Softverske metrike

Univerzitet u Beogradu
Fakultet organizacionih nauka
Dr Miloš Milić

Sadržaj

- Kvalitet softvera
- ISO/IEC 9126 standard
- ISO/IEC 14598 standard
- ISO/IEC 25000 standard
- Softverske metrike
- Zaključak

Kvalitet softvera

- Raznovrsna primena softverskih sistema u današnjem poslovanju
- Razvoj softvera je proces koji je veoma specifičan i složen
- Softver isporučiti brže, po prihvatljivoj ceni, sa minimalnim brojem defekata [Kruchten04] [Sommerville04]
- Modeli, strategije i aktivnosti

Kvalitet softvera

- “Usaglašenost proizvoda sa detaljnom specifikacijom proizvoda” [Crosby80]
- Kvalitet datog proizvoda je ponekad definisan kao “totalitet karakteristika (proizvoda ili usluga) koja nosi sa sobom sposobnost da zadovolji implicitne potrebe”
- “Kvalitet softvera znači usaglašenost sa zahtevima” [Letouzey12]
- Kvalitet softvera se često definiše kao “efikasna, efektivna i prigodna upotreba od strane datih korisnika za određenu svrhu pod određenim uslovima”

Kvalitet softvera

- “Kvalitet softvera predstavlja efikasan softverski proces čiji je rezultat koristan softverski proizvod koji pruža merljive vrednosti za one koji ga proizvode i one koji ga koriste” [Pressman10]
- Kvalitet softvera u značajnoj meri zavisi od nefukcionalnih zahteva [Sommerville11]

Kvalitet softvera

- Za svaki inženjerski proizvod postoje mnogi željeni kvaliteti
- D. Garvin, Harvard Business School
  - Transcedentalni pogled
  - Korisnički pogled
  - Pogled proizvođača
  - Pogled na proizvod
  - Pogled zasnovan na vrednostima

Kvalitet softvera

- Potrebe korisnika: određeni nivo kvaliteta, a ne samo funkcionalnost
- Važno je imati imati na umu neke moguće atribute kvaliteta softvera
- Terminologija za atribute kvaliteta se razlikuje od jednog modela softverskog kvaliteta do drugog, svaki model može imati različiti broj hijerarhijskih nivoa i različit ukupan broj atributa

Kvalitet softvera

- Sa svakim atributom kvaliteta povezan je veći broj **softverskih metrika**
- Određivanje mera kojima se prikazuju k-ke softverskog projekta
- Kvantitativni indikatori
- Stalni proces

Kvalitet softvera

- "Metrika predstavlja kvantitativnu meru do koje sistem, komponenta ili proces poseduje posmatrani atribut." [ISO24765]
- "Metrika predstavlja utvrđen metod merenja i mernu skalu." [ISO14598v1]
- "Softverska metrika je karakteristika softverskog sistema, dokumentacije sistema ili procesa razvoja sistema koja se može objektivno izmeriti." [Sommerville11]

Kvalitet softvera

- Karakteristike SM: formalizovana, prati promene karakteristike softverskog sistema, empirijski potvrđena, objektivna, proverljiva i predvidljiva, nezavisna od programskog jezika u okviru kojeg se vrši implementacija softverskog sistema, veličine softverskog sistema, vrste softverskog sistema, domena problema
- Pri procesu utvrđivanja vrednosti softverskih metrika mogu se koristiti različite merne skale [Bourque14] [ISO24765]: nominalna, ordinalna, intervalna, skala odnosa

Kvalitet softvera

- Planiranje kvaliteta softvera uključuje:
  1. Definisanje određenog proizvoda u smislu njegovih atributa kvaliteta
  2. Planiranje procesa kako bi se dostigao odgovarajući proizvod
- Menadžment softverskog inženjerstva, Projektovanje softvera, Alati i metode softverskog inženjerstva [Swebok]
- ISO standardi

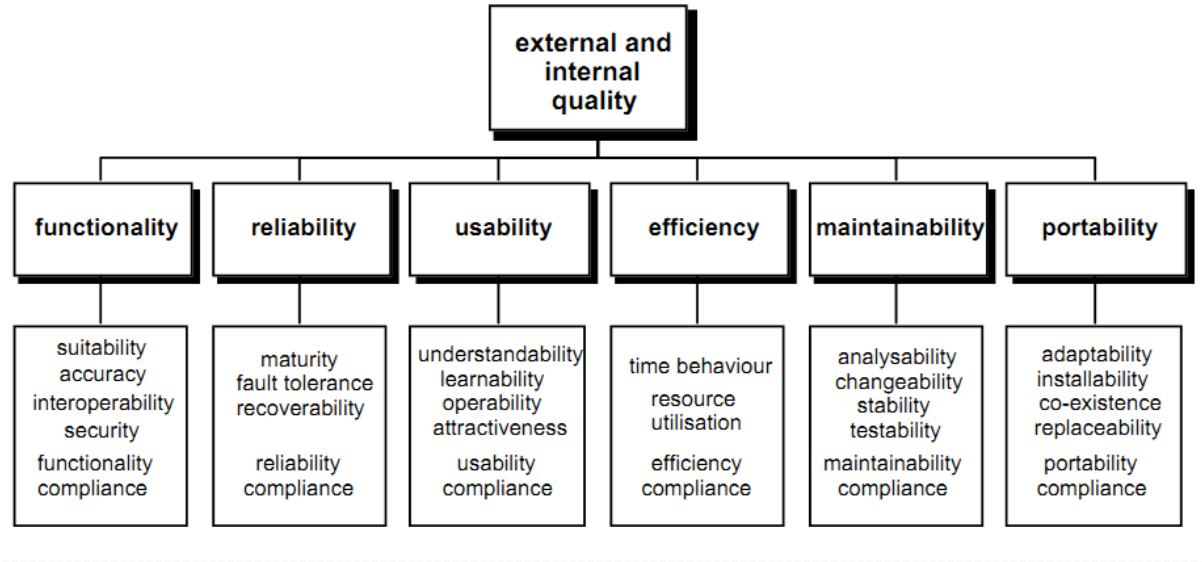
ISO/IEC 9126 standard

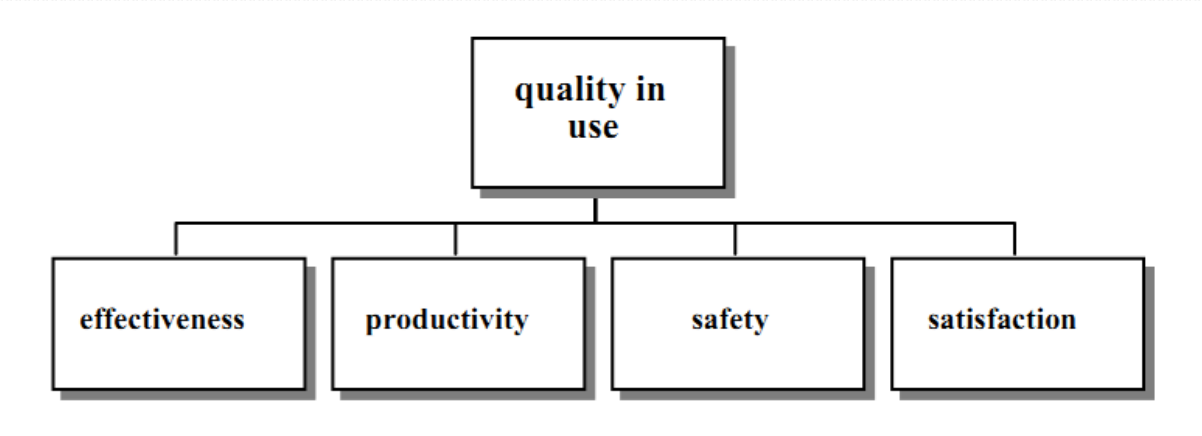
- Međunarodni standard za ocenu kvaliteta softvera
- Sastoji se iz četiri dela:
  1. ISO/IEC 9126-1, koji proučava model kvaliteta softvera (quality model)
  2. ISO/IEC 9126-2, koji proučava eksterne softverske metrike (external metrics)
  3. ISO/IEC 9126-3, koji proučava interne softverske metrike (internal metrics)
  4. ISO/IEC 9126-4, koji proučava upotrebni kvalitet softverskih metrika (Quality in use metrics)

ISO/IEC 9126 standard

- Prva tri dela se odnose na opisivanje i merenje kvaliteta softverskog proizvoda dok se četvrti deo odnosi na evaluaciju
- ISO/IEC 9126-1 opisuje model kvaliteta softvera i sastoji se iz dva dela:
  1. Interni i eksterni kvalitet
  2. Upotrebni kvalitet

ISO/IEC 9126 standard





ISO/IEC 9126 standard

Grupa	Atributi	Ključna pitanja
Funkcionalnost (Functionality)	Pogodnost (suitability) Tačnost (accuracy) Interoperabilnost (interoperability) Sigurnost (security) Mogućnost usaglašavanja funkcionalnosti sa standardima (functionality compilance)	Da li softver obavlja sve zahtevane zadatke? Da li su rezultati tačni? Da li sistem može da vrši interakciju sa drugim sistemima? Da li sistem sprečava neautorizovan pristup? Da li je sistem u skladu sa standardima?
Pouzdanost (Reliability)	Zrelost (maturity) Tolerantnost na greške (fault tolerance) Mogućnost povratka (recoverability) Mogućnost usaglašavanja pouzdanosti sa standardima (reliability compilance)	Da li je većina softverskih grešaka eliminisana kroz vreme? Da li je sistem sposoban da upravlja greškama? Da li sistem može nakon pada, da nastavi sa radom i povrati izgubljene podatke?
Upotrebljivost (Usability)	Razumljivost (understendability) Mogućnost učenja (learnability) Operativnost (operability) Atraktivnost (attractiveness) Mogućnost usaglašavanja upotrebljivosti sa standardima (usability compilance)	Da li korisnik može lako shvatiti kako da koristi sistem? Da li korisnik može brzo naučiti da koristi sistem? Da li korisnik može koristiti softver bez previše napora? Da li je korisnički interfejs privlačnog dizajna?

ISO/IEC 9126 standard

Grupa	Atributi	Ključna pitanja
Efikasnost (Reliability)	Vremensko ponašanje (time behaviour) Upotreba resursa (resource utilisation) Mogućnost usaglašavanja efikasnosti sa standardima (reliability compilance)	Koliko brzo sistem reaguje? Da li sistem efikasno upravlja resursima?
Održavanje (Maintability)	Mogućnost analize (analysability) Mogućnost promena (changeability) Stabilnost (stability) Mogućnost testiranja (testability) Mogućnost usaglašavanja održavanja sa standardima (maintability compilance)	Da li se sistemske greške mogu lako utvrditi? Da li je sistem jednostavan za izmene? Da li sistem može nastaviti sa funkcionisanjem tokom izmena? Da li je omogućeno lako testiranje softvera?
Prenosivost (Portability)	Adaptivnost (adaptability) Mogućnost instalacije (installability) Zajedničko postojanje (co-existence) Mogućnost zamene (replaceability) Mogućnost usaglašavanja prenosivosti sa standardima (portability compilance)	Da li se softver može preneti u drugo okruženje? Da li se softver lako može instalirati? Da li softver može lako da zameni drugi softver? Da li je softver u sagalsnosti sa standardima prenosivosti?

ISO/IEC 9126 standard

- **ISO/IEC 9126-2** standard definiše eksterne metrike za merenje kvaliteta softvera u smislu karakteristika i podkarakteristika definisanih u ISO/IEC 9126-1 delu i namenjen je za korišćenje zajedno sa njim
- Mogu biti korisne u definisanju korisničkih zahteva kao i ocenjivanju razvijenog softverkog proizvoda.
- Mogu biti od velike koristi kako programerima, tako i inženjerima koji se bave kvalitetom softvera i naručiocima softverskog sistema

ISO/IEC 9126 standard

- Za svaki atribut softverskog sistema koji je definisan u standardu ISO/IEC 9126-1 se definišu metrike koje sadrže: naziv metrike, svrhu metrike, formulu i interpretaciju izmerene vrednosti.

Naziv metrike	**Svrha metrike**	**Formula**	**Interpretacija izmerene metrike**
Mogućnost promene preko parametara (Parameterised modifiability)	Da li se u softverskom sistemu mogu lako izvršiti izmene ili rešiti određeni problem putem promene određenog parametra?	X=1-A/B, gde je: A-broj slučajeva u kojima se softver ne može promeniti preko parametara. B-broj slučajeva u kojima se pokušava promena sistema preko parametara	0<=X<=1 Bolje je ukoliko X teži 1 i to znači da je mogućnost promene softvera preko parametara veća.

ISO/IEC 9126 standard

- Kroz **ISO/IEC 9126-3** standard se definišu interne softverske metrike, koje se mogu primeniti na softverski sistem u toku projektovanja i pisanja programskog koda.

Naziv metrike	**Svrha metrike**	**Formula**	**Interpretacija izmerene metrike**
Pamćenje promena (Change recordability)	Da li su promene u specifikaciji softverskog sistema i promene u programskim modulima na odgovarajući način zabeležene u programskom kodu, sa odgovarajućim komentarima?	X=A/B, gde je: A-broj promena u funkcijama/modulima koji imaju promenjene komentare, potvrđene u reviziji. B-Ukupan broj funkcija/modula koji su promenjeni.	0<=X<=1 Bolje je ukoliko X teži 1 i to znači da su promene bolje zabeležene. Ukoliko vrednost X teži 0, promene su slabije zabeležene ili je izvršeno malo promena, što opet može ukazati na visoku stabilnost softverskog sistema.

ISO/IEC 9126 standard

- **ISO/IEC 9126-4** standard definiše upotrebni kvalitet (quality in use metrics)
- Na ovaj način se može utvrditi da li softverski proizvod zadovoljava specifične potrebe korisnika kako bi se postigli određeni ciljevi u pogledu efektivnosti, produktivnosti, bezbednosti i zadovoljstva korisnika
- Metrike za utvrđivanje efektivnosti, produktivnosti, sigurnosti i zadovoljenja

ISO/IEC 9126 standard

Naziv metrike	**Svrha metrike**	**Formula**	**Interpretacija izmerene vrednosti**
Error frequency	Potrebno je utvrditi frekvenciju grešaka koje se mogu javiti pri korišćenju softvera od strane korisnika.	X=A/T A=Broj grešaka koje napravi korisnik pri korišćenju softvera. T=vreme ili broj zadataka koje izvršava korisnik.	X>=o Bolje je ukoliko vrednost X teži o. U tom slučaju je broj grešaka koje napravi korisnik pri korišćenju softvera manji.

ISO/IEC 14598 standard

- Standardom ISO/IEC 14598 definiše se način evaluacije kvaliteta softvera pri čemu se, kao osnova za evaluaciju kvaliteta, koristi ISO/IEC 9126 standard kvaliteta softvera
  - ISO/IEC 14598-1:1999 Information technology - Software product evaluation - Part 1: General overview
  - ISO/IEC 14598-2:2000 Software engineering - Product evaluation - Part 2: Planning and management
  - ISO/IEC 14598-3:2000 Software engineering - Product evaluation - Part 3: Process for developers
  - ISO/IEC 14598-4:1999 Software engineering - Product evaluation - Part 4: Process for acquirers
  - ISO/IEC 14598-5:1998 Information technology - Software product evaluation - Part 5: Process for evaluators
  - ISO/IEC 14598-6:2001 Software engineering - Product evaluation - Part 6: Documentation of evaluation modules

ISO/IEC 25000 standard

- Standardi ISO/IEC 9126 (Software product quality) i ISO/IEC 14598 (Software product evaluation)
  - Imaju zajednički normativ
  - Predstavljaju skup komplementarnih standarda
  - Nezavisni životni ciklus doveo je do njihove nekozistentnosti

ISO/IEC 25000 standard

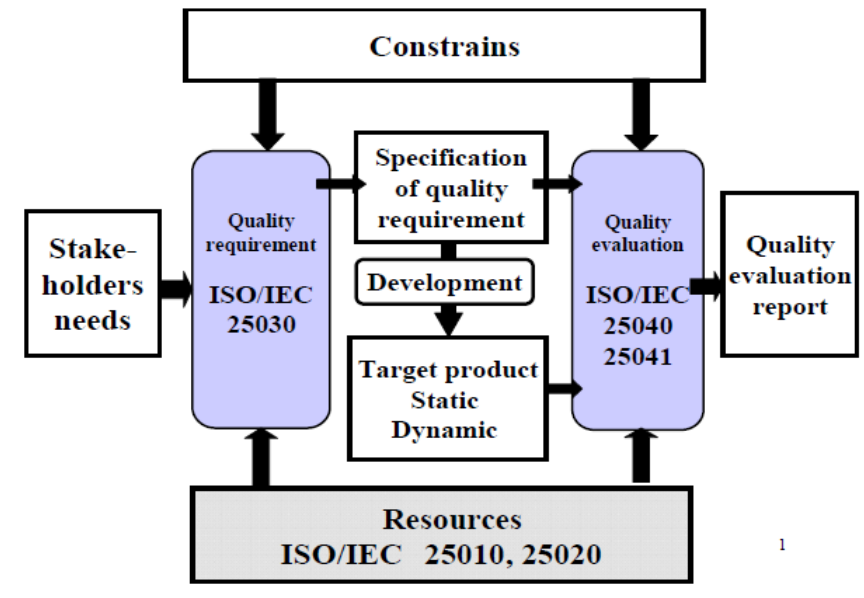
- ISO/IEC 2500 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) je naslednik ISO/IEC 9126 i ISO/IEC 14598
- Cilj: kreirati logički dobro organizovanu, obogaćenu i jedinstvenu seriju standarda koja obuhvata:
  - Specifikaciju zahteva kvaliteta softvera
  - Evaluaciju kvaliteta softvera
  - Podršku kroz proces za merenje kvaliteta softvera

ISO/IEC 25000 standard

- Struktura standarda:
  - ISO/IEC 2500n, Quality Management Division
  - ISO/IEC 2501n, Quality Model Division
  - ISO/IEC 2502n, Quality Measurement Division
  - ISO/IEC 2503n, Quality Requirements Division
  - ISO/IEC 2504n, Quality Evaluation Division
- ISO/IEC 25050 - ISO/IEC 25099 – rezervisani su za SQuaRE ekstenzije

ISO/IEC 25000 standard

- K. ESAKI, System Quality Requirement and Evaluation, Importance of application of the ISO/IEC25000 series, Global Perspectives on Engineering Management, May 2013, Vol. 2 Issue 2, PP. 52-59



ISO/IEC 25000 standard

- Model kvaliteta prema ISO/IEC 25010

ISO/IEC 25000 standard

- Pojedini standardi su u fazi izrade (npr. ISO/IEC DIS 25022, ISO/IEC DIS 25023 i ISO/IEC DIS 25024 kojima se specificiraju softverske metrike za evaluaciju upotrebnog kvaliteta, eksternog i internog kvaliteta i kvaliteta podataka, respektivno)
- ISO/IEC 25040 standard definiše faze evaluacije kvaliteta softvera

Softverske metrike

- Statička analiza bez pokretanja (izvršavanja) softverskog sistema
- Alati za statičku analizu:
  - Swat4J
  - SonarQube
  - FindBugs
  - Coverity
  - …

Softverske metrike

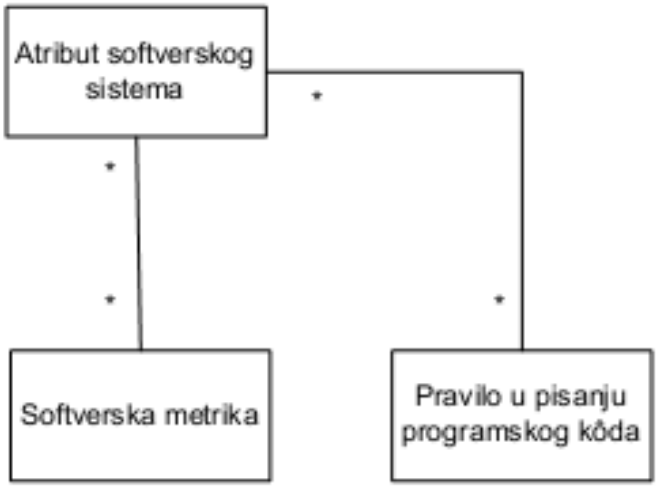
- Swat4j je softverski paket namenjen praćenju i upravljanju aktivnostima razvoja i održavanja softverskog sistema koji je napisan u programskom jeziku Java
- Zasnovan je na principima standarda ISO/IEC 9126-1 (Quality model) i ISO/IEC 9126-3 (Software Product Quality, Internal Metrics)

Softverske metrike

- Atributi kvaliteta softvera
  - Testiranje (*Testability*)
  - Kvalitet projektovanja (*Design quality*)
  - Performanse (*Performance*)
  - Razumljivost (*Understandability*)
  - Održavanje (*Maintainability*)
  - Ponovno korišćenje (*Reusability*)

Softverske metrike

- Pronalaženje grešaka (bugs) u programu
- Integrisano preko 30 metrika i preko 100 standarda koji se odnose na pravila najbolje prakse u pisanju programa (Best practice rule)



Softverske metrike

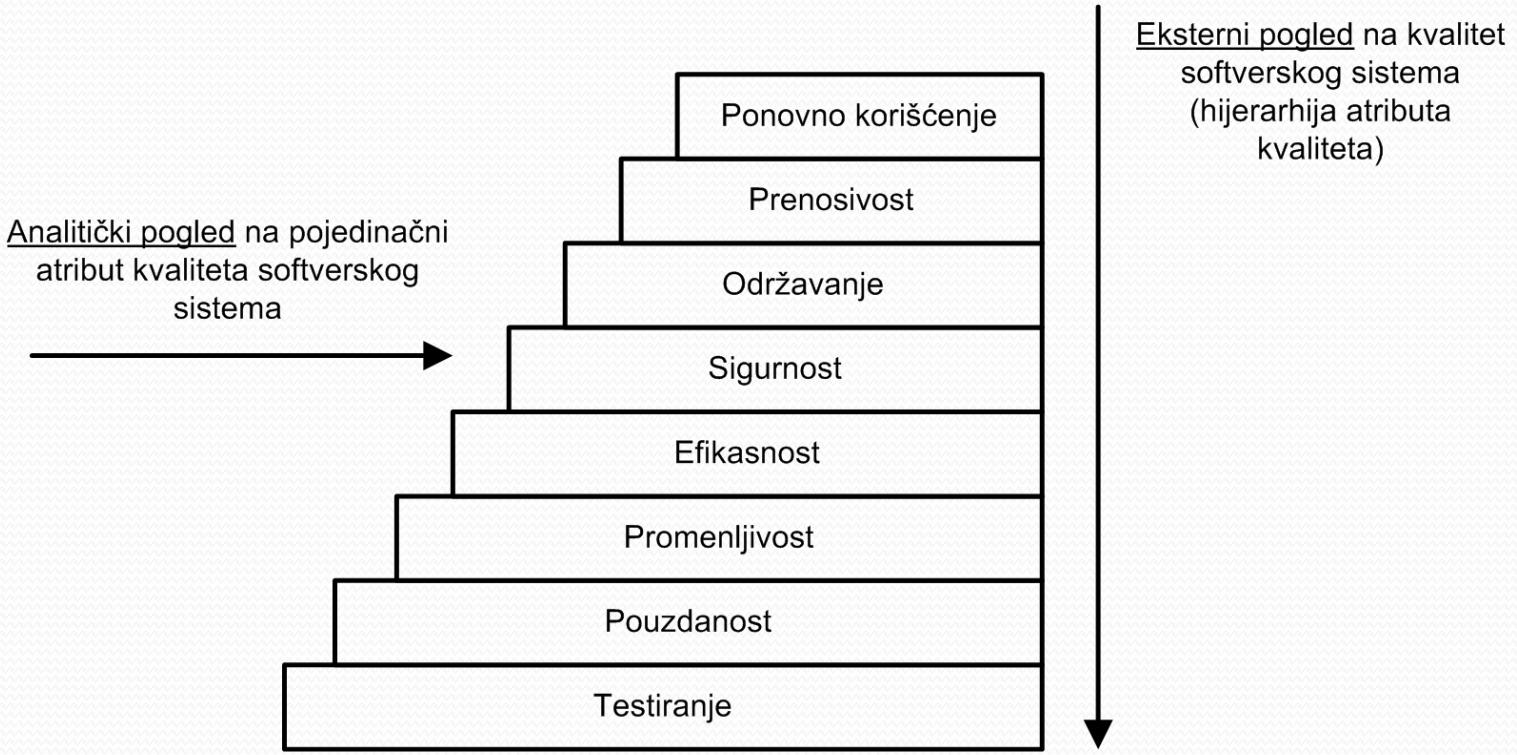
- SonarQube je alat za statičku analizu kvaliteta softverskog sistema
- Podrška za različite programske jezike, izrada dodataka
- Open source projekat
- Zasnovan je na ISO/IEC 9126 modelu kvaliteta softvera (eksterni i interni kvalitet softvera) – SQALE model kvaliteta softvera [Letouzey12]

Softverske metrike

- Atributi kvaliteta softver a u SQALE modelu kvaliteta:
  - Testiranje (Testability)
  - Pouzdanost (Reliability)
  - Promenljivost (Changeability)
  - Efikasnost (Efficiency)
  - Sigurnost (Security)
  - Održavanje (Maintainability)
  - Prenosivost (Portability)
  - Ponovno korišćenje (Reusability)

Softverske metrike

- Atributi kvaliteta softver a u SQALE modelu kvaliteta [Letouzey12]



Softverske metrike

- Za svaki atribut kvaliteta vezano je više podkarakteristika.
- S druge strane, za jednu podkarakteristiku vezano je više softverskih metrika koje se u okviru ovog alata nazivaju **pravila**.
- Pravilom se definiše postupak merenja neke vrednosti ali se takođe definišu i granične vrednosti za posmatrano pravilo. Ukoliko izmerena vrednost nije u okviru graničnih vrednosti pravilo je narušeno i obrnuto.

Softverske metrike

- Računanje tehničkog duga (eng. Technical Debt)
- Suma troškova sanacije
- Moguće je izračunati različite indikatore kvaliteta softvera
- (komercijalni dodatak; osnovna verzija u okviru SonarQube alata)

Softverske metrike

- Različite klasifikacije softverskih metrika
- Objektno-orijentisane metrike (Object-oriented metrics)
- Metrike za određivanje složenosti (Complexity metrics)
- Metrike za određivanje indeksa održavanja (Maintainability Index metric)
- Metrike koje se odnose na programski kod (Code metrics)

Softverske metrike

- Objektno-orijentisane metrike - merenje kvaliteta objektnog projektovanja, veza između objekata, kao i drugih principa:
  - Složenost ponderisanih metoda (Weighted Methods Complexity – WMC) – WMC se definiše kao suma složenosti metoda
  - Broj odgovora klase (Response for Class - RFC) – Definiše skup svih metoda koje mogu biti pozvane kao odgovor na poruku objekta klase
  - Nedostatak kohezivnosti metoda u klasi (Lack Of Cohesive Methods) – Predstavlja meru međusobne povezanosti (bliskosti) metoda

Softverske metrike

- Objektno-orijentisane metrike (nastavak):
  - Povezanost objekata (Coupling Between Objects – CBO) - CBO se zasniva na ideji da je objekat povezan sa drugim objektom ukoliko jedan objekat koristi osobine i metode drugog objekta (na primer, metoda prvog objekta koristi metode ili pojavljivanja drugog objekta)
  - Dubina stabla nasleđivanja (Depth of Inheritance Tree – DIT) - definiše se kao maksimalni broj nivoa od posmatranog čvora do korenog (root) elementa
  - Broj podklasa (Number of Children – NOC) - NOC računa broj neposrednih podklasa posmatrane klase/interfejsa u hijerarhiji klasa

Softverske metrike

- Metrike za određivanje složenosti - Složenost sistema ili njegovih komponenti predstavlja težinu razumevanja softverskog sistema ili komponenti softverskog sistema. Razlikujemo:
  - Cikličnu složenost (Cyclomatic Complexity – CC) – Meri se računanjem broja tačaka odlučivanja (decision points) ili uslovnih iskaza (conditional statements) posmatranog programskog jezika
  - Halstedovu složenost (Halstead Complexity Metrics) - Služi za merenje složenosti modula direktno iz izvornog koda programa korišćenjem operatora i operanada

Softverske metrike

- Metrike za određivanje indeksa održavanja - Predstavlja kvantitativnu meru namenjenu merenju i praćenju održavanja
- Metrike koje se odnose na programski kod - Uvid u kvalitet koda. Koriste se u kombinaciji sa spec. metrikama, npr. sa OO metrikama ili sa metrikama za određivanje složenosti. Posmatramo ih na nivou klase, metode, fajla, paketa:
  - Broj linija programskog koda (Lines of Code – NLOC)
  - Procenat komentara (Percentage of comments – POC)
  - Broj promenljivih (Number of Variables - NOV)
  - …

Softverske metrike

- Složenost ponderisanih metoda (WMC)
  - WMC se definiše kao suma složenosti metoda.
  - Ostavljena je mogućnost izbora "složenosti metode" koju treba uzeti u razmatranje
  - Pretpostavimo da imamo klasu C sa metodama M1, M2, …, Mn. Neka je C1, C2, …, Cn složenost metoda, respektivno. Tada je WMC=C1+C2+…+Cn
  - Kao mera složenosti metoda koristi se Ciklična složenost (CC)
  - WMC>=1

Softverske metrike

- Ciklična složenost (CC)
  - Predstavlja meru složenosti primenjenog algoritma
  - Meri se računanjem broja tačaka odlučivanja ili uslovnih iskaza u datom programskom jeziku
  - if, else, for, while, do-while, catch, case, default
  - &, |, &&, ||, ?:
  - CC=P+1, P – broj predikata ili broj uslova ili broj binarnih čvorova, 1 – ulazna putanja f-je
  - CC>=1

Softverske metrike

- Broj odgovora klase (RFC)
  - U OOP objekti komuniciraju razmenom poruka, npr. određena poruka može dovesti do određenog ponašanja objekta na taj način što će pozvati neku njegovu metodu
  - Metode kao odgovor na određene poruke
  - RFC=|RS|, RFC=M+R
  - M – Broj metoda u klasi koje mogu biti pozvane kao odgovor klase, R – Ukupan broj drugih metoda koje se pozivaju od strane metoda klase
  - Ukoliko je RFC veliki, klasa je složenija

Softverske metrike

- Nedostatak kohezivnosti metoda u klasi (LCOM)
  - Ukoliko klasa ima metode koje se izvršavaju nad istim skupom atributa za klasu se kaže da je kohezivna. Kohezija je usmerena na atribute objekta, kao i na metode koje pristupaju atributima.
  - LCOM je mera međusobne povezanosti metoda
  - LCOM = (m – sum(mA) / a) / (m – 1)
  - m – broj metoda, a – broj atributa, mA – broj metoda koje pristupaju atributu a
  - LCOM≥0

Softverske metrike

- Povezanost objekata (CBO)
  - Dve klase su povezane ukoliko metode jedne klase koriste atribute ili metode druge klase
  - CBO dobijamo brojanjem povezanih klasa
  - Prevelika povezanost objekata klasa dovodi do narušavanja modularnog projektovanja i sprečava ponovno korišćenje softverskih komponenti (reusability)
  - Potrebno je svesti povezanost klasa na minimum

Softverske metrike

- Dubina stabla nasleđivanja (DIT)
  - Dubina klase u hijerarhiji nasleđivanja definiše se kao maksimalni broj nivoa od posmatranog čvora do korenog (root) elementa
  - DIT za određenu klasu se dobija računanjem broja nadklasa u hijerarhiji nasleđivanja. Ukoliko ima više nadklasa računanje se ponavlja za sve putanje (DIT = najveći broj nivoa)
  - Preporučljivo je da DIT vrednost bude što manja

Softverske metrike

- Broj podklasa (NOC)
  - NOC računa broj neposrednih podklasa posmatrane klase/interfejsa u hijerarhiji klasa
  - NOC vrednost se dobija brojanjem neposrednih podklasa posmatrane klase
  - Ukoliko je NOC veći, povećava se mogućnost ponovnog korišćenja. Takođe, sa rastom NOC povećava se i verovatnoća nedogovarajuće apstrakcije

Softverske metrike

- Ciklična složenost (CC)
  - Predstavlja meru složenosti primenjenog algoritma
  - Meri se računanjem broja tačaka odlučivanja ili uslovnih iskaza u datom programskom jeziku
  - if, else, for, while, do-while, catch, default
  - &, |, &&, ||, ?:, !
  - CC=P+1, P – broj predikata ili broj uslova ili broj binarnih čvorova, 1 – ulazna putanja f-je
  - CC>=1

Softverske metrike

- Halstedova složenost (HE)
  - Merenje složenosti modula programa korišćenjem operatora i operanda. Indikator složenosti programa.
  - HE=V*D ili HE=V/L, n1 – broj različitih operatora, n2 – broj različitih operanada, N1 – ukupan broj operatora, N2 – ukupan broj operanada
  - N=N1+N2 (Halstedova dužina programa)
  - n=n1+n2 (Veličina rečnika)
  - V=n*log2(n) (Obim programa)
  - D=(n1/2)*(N2/n2) (Nivo težine)
  - L=1/D ili L=(2*n2)/(n1*N2) (Nivo programa)
  - Ako je vrednost veća, veći je napor potreban za održavanje

Softverske metrike

- Indeks održavanja (MI)
  - Predstavlja kvantitativnu meru namenjenu merenju i praćenju održavanja
  - MI=171-5.2*log2(aveV)-0.23*aveV(g')-16.2*log2(aveLOC)+50*sin(sqrt(2.46*perCM))
  - AveV – prosečan Halstedov obim programa po modulu, AveV(g) – Prosečna ciklična složenost po modulu, aveLOC – Prosečan broj linija koda po modulu, perCM – prosečan procenat linija komentara po modulu
  - MI<65 – Mogućnosti održavanja softvera male
  - 65≤MI<85 – Mogućnosti održavanja softvera dobre
  - MI≥85 – Mogućnosti održavanja softvera odlične

Zaključak

- Standardi kvaliteta softvera na sveobuhvatan način posmatraju kvalitet softverskog sistema i procesa razvoja softvera
- Softverske metrike i atributi kvaliteta softvera
- Alati za statičku analizu kvaliteta softvera se operativno koriste za analizu kvaliteta softvera

Softverske metrike

Univerzitet u Beogradu
Fakultet organizacionih nauka
Dr Miloš Milić