

Poglavlje 24

Menadžment kvaliteta

DARKO STOŠIĆ 1059/2014

Menadžment kvaliteta

- ▶ Menadžment kvaliteta predstavlja menadžment softverskog procesa i proizvoda

Ciljevi

- ▶ Upoznavanje sa procesom menadžmenta kvaliteta i glavnim aktivnostima vezanih za tu temu
- ▶ Objašnjenje uloge standarda u menadžmentu kvaliteta
- ▶ Objašnjenje koncepta metrike softvera, metrike predviđanja i metrike kontrole
- ▶ Objašnjenje koje mere mogu biti preduzete pri određivanju kvaliteta softvera

Pokrivenene teme

- ▶ Garantovanost kvaliteta i standardi
- ▶ Kvalitetno planiranje
- ▶ Kvalitetna kontrola
- ▶ Mere i metrike softvera

Menadžment kvaliteta softvera

- ▶ Zabrinutost za obezbeđivanjem potrebnog nivoa kvaliteta softvera
- ▶ Uključuje definisanje prikladnih standarda kvaliteta i procedura, takođe garantujući da će biti poštovane
- ▶ Treba uvek ciljati ka razvitku “kulture kvaliteta”, gde je kvalitet viđen kao svačija odgovornost

Šta je kvalitet?

- ▶ Kvalitet, laički, znači da proizvod treba da zadovolji svoju specifikaciju
- ▶ To predstavlja problem za softverske sisteme
 - ▶ Tenzije između zahteva za kvalitetom mušterije(efikasnost, pouzdanost, itd...) i zahteva za kvalitetom razvijaoca(održavanje, upotrebljivost, itd...)
 - ▶ Neki zahtevi kvaliteta su teški za jednosmislenu specifikaciju
 - ▶ Softverske specifikacije su često nekompletne i nesaglasne

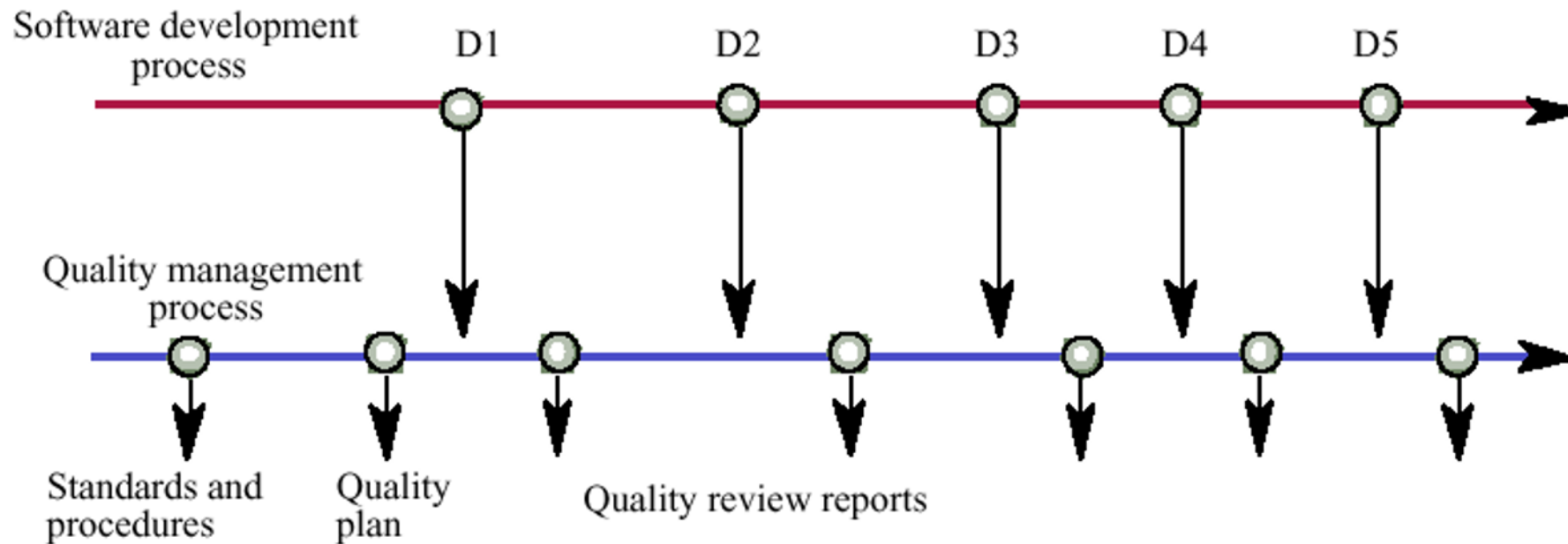
Kompromis kvaliteta

- ▶ Ne može se čekati da se specifikacije poboljšaju pre nego što obratimo pažnju na menadžment kvaliteta
- ▶ Moraju se postaviti procedure koje poboljšavaju kvalitet bez obzira na nesavršenost specifikacije
- ▶ Što će reći, menadžment kvaliteta nije samo skoncentrisan na eliminisanje defekata već obraća pažnju i na druge kvalitete proizvoda

Aktivnosti menadžmenta kvaliteta

- ▶ Garantovanost kvaliteta
 - ▶ Uspostavljanje organizacionih procedura i standarda kvaliteta
- ▶ Planiranje kvaliteta
 - ▶ Odabir primenljivih procedura i standarda za specifičan projekat i njihova modifikacija po potrebi
- ▶ Kontrola kvaliteta
 - ▶ Obezbediti da su procedure i standardi praćeni od strane razvojnog tima
- ▶ Menadžment kvaliteta treba biti odvojen od menadžmenta projekta da bi se zadržala njegova nezavisnost

Menadžment kvaliteta i razvoj softvera



ISO 9000

- ▶ Internacionalni skup standarda za menadžment kvaliteta
- ▶ Primenljiv na širok diapazon organizacija, od proizvodnje do servisne industrije
- ▶ ISO 9001 je primenljiv na organizacije koje dizajniraju, razvijaju i održavaju proizvode
- ▶ ISO 9001 je generički model procesa kvaliteta, primena se razlikuje od organizacije do organizacije

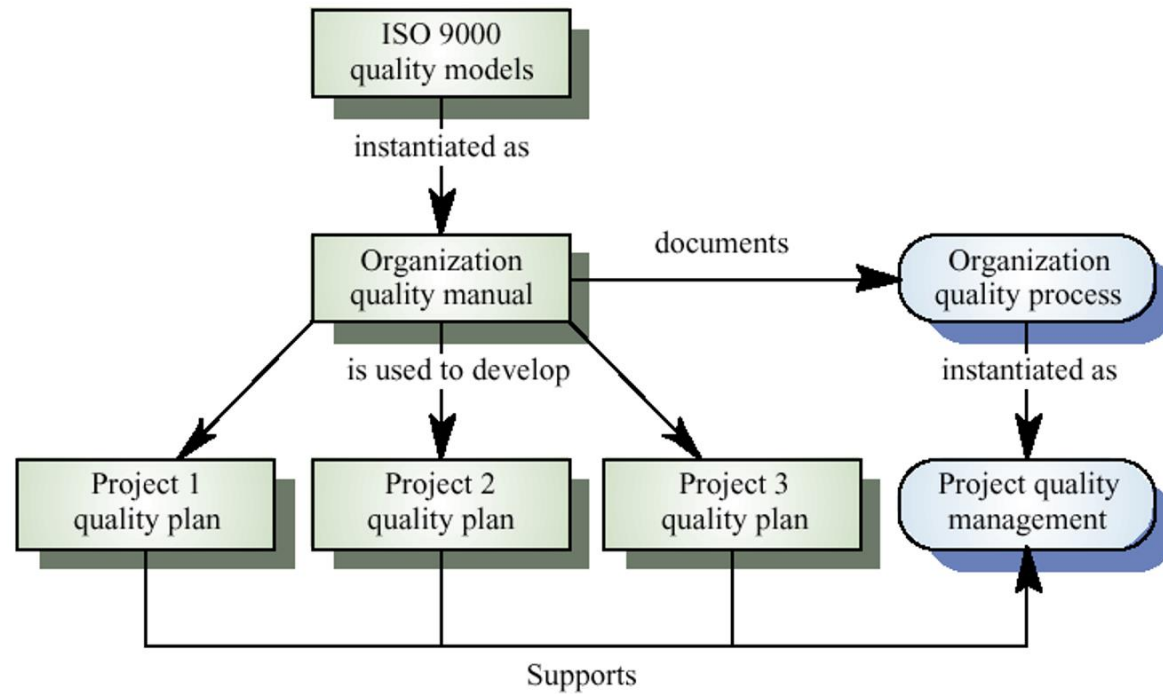
ISO 9001

Management responsibility	Quality system
Control of non-conforming products	Design control
Handling, storage, packaging and delivery	Purchasing
Purchaser-supplied products	Product identification and traceability
Process control	Inspection and testing
Inspection and test equipment	Inspection and test status
Contract review	Corrective action
Document control	Quality records
Internal quality audits	Training
Servicing	Statistical techniques

ISO 9000 potvrda

- ▶ Standardi kvaliteta i procedure trebaju biti dokumentovane u organizacionom uputstvu za kvalitet
- ▶ Spoljašnje telo može potvrditi da je organizaciono uputstvo za kvalitet prilagođeno ISO 9000 standardima
- ▶ Mušterije, sve više i više, traže da su dobavljači ISO 9000 odobreni

ISO 9000 i menadžment kvaliteta



Garantovanost kvaliteta i standardi

- ▶ Standardi su ključ ka efektivnom menadžmentu kvaliteta
- ▶ Mogu biti internacionalni, nacionalni, organizacioni ili projektni standardni
- ▶ Standardi proizvoda definišu karakteristike koje sve komponente trebaju da prikazuju npr. opšti stil programiranja
- ▶ Standardi procesa definišu kako proces softvera treba biti usvojen

Značaj standarda

- ▶ Enkapsulacija najboljih praksi – izbegavanje grešaka iz prošlosti
- ▶ Okvir za proces garantovanja kvaliteta – uključuje proveru saglasnosti standarda
- ▶ Obezbediti kontinuitet – novi članovi mogu da razumeju organizaciju preko razumevanja primenjenih standarda

Standardi proizvoda i procesa

Product standards	Process standards
Design review form	Design review conduct
Document naming standards	Submission of documents to CM
Procedure header format	Version release process
Ada programming style standard	Project plan approval process
Project plan format	Change control process
Change request form	Test recording process

Problemi sa standardima

- ▶ Nisu viđeni kao relevantni i ažurni od strane softverskih inženjera
- ▶ Uključuju previše birokratskog popunjavanja formi
- ▶ Nisu podržani od strane softverskih alata i uključuju monoton ručni rad oko održavanja standarda

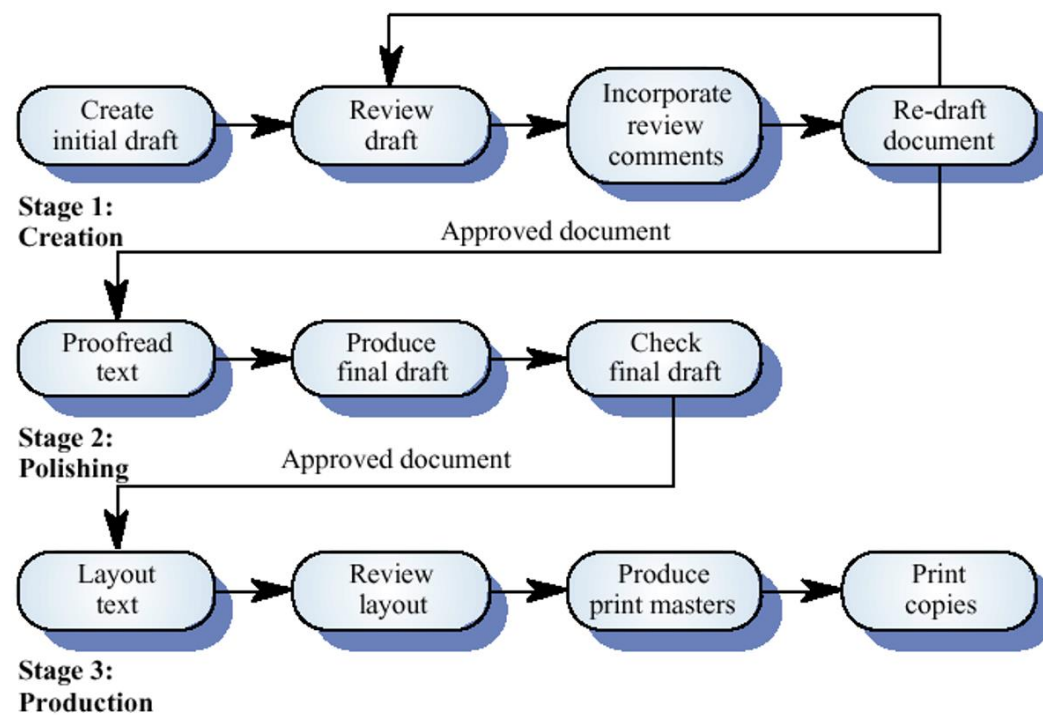
Razvijanje standarda

- ▶ Uključiti ljude koji ih praktikuju u razvoj. Inženjeri treba da razumeju ideju iza standarda
- ▶ Recenzije standarda i njihova česta upotreba. Standardi brzo mogu postati zastareli i time gube kredibilitet kod praktikanata
- ▶ Detaljni standardi trebaju imati softversku pratnju. Prekomeran papirološki rad je glavna žalba protiv standarda

Dokumentacioni standardi

- ▶ Naročito važni – dokumenti su vidljiva manifestacija softvera
- ▶ Standardi procesa dokumentacije
 - ▶ Kako dokument treba biti razvijan, validiran i održavan
- ▶ Standardi dokumenta
 - ▶ Zabrinutost za sadržaj dokumenta, strukturu i izgled
- ▶ Standardi razmene dokumenata
 - ▶ Kako se dokumenti čuvaju i razmenjuju između različitih sistema za dokumentaciju

Proces dokumentacije



Standardi dokumenta

- ▶ Standardi identifikacije dokumenta
 - ▶ Kako se dokumenti jedinstveno identifikuju
- ▶ Standardi structure dokumenta
 - ▶ Standardna struktura za projektne dokumente
- ▶ Standardi prezentacije dokumenta
 - ▶ Definisanje fontova, stilova, korišćenje logoa, itd...
- ▶ Standardi ažuriranja dokumenta
 - ▶ Definisanje kako su promene iz prethodnih verzija reflektovane u dokumentu

Standardi razmene dokumenata

- ▶ Dokumenti se proizvode koristeći različite sisteme i na različitim računarima
- ▶ Standardi razmene omogućavaju da se elektronski dokumenti razmenjuju, šalju mejlom, itd...
- ▶ Potreba za arhiviranjem. Životni vek sistema za obradu teksta može biti mnogo kraći od životnog veka softvera koji se dokumentuje
- ▶ XML je jedan od standarda koji se koristi za razmenu dokumenata

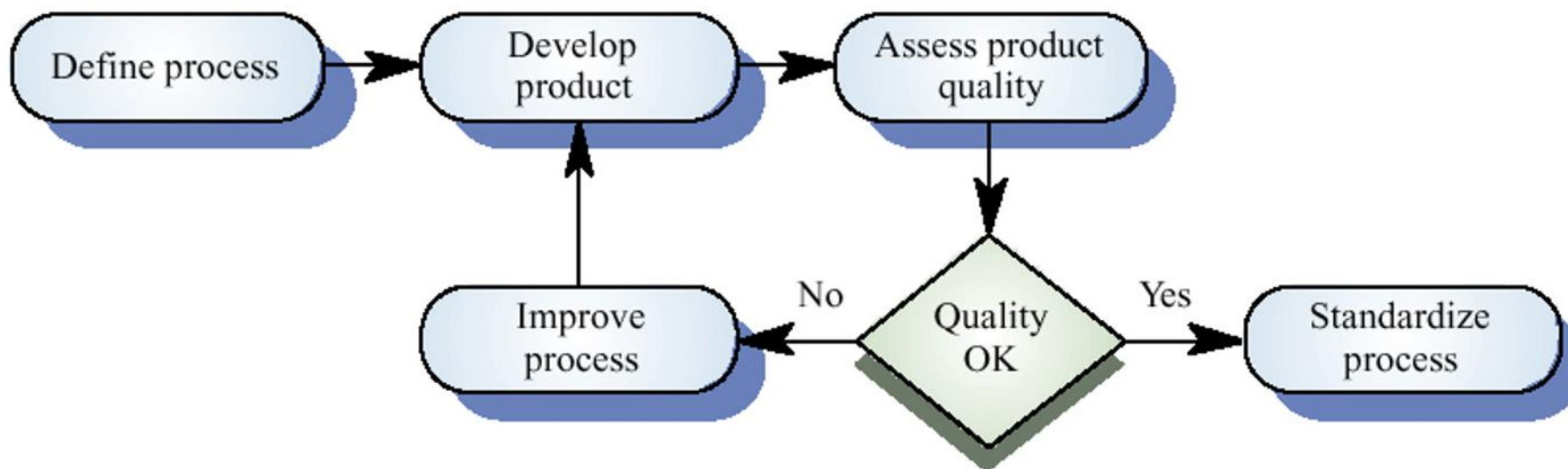
Proces i kvalitet proizvoda

- ▶ Na kvalitet razvijenog proizvoda utiče kvalitet produkcionog procesa
- ▶ Posebno je bitan u razvoju softvera, jer su neki atributi kvaliteta proizvoda teški za procenu
- ▶ Ipak, postoji veoma kompleksna i loše razumljiva veza između softverskog procesa i kvaliteta proizvoda

Kvalitet zasnovan na procesu

- ▶ Jednostavna veza između procesa i proizvoda u robi
- ▶ Kompleksnija za softver:
 - ▶ Aplikacija individualnih veština i iskustva je vrlo bitna u razvoju softvera
 - ▶ Spoljašnji faktori kao što su “novina nove” aplikacije ili potreba za ubrzanim razvojem može loše uticati na kvalitet proizvoda
- ▶ Treba voditi računa da se ne postave neadekvatni standardi procesa

Kvalitet zasnovan na procesu



Kvalitet zasnovan na procesu u praksi

- ▶ Definirati standarde procesa poput načina vršenja recenzija, konfiguracionog menadžmenta, itd...
- ▶ Posmatranje procesa razvoja da bi se obezbedilo primenjivanje i praćenje standarda
- ▶ Izveštavati menadžment zadužen za projekt i softverskog dobavljača o procesu

Kvalitetno planiranje

- ▶ Kvalitetan plan postavlja željene kvalitete proizvoda unapred i njihov način ocenjivanja. Takođe definiše najbitnije attribute kvaliteta
- ▶ Treba da definiše proces procenjivanja kvaliteta
- ▶ Treba da postavi organizacione standarde koji će se primenjivati i, ako je potrebno, da definiše nove

Struktura kvalitetnog plana

- ▶ Opis proizvoda
- ▶ Planovi za proizvod
- ▶ Opis procesa
- ▶ Ciljevi za kvalitet
- ▶ Rizici i njihov menadžment
- ▶ Kvalitetni planovi trebaju biti kratki, sažeti dokumenti
 - ▶ Ako su dugački, niko ih neće pročitati

Atributi softverskog kvaliteta

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

Kontrola kvaliteta

- ▶ Proveravanje procesa razvoja softvera da bi se obezbedilo da su procedure i standardi praćeni
- ▶ Dva pristupa kontroli kvaliteta
 - ▶ Pregled kvaliteta
 - ▶ Automatizovano procenjivanje softvera i njegovih mera

Kvalitetni pregledi

- ▶ Predstavljaju glavni metod validacije kvaliteta procesa ili proizvoda
- ▶ Grupni pregled dela procesa (ili celog) ili sistema i njegove dokumentacije radi pronalaženja potencijalnih problema
- ▶ Postoji različite vrste pregleda sa različitim ciljevima
 - ▶ Inspekcije za otklanjanje defekata (proizvod)
 - ▶ Pregledi za procenu napretka (proizvod i proces)
 - ▶ Kvalitetni pregledi (proizvodi i standardi)

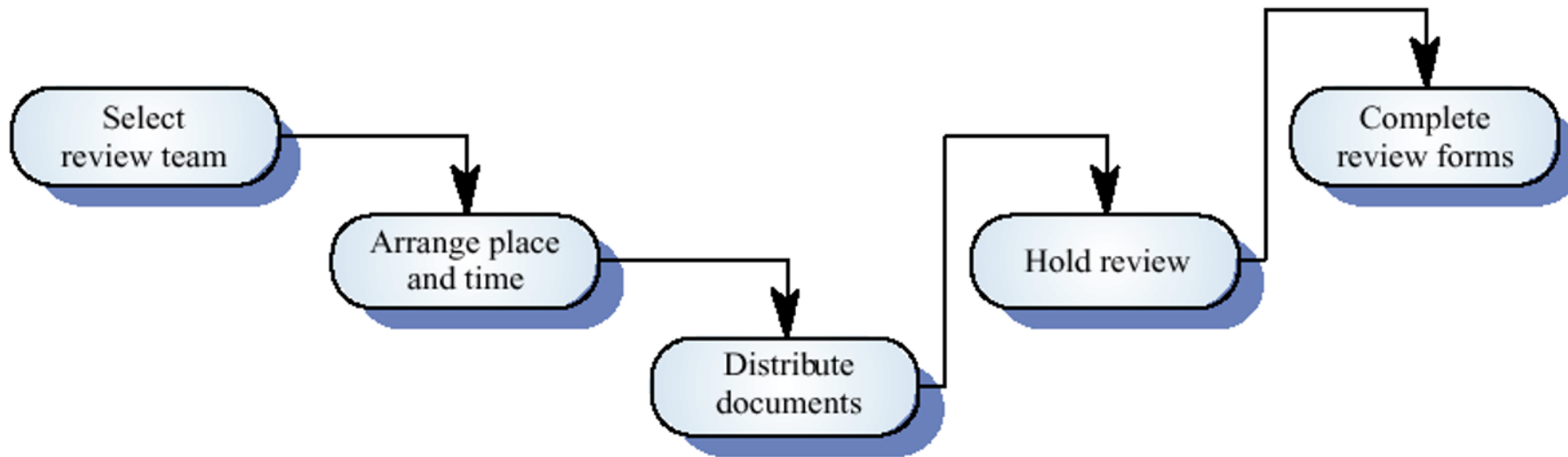
Vrste pregleda

Review type	Principal purpose
Design or program inspections	To detect detailed errors in the design or code and to check whether standards have been followed. The review should be driven by a checklist of possible errors.
Progress reviews	To provide information for management about the overall progress of the project. This is both a process and a product review and is concerned with costs, plans and schedules.
Quality reviews	To carry out a technical analysis of product components or documentation to find faults or mismatches between the specification and the design, code or documentation. It may also be concerned with broader quality issues such as adherence to standards and other quality attributes.

Kvalitetni pregledi

- ▶ Grupa ljudi pažljivo istražuju deo ili ceo softveski sistem i vezanu dokumentaciju
- ▶ Kod, dizajn, specifikacije, planovi testiranja, standardi, itd... sve se pregleda
- ▶ Softver ili dokumenti mogu biti “otpisani” pri pregledu što označava da je napredak ka sledećoj fazi odobren od menadžmenta

Proces pregleda



Funkcije pregleda

- ▶ Funkcija kvaliteta – deo generalnog procesa upravljanja kvalitetom
- ▶ Funkcija upravljanja projektom – obezbeđuju informacije za projektne menadžere
- ▶ Funkcija treninga i komunikacije – znanje o proizvodu se prenosi između članova razvojnog tima

Kvalitetni pregledi

- ▶ Cilj je pronalaženje sistemskih defekata i nedoslednosti
- ▶ Svaki dokument stvoren u procesu može biti pregledan
- ▶ Timovi za pregled trebaju biti relativno mali i sami pregledi relativno kratki
- ▶ Pregledi se trebaju evidentirati. Evidencije se trebaju održavati

Rezultati pregleda

- ▶ Komentari načinjeni tokom pregleda se trebaju klasifikovati
 - ▶ Bez akcije. Nije potrebno menjati softver ili dokumentaciju
 - ▶ Potreba za popravkom. Dizajner ili programer je zadužen za ispravku identifikovane greške
 - ▶ Preispitati celokupni dizajn. Problem identifikovan u pregledu utiče na ostale delove dizajna. Treba doneti presudu o najboljem rešenju koje uzima u obzir odnos cena-kvalitet
- ▶ Greške u zahtevima i specifikaciji možda zahtevaju učešće klijenta

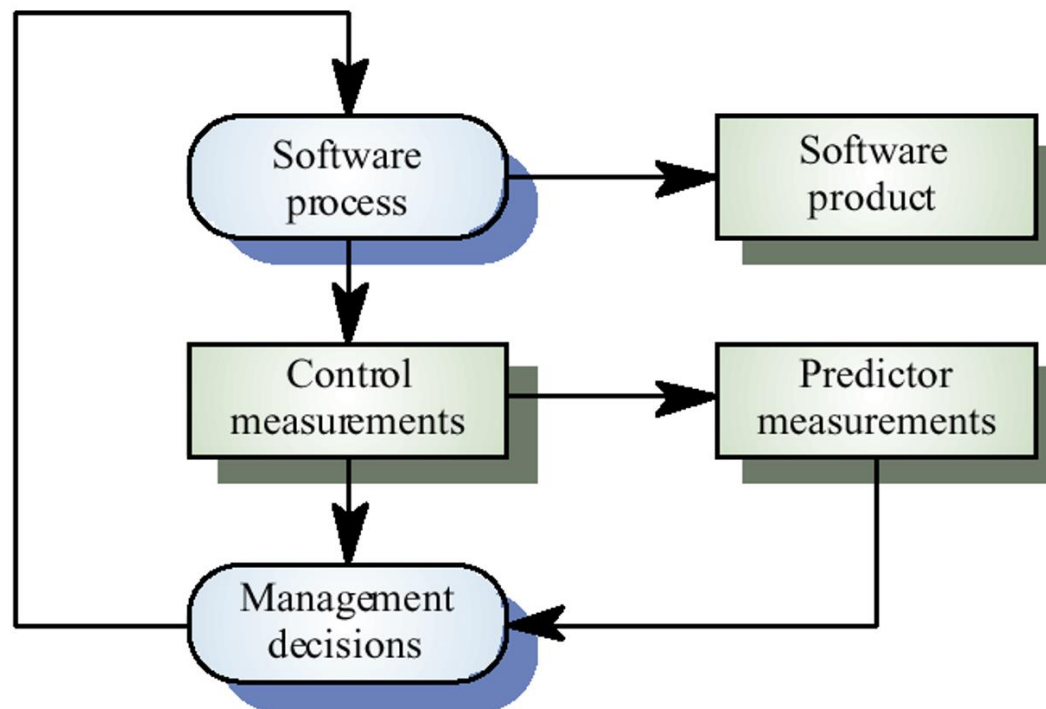
Mere i metrike softvera

- ▶ Mera softvera je skoncentrisana na izvođenje numeričke vrednosti za atribut softverskog proizvoda ili procesa
- ▶ Ovo dozvoljava za objektivna poređenja između tehnika i procesa
- ▶ Iako su neke kompanije uvele programe merenja, sistematska upotreba mera i dalje nije česta
- ▶ Postoji vrlo mala standarda u ovoj oblasti

Metrika softvera

- ▶ Bilo koja vrsta mere koja je vezana za softverski sistem, proces ili povezane dokumentacije
- ▶ Dozvoljava da se softver i softverski proces kvantifikuju
- ▶ Mere softverskog procesa ili proizvoda
- ▶ Može se koristiti da predvidi attribute proizvoda ili da kontroliše softverski proces

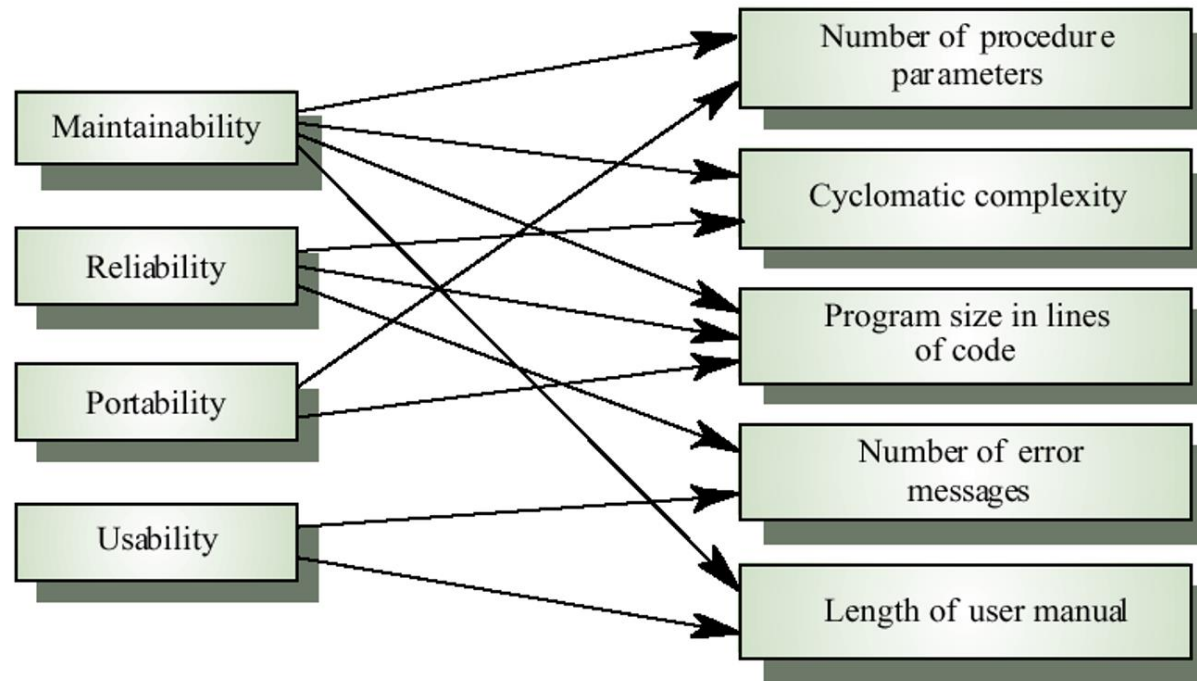
Predskazivač i metrike kontrole



Pretpostavke metrike

- ▶ Softverska imovina se može izmeriti
- ▶ Postoji veza između onog što možemo da merimo i onog što želimo da znamo
- ▶ Ova veza je formalizovana i validirana
- ▶ Može biti teško vezati sve što se može meriti sa željenim atributima kvaliteta

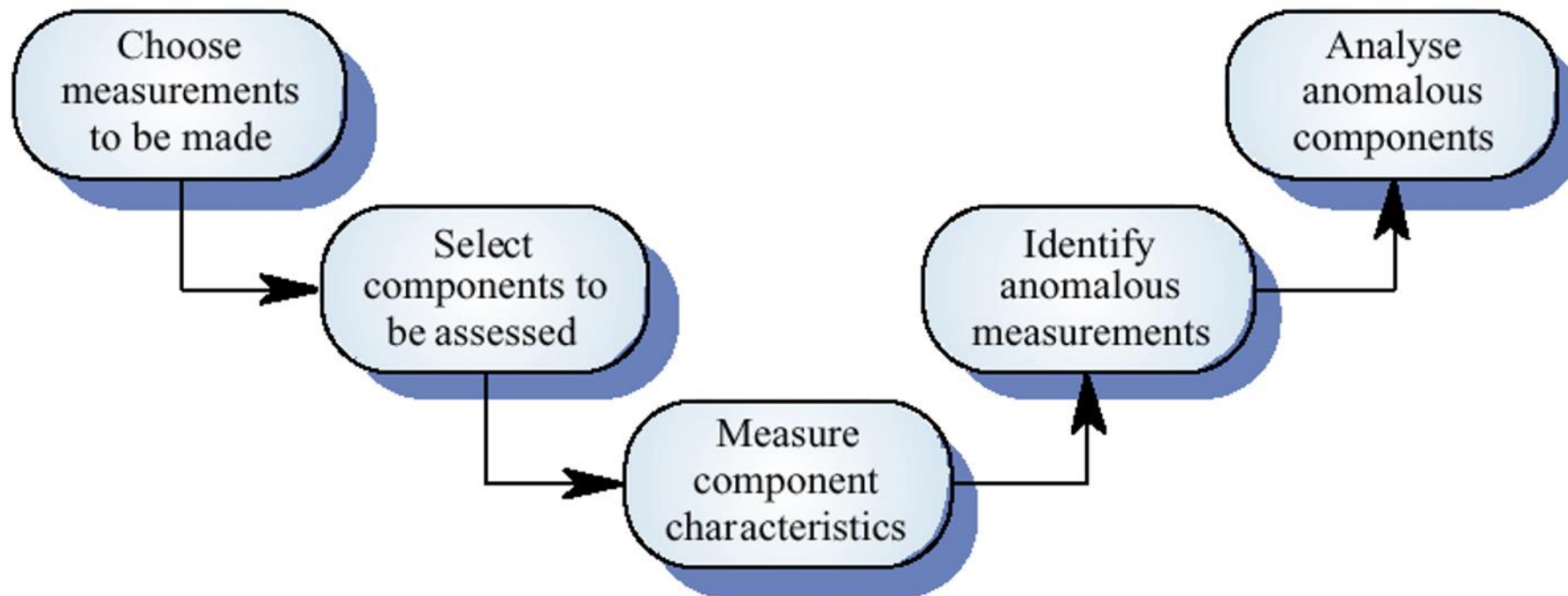
Unutrašnji i spoljašnji atributi



Proces merenja

- ▶ Proces merenja softvera može biti deo procesa kontrole kvaliteta
- ▶ Podaci sakupljeni tokom ovog procesa trebaju se održavati kao organizacioni resurs
- ▶ Kada se baza merenja uspostavi, poređenja između projekata postaju moguća

Proces merenja proizvoda

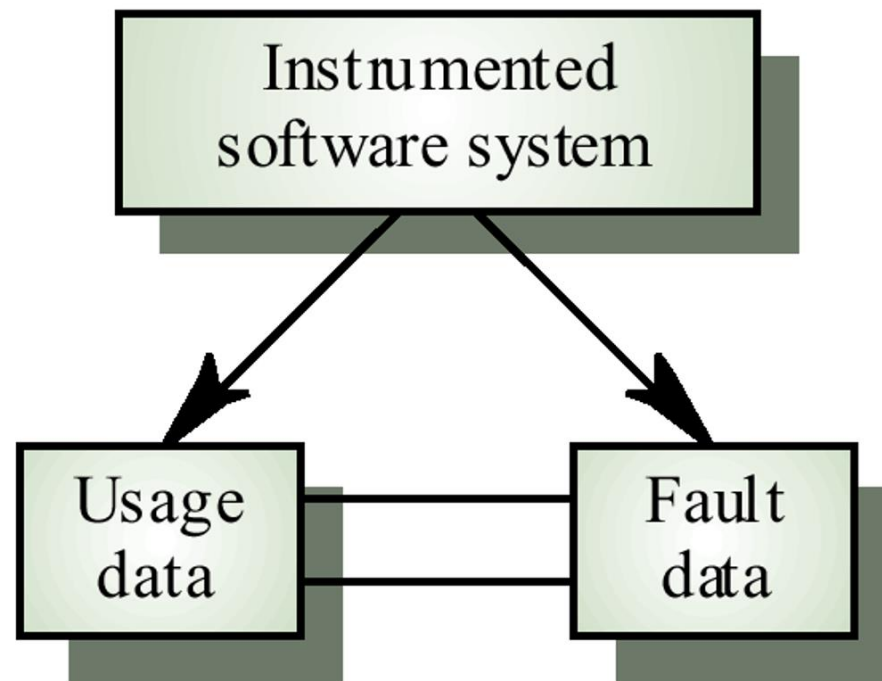


Skupljanje podataka

- ▶ Program zadužen za metriku treba biti zasnovan na skupu podataka od proizvoda i procesa
- ▶ Podaci se trebaju odmah skupljati i ako je moguće automatski
- ▶ Tri tipa automatskog prikupljanja podataka
 - ▶ Statička analiza proizvoda
 - ▶ Dinamička analiza proizvoda
 - ▶ Proces svrstavanja podataka

Automatizovano prikupljanje podataka

45



Preciznost podataka

- ▶ Ne skupljati nepotrebne podatke
 - ▶ Pitanja na koja se traži odgovor se trebaju odrediti unapred i potrebni podaci se identifikovati
- ▶ Reći ljudima zašto se podaci sakupljaju
 - ▶ Ne treba da bude deo evaluacije osoblja
- ▶ Ne zavisiti od memorije
 - ▶ Skupljati podatke kada su generisani ne kad je projekat završen

Metrike proizvoda

- ▶ Metrika kvaliteta trebala bi da bude predskazivač za kvalitet proizvoda
- ▶ Klase metrike proizvoda
 - ▶ Dinamičke metrike koje se prikupljaju preko mera koje je napravio program pri izvršavanju
 - ▶ Statičke metrike koje se prikupljaju preko mera napravljenih od reprezentacija sistema
 - ▶ Dinamičke metrike pomažu u proceni efikasnosti i pouzdanosti; statičke metrike pomažu u procenjivanju kompleksnosti, razumljivosti i održivosti

Dinamičke i statičke metrike

- ▶ Dinamičke metrike su usko povezane sa kvalitetima atributa softvera
 - ▶ Relativno je lako izmeriti vreme odziva sistema(atribut performansi) ili broj neuspeha(atribut pouzdanosti)
- ▶ Statičke metrike imaju indirektan odnos sa atributima kvaliteta
 - ▶ Treba pokušati sa izvođenjem odnosa između ovih metrika i svojstava poput kompleksnosti, razumljivosti i održivosti

Metrike softverskog proizvoda

Software metric	Description
Fan in/Fan-out	Fan-in is a measure of the number of functions that call some other function (say X). Fan-out is the number of functions which are called by function X. A high value for fan-in means that X is tightly coupled to the rest of the design and changes to X will have extensive knock-on effects. A high value for fan-out suggests that the overall complexity of X may be high because of the complexity of the control logic needed to coordinate the called components.
Length of code	This is a measure of the size of a program. Generally, the larger the size of the code of a program's components, the more complex and error-prone that component is likely to be.
Cyclomatic complexity	This is a measure of the control complexity of a program. This control complexity may be related to program understandability. The computation of cyclomatic complexity is covered in Chapter 20.
Length of identifiers	This is a measure of the average length of distinct identifiers in a program. The longer the identifiers, the more likely they are to be meaningful and hence the more understandable the program.
Depth of conditional nesting	This is a measure of the depth of nesting of if-statements in a program. Deeply nested if statements are hard to understand and are potentially error-prone.
Fog index	This is a measure of the average length of words and sentences in documents. The higher the value for the Fog index, the more difficult the document may be to understand.

Objektno orjentisane metrike

Object-oriented metric	Description
Depth of inheritance tree	This represents the number of discrete levels in the inheritance tree where sub-classes inherit attributes and operations (methods) from super-classes. The deeper the inheritance tree, the more complex the design as, potentially, many different object classes have to be understood to understand the object classes at the leaves of the tree.
Method fan-in/fan-out	This is directly related to fan-in and fan-out as described above and means essentially the same thing. However, it may be appropriate to make a distinction between calls from other methods within the object and calls from external methods.
Weighted methods per class	This is the number of methods included in a class weighted by the complexity of each method. Therefore, a simple method may have a complexity of 1 and a large and complex method a much higher value. The larger the value for this metric, the more complex the object class. Complex objects are more likely to be more difficult to understand. They may not be logically cohesive so cannot be reused effectively as super-classes in an inheritance tree.
Number of overriding operations	These are the number of operations in a super-class which are over-ridden in a sub-class. A high value for this metric indicates that the super-class used may not be an appropriate parent for the sub-class.

Analiza mera

- ▶ Nije uvek jasno šta podaci označavaju
 - ▶ Analiza prikupljenih podataka je vrlo teška
- ▶ Profesionalni statističari se trebaju konsultovati ako su dostupni
- ▶ Analiza podataka treba uzeti u obzir lokalne okolnosti

Iznenadjenja mera

- ▶ Smanjenje broja grešaka u programu dovodi do povećanog broja poziva tehničkoj podršci
 - ▶ Program se sada smatra pouzdanijim i ima šire i raznoliko tržište. Procenat korisnika koji zovu tehničku podršku se možda smanjuje ali ukupan broj poziva se povećava
 - ▶ Pouzdaniji sistem se koristi na drugačiji način od sistema gde korisnici moraju sami da prevazilaze greške. Ovo dovodi do više poziva tehničkoj podršci

Ključne tačke

- ▶ Upravljanje kvalitetom softvera je zaduženo da obezbedi da softver ispunjava tražene standarde
- ▶ Procedure garancije kvaliteta trebaju biti dokumentovane u organizacionom uputstvu kvaliteta
- ▶ Standardi softvera su enkapsulacija najboljih praksi
- ▶ Pregledi su najrasprotranjeniji pristup za procenu kvaliteta softvera

Ključne tačke

- ▶ Mere softvera prikupljaju informacije o softverskom procesu i softverskom proizvodu
- ▶ Metrike kvaliteta proizvoda trebaju se koristiti da identifikuju potencijalne problematične komponente
- ▶ Ne postoje standardizovane i universalno primenljive metrike softvera