
- Skladnost (conciseness)
- Bralnost (legibility)

Prilagodljivost:

- Strukturiranost (structuredness)
- Razširljivost (augmentability)

Boehmov model kakovosti:

15

- Zelo podoben McCallovemu, vendar bolj usmerjen na uporabniški vidik kakovosti – uporabnost.
- Glede na McCall dodaja tudi preverjanje lastnosti , kot so:
 - jasnost
 - možnost sprememb
 - dokumentiranje,
 - razumljivost
 - vrednost,
 - splošnost,
 - ekonomičnost,
 - ipd.

FURPS/FURPS+ model kakovosti:

16

- Isti pristop kot prejšnja modela
- Grady – (IBM) Rational Software
- Zajema širši spekter programskih izdelkov
- Dve vrsti kakovosti:
 - Funkcionalna kakovost:
 - Funkcionalnost (functionality): zmožnost, varnost
 - Nefunkcionalna kakovost:
 - Uporabnost (usability): človeški dejavniki
 - Zanesljivost (reliability): odpovedi, natančnost, vzdrževalnost ,...
 - Doseganje performans (performance): hitrost, natančnost, ...
 - Podpora (serviceability): razširljivost, vzdrževalnost, ...

CUPRIMDSO model kakovosti:

17

- Podroben pristop kot prejšnji model
- Širši pogled pri določanju kakovosti
- Nabor lastnosti:
 - Zmožnost (capability),
 - Uporabnost (usability),
 - Doseganje performans (performance),
 - Zanesljivost (reliability),
 - Sposobnost nameščanja (installability),
 - Vzdrževalnost (maintelability),
 - Dokumentiranost (documentation),
 - Podpora (Service),
 - Skupna kakovost (Overall)

Dromeyev model kakovosti:

18

- Splošnejši pogled - ne samo za izdelke
- Razmerje: lastnosti kakovosti – atributi, ki jo določajo:
 - Pravilnost (correctness):
 - Funkcionalnost, Zanesljivost
 - Notranje lastnosti izdelka:
 - Vzdrževalnost, Učinkovitost, Zanesljivost,
 - Lastnosti okolja:
 - Vzdrževalnost, Ponovna uporabljivost, Prenosljivost, Zanesljivost
 - Opisne lastnosti izdelka:
 - Vzdrževalnost, Ponovna uporabljivost, Prenosljivost, Zanesljivost

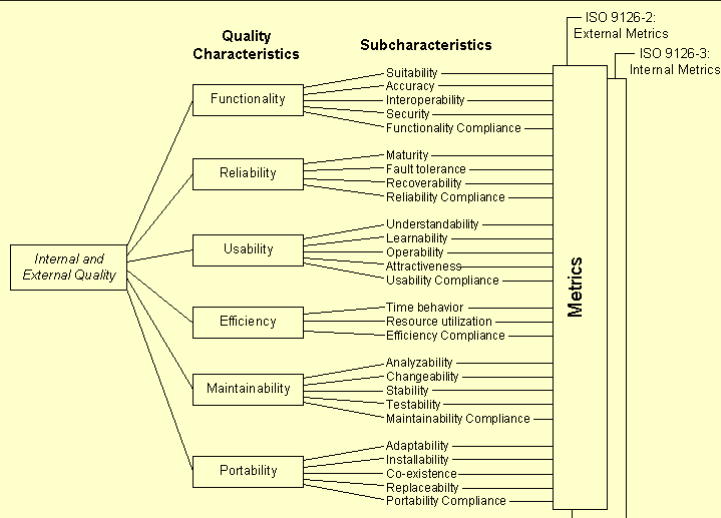
ISO/IEC 9126 model:

19

- Software Product Evaluation: Quality Characteristics and Guidelines
- Enaka zasnova kot McCall in Boehm
- Posplošitev za (nove) različne vrste programskih izdelkov
- 6 faktorjev - značilnosti kakovosti:
 - **Funkcionalnost**: ali podpira zahtevane funkcije?
 - **Zanesljivost**: ali deluje v skladu s pričakovanji?
 - **Uporabnost**: je enostaven za uporabo?
 - **Učinkovitost**: kako učinkovito deluje?
 - **Vzdrževalnost**: je enostavno nadgrajevati, vzdrževati?
 - **Prenosljivost**: kako težko je prenesti sistem v drugo okolje?
- Vsak od teh je natančneje določen z nekaj izmed skupno 27 podznačilnosti
- Zunanje in notranje metrike

ISO 9126 model:

20



© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

ISO/IEC 250xx model:

21

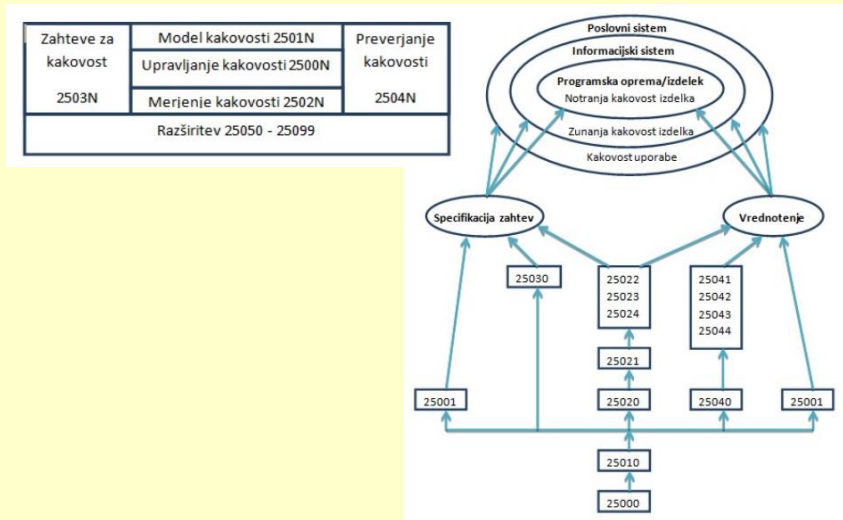
- **SQuaRE - Software product Quality Requirements and Evaluation**
- Posodobitev – razširitev modela ISO 9126
- Pravzaprav družina standardov:
 - ISO/IEC 2500n – upravljanje kakovosti
 - **ISO/IEC 2501n – opis modela kakovosti**
 - ISO/IEC 2502n – merjenje kakovosti
 - ISO/IEC 2503n – zahteve za kakovost
 - **ISO/IEC 2504n –preverjanje kakovosti** (postopek, modeli vrednotenja)
- Enaka zasnova:
 - značilnosti, podznačilnosti, atributi, (zunanje in notranje) metrike
- Dva modela:
 - a) kakovost izdelka (8 značilnosti, skupno 31 podznačilnosti)
 - b) kakovost uporabe sistema (4 značilnosti)

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

ISO/IEC 250xx model:

22



© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

ISO/IEC 25010 model:

23

a) Model kakovosti izdelka (Software product quality model)

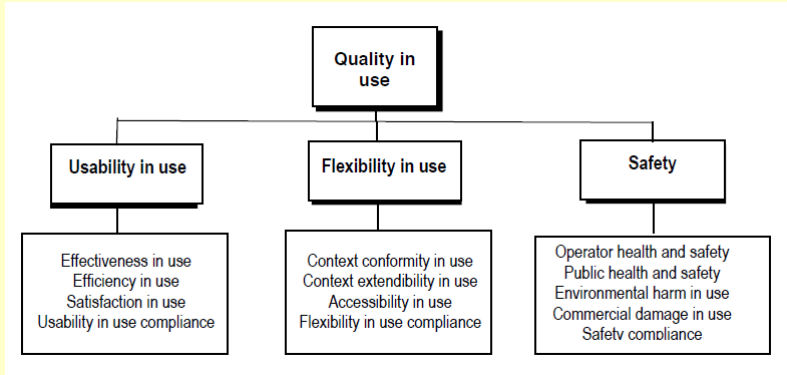
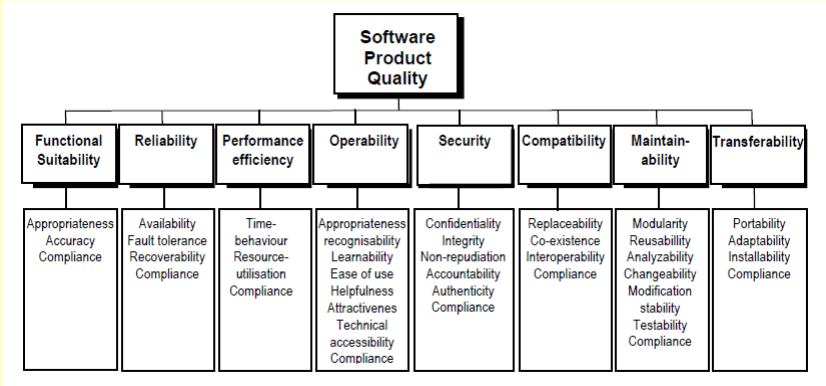
- Funkcijska ustreznost (Functional suitability)
- Zanesljivost (Reliability)
- Učinkovitost izvedbe (Performance efficiency)
- Operativnost (Operability)
- Varnost (Security)
- Usklajenost (Compatibility)
- Vzdrževalnost (Maintainability)
- Prenosljivost (Transferability)

b) Model kakovosti sistema pri uporabi (System quality in use model)

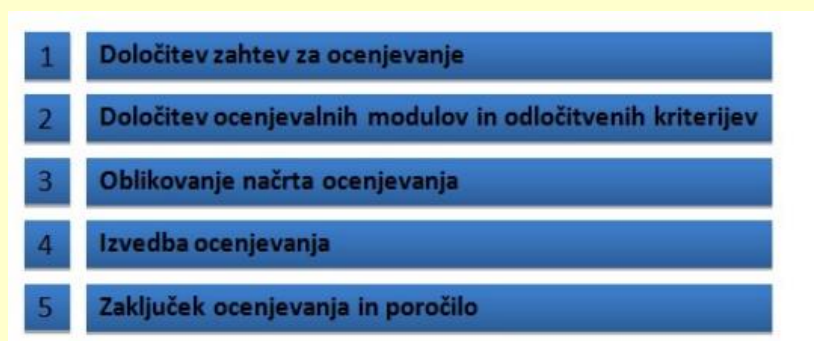
- Kakovost pri uporabi (Quality in use)
- Uporabnost pri uporabi (Usability in use)
- Prilagodljivost pri uporabi (Flexibility in use)
- Varnost pri uporabi (Safety)

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Preverjanje kakovosti



1. Določitev zahtev za ocenjevanje

- namen vrednotenja
- pridobitev zahtev za kakovost
- določitev delov za vrednotenje
- strogost ocenjevanja

2. Ocenjevalni moduli in odločitvenih kriterijev

- izbira meril kakovosti
- določitev odločitvenih kriterijev kakovosti
- določitev odločitvenih kriterijev za ocenjevanje

3. Oblikovanje načrta za ocenjevanje

4. Izvedba vrednotenja

- izvedba meritev
- uporaba odločitvenih kriterijev za merila
- uporaba odločitvenih kriterijev za ocenjevanje

5. Zaključek ocenjevanja in poročilo

- pregled rezultatov
- izdelava poročila
- preučitev izvedbe s povratno informacijo
- dispozicija in arhiviranje podatkov

Paradigma TQM: dolgoročen uspeh je mogoče doseči le tako, da pri upravljanju stremimo h kakovosti izdelkov, kot jo razumejo uporabniki

Štirje stebri TQM:

- a) Usmerjenost na uporabnika
 - b) Izboljševanje procesa razvoja PO
 - c) Osrednja vloga ljudi pri zagotavljanju kakovosti v organizaciji
 - d) Analiza in merjenje
- +
- e) Učinkovito vodenje s poudarkom na usposabljanju, moderni infrastrukturi in planiranju

Zagotavljanje kakovosti programske opreme:

skupek aktivnosti, metod in postopkov, ki nas vodijo po zanesljivi poti do kakovostnega programskega izdelka.

zagotavljanje kakovosti \neq preverjanje kakovosti



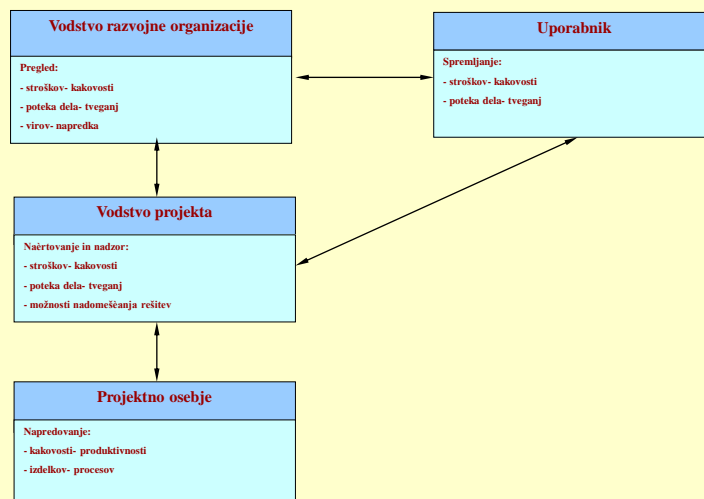
Kakovost programske opreme je direktna posledica kvalitetnega razvojnega procesa kot celote



Zato moramo mersko spremljati:

- izdelke (prog. specifikacije, kodo, dokumentacijo, podatke,...)
- proces razvoja programske opreme (z vidika postopkov)
- projekt (z vidika porabe virov)

Različni udeleženci imajo različne želje glede meritev... 31



Meritve morajo biti ustrezno operativno definirane 32

2 pogoja:

- **Komunikativnost:** kdorkoli natančno razume, kaj je bil namen meritev ter kakšne so bile predpostavke in okoliščine meritev

- **Ponovljivost:** kdorkoli lahko ponovi meritev in dobi enake rezultate

V ta namen obstajajo formalne metode na osnovi vprašalnikov (ang. checklist).

Primer: Data Definition Framework (DDF).

Vloga meritev v praksi 33

... v vsakdanjem življenju

... v tehničnih disciplinah (znanosti):

- zelo pomembna, običajno jasno opredeljena vloga
- primer: Ohmov zakon

... in v programski opremi:

- pomembna, vendar obstajajo težave na tem področju
- zaradi tega pogosto zanemarjeno področje
- v zadnjem času pridobiva na pomenu

1) težave pri določanju merljivih ciljev

- primer: uporabniku prijazna PO
- Princip Toma Gilba: "Projekti brez jasno določenih ciljev bodo dosegli nejasne cilje"

2) težave pri določanju stroškov razvoja

- primer: stroški faze načrtovanja, kodiranja, testiranja

3) težave pri določanju in spremljanju kakovosti

- primer: zanesljivost PO ob spremembi delovne platforme



4) privrženost "revolucionarnim" tehnološkim spremembam

- brez dejanskih dokazov, kaj ta sprememba v izbranem okolju pomeni

5) težave pri zagotavljanju konsistentnosti

- primer: merjenje števila vrstic programske kode

6) težave zaradi nepopolnih meritev

- primer: spremljanje števila napak ob odpovedih

... zaradi tega so bile v preteklosti ...

36

- Učinkovite meritve PO običajno lepe, neizpolnjene želje
- Skrite v množici aktivnosti, kot so:

Metode in orodja za ovrednotanje
Upravljanje z metrikami
Zmožnostno-zrelostno ocenjevanje
Metrike strukture in kompleksnosti
Modeli ovrednotenja performans
Modeli zanesljivosti
Metrike in modeli kakovosti
Zbiranje podatkov
Metrike in modeli učinkovitosti
Ocenjevanje stroškov in napora

... vendar obstajajo tehtni razlogi ZA meritve PO

37

Trije razlogi ZA merjenje:

Razumevanje PO
Upravljanje projektov razvoja PO
Izboljševanje procesa razvoja PO

Danes so meritve v TPO aktivno področje raziskav

Razumevanje PO:

38

Najpomembnejši razlog: predpogoj za izvedbo ostalih aktivnosti

Želimo spoznati proces razvoja PO in izdelke ter jih primerno opisati z razmerji med posameznimi elementi

Običajen pristop:

- Z merjenjem lastnosti spoznamo značilnosti procesa razvoja PO in izdelkov
- Analiziramo izmerjene vrednosti
- Zgradimo model, ki ustrezno predstavlja razmerja med značilnostmi
- Na osnovi modela načrtujemo nadaljnje ukrepe

Razumevanje PO: Primer

39

Želimo razumeti, koliko stane razvoj programske opreme v našem podjetju ...

Ključne značilnosti:

- Delitev stroškov za posamezne faze razvoja PO
(načrtovanje, kodiranje, testiranje)
- Povprečen strošek vrstice programske kode
- Strošek vzdrževanja vrstice programske kode
- Potreben napor za pisanje dokumentacije na vrstico programske kode
- Stroški obrabe strojne opreme
- Pričakovani stroški nameščanja programskega paketa
- ipd.

Upravljanje projektov razvoja PO

40

Razumevanje nam omogoča učinkovit nadzor in upravljanje projektov razvoja PO

Osredotočimo se na uporabo ugotovitev (in ne na zbiranje podatkov)

Poznavanje sistema uporabimo za:

- Načrtovanje in napovedovanje elementov projekta (stroški, napor, ...)
- Spremljanje meritev značilnosti projekta glede na načrtovane vrednosti
- Vrednotenje organizacijskega modela kot osnove za napovedovanje

Upravljanje projektov razvoja PO: Primer

41

Če poznamo: Obseg dela (v urah) = DLOC / Učinkovitost

Učinkovitost = 3.2 DLOC na uro

Čas trajanja (v mesecih) = $4.9 * \text{Obseg dela [v človek-mesecih]}^{0.3}$



Potem za razvoj 99 000 DLOC programske kode potrebujemo:

- 200 človek-mesecev
- 24 mesecev

V ostalih dveh delih spremljamo ujemanje izmerjenih vrednosti z napovedmi ter usklajenost rezultatov z modelom

Izboljševanje procesa razvoja PO

42

Smislen cilj vsake organizacije za razvoj PO je sposobnost izdelave čim kakovostnejših izdelkov s predvidenimi viri

- Za izboljševanje izdelka je treba izboljševati proces razvoja izdelka
- Izboljšanje procesa lahko dosežemo z izboljšavami v upravljanju ali uvajanjem novih tehnologij

V obeh primerih je merjenje nepogrešljiv del

Izboljševanje procesa razvoja PO: Primer

43

Uporaba modela CMM(I) za zagotavljanje kakovosti razvoja PO

V CMM je bilo merjenje prisotno povsod, a nikjer eksplicitno v posebnem ključnem področju:

- skupina aktivnosti Merjenje in analiza
- postopki za zagotavljanje kakovosti
- testiranje, dokumentiranje

CMMI uvede merjenje kot posebno ključno področje

- že na 2. nivoju zrelosti
- 3 cilje je treba doseči

Cilj 1: Uskladi aktivnosti merjenja in analize

- določi merske cilje;
- določi meritve;
- določi zbiranje podatkov in postopke hranjenja;
- določi postopke analize.

Cilj 2: Poskrbi za rezultate merjenja

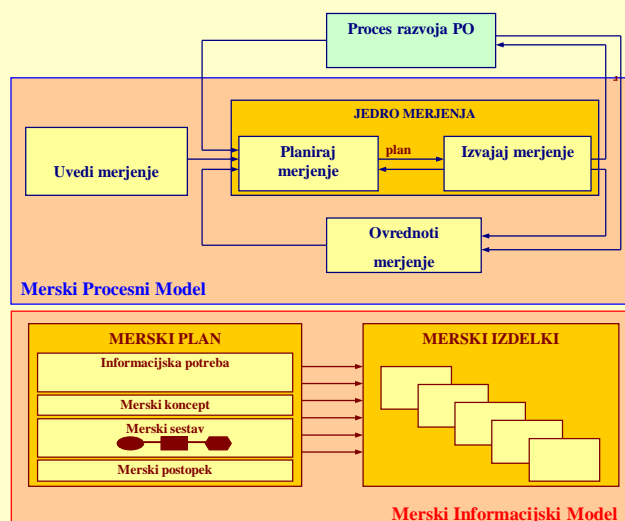
- zbiraj merske podatke;
- analiziraj merske podatke;
- shranjuj podatke in rezultate;
- posreduj rezultate.

Cilj 3: Institucionaliziraj merski proces

- vzpostavi pristop na ravni organizacije;
- planiraj merski proces;
- priskrbi vire;
- določi odgovornosti;
- usposobi izvajalce;
- upravljaj konfiguracije;
- identificiraj in upoštevaj pomembne lastnike;
- spremljaj in nadzoruj proces;
- objektivno ocenjuj postopke;
- preverjaj status z vodstvom.

Merski procesni model:

- razvil Joint Group for Systems Engineering, Newport, 1998
- ustreza zahtevam standarda ISO/IEC 15939 in modela CMMI
- opisan z:
 - merskim procesnim modelom in
 - merskim informacijskim modelom
- primeren način merjenja za majhne organizacije
- merjenje definira (predvsem) kot aktivnost na ravni projekta

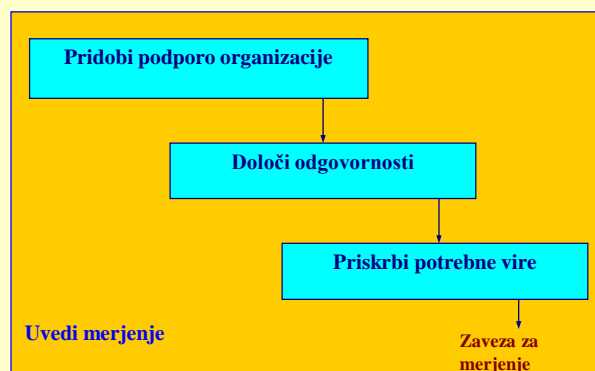


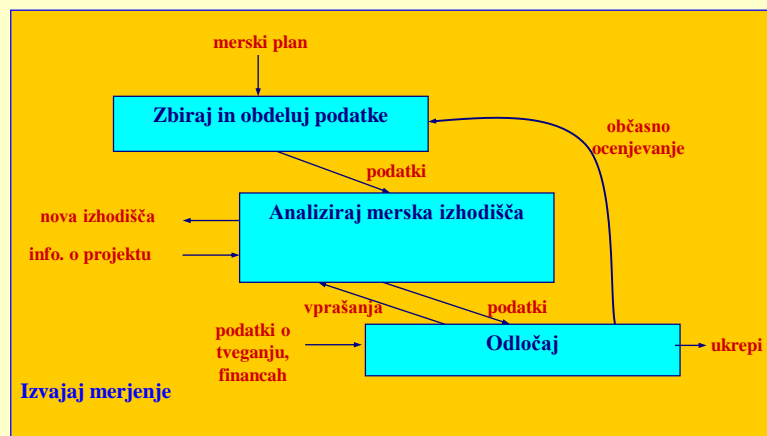
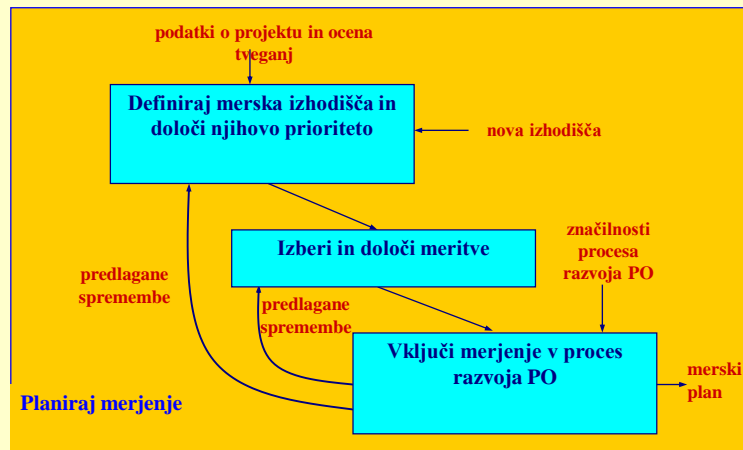
Merski procesni model

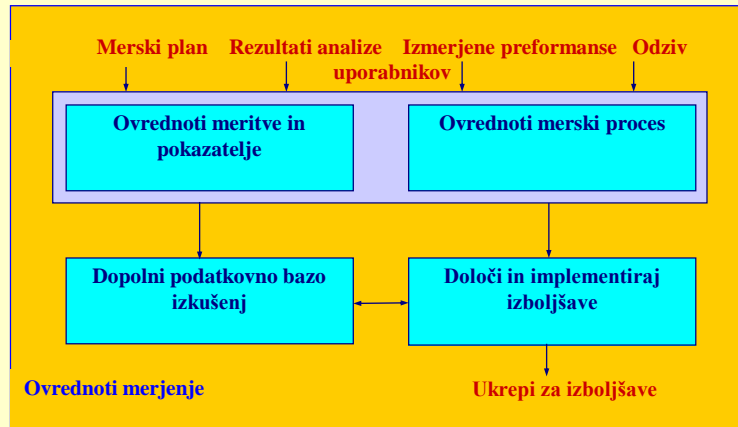
- definira zaporedje aktivnosti za učinkovito merjenje procesa razvoja
- bistvo: merski proces je sestavni del procesa razvoja

Štiri glavne aktivnosti:

- uvedi merjenje
- planiraj merjenje
- izvajaj merjenje
- ovrednoti merjenje

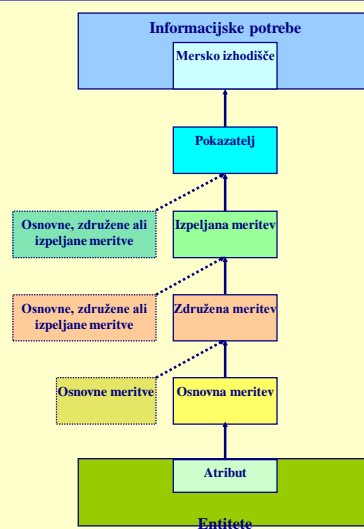
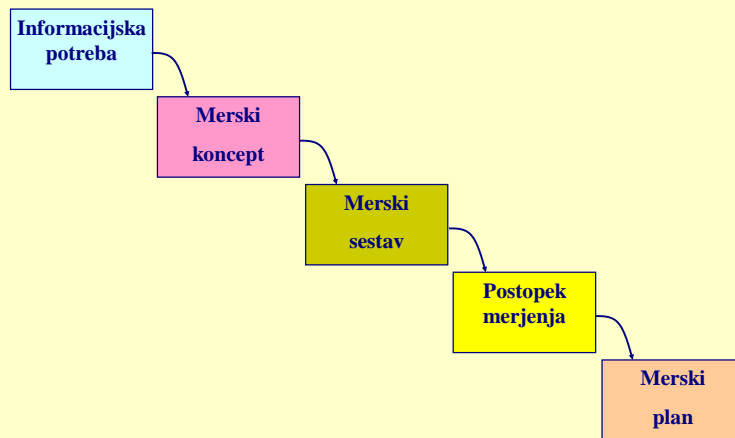






Merski informacijski model

- hierarhičen način opisa merjenja in merskih podatkov
- mehanizem za opis povezave merskih izhodišč z meritvami
- definira jasno strukturo merskih elementov, ki omogočajo natančen opis vseh potrebnih vidikov merjenja
- celoten opis je združen v merskem planu



Tehnologija programske opreme pokriva vrsto dejavnosti na področju sistematičnega razvoja, uporabe in vzdrževanja programske opreme

Meritve sodijo na področje zagotavljanja kakovosti programske opreme

Meritve v TPO merijo izdelke, proces razvoja PO in projekt

Meritve morajo biti ustrezno operativno definirane, odvisne so od gledišča uporabnika.

V primerjavi z drugimi tehnološkimi disciplinami se pri meritvah PO pojavlja vrsta težav, kar je v preteklosti botrovalo manjši uporabi.

Trije razlogi za merjenje:

- RAZUMEVANJE PO
- UPRAVLJANJE PROJEKTOV RAZVOJA PO
- IZBOLJŠEVANJE PROCESA RAZVOJA PO

V zadnjem času so se raziskovalci zavedli pomembnosti področja in obstaja vrsta raziskav.

Sodoben način: merjenje z upoštevanjem standardov in modelov:

- pomembni ISO standardi
- v sodobnih modelih (CMMI) je merjenje ustrezno vkomponirano
- implementacija merjenja - PSM :
 - merski proces
 - merski informacijski model

1. N. Fenton, S. L. Pfleeger: **Software Metrics, A Rigorous and Practical Approach** (PWS Publishing Company, 1997)
2. NASA: **Software Measurement Guidebook** (Software Engineering Laboratory Series SEL-94-102, 1995)
3. I.Sommerville: **Software Engineering, Eight Edition** (Addison Wesley, 2007)
4. ISO: <http://www.iso.org>
5. McGarry, J. et al: **Practical Software Measurement: Objective Information for Decision Makers** (Addison-Wesley, 2001).
6. PSM: <http://www.psmc.com>