

INFO-1072

Primljeno/Received: 2009-05-16

UDK: 167/168:001:130.02:007

Prethodno priopćenje / Preliminary Communication

## SUSTAVI ZNANJA U POTPORI UPRAVLJANJU KVALITETOM SOFTVERSKOG PROIZVODA

### KNOWLEDGE-BASED SYSTEM SUPPORTING SOFTWARE QUALITY MANAGEMENT

*Vanja Bevanda , Giorgio Sinković*

Sveučilište Juraj Dobrile u Puli, Odjel ekonomije i turizma, Pula, Hrvatska  
 Juraj Dobrila University of Pula, Department of Economics and Tourism, Pula, Croatia

#### *Sažetak*

Softverski proizvod rijetko postiže očekivanu kvalitetu i samo jedan od pet projekata razvoja uspješno zadovoljava kriterije kvalitete. Osnovni kriterij za njenu procjenu, kao i kod ostalih proizvoda, je zadovoljstvo kupca. Zbog specifičnosti softverskog proizvoda, jedan od problema upravljanja kvalitetom je činjenica da ni kupac ni proizvođač nemaju točnu predodžbu konačnog proizvoda niti njegove kvalitete, a ukoliko ju i imaju, podložna je promjenama tijekom životnog ciklusa razvoja softverskog proizvoda. Kupac mijenja svoje zahtjeve tijekom razvoja softvera i s time i temeljne kriterije za ocjenu kvalitete. Relativnu važnost značajki kvalitete softverskog proizvoda nije moguće generalizirati i ona ovisi kako o proizvodu tako i o konkretnom kupcu. Može se reći da je postizanje nižih troškova osiguranja kvalitete u odnosu na postignuti efekt smanjenja troškova ne kvalitete i propuštenih prilika na tržištu, jedan od temeljnih ciljeva upravljanja kvalitetom. U radu je istražena mogućnost primjene sustava znanja u potpori procesu identifikacije važnih značajki kvalitete softverskog proizvoda u određenom trenutku u cilju racionalnog upravljanja kvalitetom. Primijenjen je modificiran C4.5 algoritam i inducirani su atributi s najvišom informacijskom vrijednošću. Dobivena pravila opisuju različitu percepciju relativne važnosti značajki kvalitete danog softverskog proizvoda proizvođača i korisnika, te time "signaliziraju" potrebu poduzimanja aktivnosti upravljanja kvalitetom softverskog proizvoda.

#### *Abstract*

The quality of software products rarely achieves the level expected by managers purchasing the software. Only one of five implemented software projects successfully fulfills the required quality criteria. As with other products, customer satisfaction is the basic quality criterion for software products. Because each software product must be designed for an individual user, neither the customer nor the software producer has an exact image of the final product or its quality. Even when the customer and producer have similar images, the product is subject to change during the software development lifecycle. Customers often modify software requirements as the underlying needs of the customer change. It is not possible to generalize the relative importance of software quality characteristics because they are unequal for different products as well as for different customers. The aim of this research is to explore the possibilities of knowledge-based systems to support and automate the process of identifying important software quality characteristics at a specific moment in time. Identifying quality characteristics will facilitate software quality management. We applied a modified C4.5 algorithm and induced the most informative attributes with rules describing a variety of perceptions of software quality characteristics from customer and producer perspectives. The findings provide managers with useful insights that will signal needs for software quality improvements, which will enhance and refine their business performance.

#### **1. Uvod**

U definiranju pojma kvalitete ključna je uloga kupca proizvoda, odnosno onog tko je na bilo koji način zainteresiran za proizvod. Na njegovo zadovoljstvo utječu dvije skupine značajki:

1. Svojstva proizvoda: koja je kupac specificirao ili ih je podrazumijevao, kao na primjer želi

automobil koji je pouzdan, prostran, udoban, klimatiziran, itd.

2. Oslobodjenost od nedostataka odnosno da proizvod nema greške kao na primjer da se automobil ne gasi u tijeku vožnje, da nema slabe kočnice i slično.

Kvalitetan proizvod zadovoljava svojstva obje navedene skupine. Kod većine proizvoda svojstva

se definiraju u fazi oblikovanja i razvoja (dizajna), dok su nedostaci najčešće rezultat propusta u proizvodnji. Nije rijedak slučaj da proizvođač proizvede određenu količinu identičnih proizvoda jednake kvalitete i nakon što ih isporuči različitim kupcima utvrdi različiti stupanj zadovoljstva isporučenim proizvodom. Iako je riječ o istim proizvodima, s jednakim svojstvima i jednake oslobođenosti od nedostataka, očekivanja kupaca su bila različita, a time i različit stupanj zadovoljstva.

Kvaliteta proizvoda koja je rezultat percepcije kupca može se opisati pojmom vanjske kvalitete proizvoda, odnosno, unutarnja kvaliteta proizvoda je predodžba kvalitete sa stajališta proizvođača. Pretpostavka je da se određeni stupanj vanjske kvalitete ne može ostvariti bez odgovarajuće razine unutarnje kvalitete. Kvaliteta softverskog proizvoda se može definirati na jednak način kao i za ostale proizvode uz definiranje dodatnih svojstava i načina njihovog mjerenja. Dosadašnja iskustva proizvodnje softvera pokazuju vrlo visok postotak loše kvalitete softvera. Samo 20 % projekata rezultira potpunim zadovoljstvom kupca, a gotovo isto toliko projekata se napusti u fazi razvoja ili neposredno nakon primjene. Radi se o ogromnim izgubljenim sredstvima /1/. Upravljanje kvalitetom trebalo bi biti usmjereno prema cilju da troškovi osiguranja kvalitete budu manji od pozitivnog efekta koji se sastoji u smanjenju troškova vezanih za ne kvalitetu ( u koje treba ubrojiti i propuštene prilike na tržištu). Zbog specifičnosti softverskog proizvoda kao problem upravljanja kvalitetom se javlja činjenica da ni kupac ni proizvođač nemaju točnu predodžbu konačnog proizvoda niti njegove kvalitete, a ukoliko ju i imaju, podložna je promjenama tijekom životnog ciklusa razvoja softverskog proizvoda. Kupac mijenja svoje zahtjeve tijekom razvoja softverskog proizvoda i s time i temeljne kriterije za ocjenu kvalitete. Relativnu važnost značajki kvalitete softverskog proizvoda nije moguće generalizirati i ona ovisi kako o proizvodu tako i o konkretnom kupcu. Proizvođač, ukoliko želi racionalno upravljati kvalitetom, ima potrebu za identifikacijom i analizom relativne važnosti značajki kvalitete proizvoda kako ju kupac percipira, kako bi prilagođavao svoje procese i svojstva unutarnje kvalitete njihovom zadovoljenju. Kako se zahtjevi kupca mijenjaju tijekom životnog ciklusa softvera, a različito se percipiraju sa stajališta zainteresiranih strana, poželjno je kreiranje snimki relativne važnosti značajki kvalitete softverskog proizvoda u određenom trenutku. U radu je istražena mogućnost automatizacije i potpore tom procesu primjenom sustava znanja. Anketiranjem razvojnog tima jednog hrvatskog proizvođača softvera, kao i određenog broja njihovih korisnika prikupljene su

dvije zasebne skupine slučajeva koji su služile kao ulaz za induktivno zaključivanje. Primijenjen je modificiran C4.5 algoritam i inducirani atributi s najvišom informacijskom vrijednošću. Na taj način su dobivena pravila koja opisuju različitu percepciju važnosti značajki kvalitete danog softverskog proizvoda proizvođača i korisnika.

## 2. Specifičnosti utvrđivanja kvalitete softverskog proizvoda

Softverski se proizvodi po mnogim elementima razlikuju od hardverskih jer njihova neopipljivost ograničava mogućnosti ocjenjivanja nekih svojstava. Proces proizvodnje hardverskih proizvoda može se podijeliti u dvije po mnogo čemu ravnopravne faze: dizajn proizvoda i proizvodnja. Kod softverskog proizvoda nakon završetkom faze razvoja ne postoji proizvodnja u klasičnom smislu. Naime, ako se softver razvija za jednog korisnika onda ni nema proizvodnje, a ako se radi za više korisnika on se jednostavno replicira. Replikacija softvera se može smatrati proizvodnjom, ali su mogućnosti da se svojstva pojedinih repliciranih kopija softvera razlikuju, više teorijske nego praktičke prirode. Dakle, ako se govori o trenutku kada je proizvod isporučen i implementiran kod korisnika, tj. o unutarnjoj kvaliteti softvera, ona nastaje u fazi razvoja bez obzira radi li se o njegovim svojstvima ili oslobođenosti od nedostataka. Vanjska kvaliteta softvera, kao i kod drugih proizvoda, ovisi o zadovoljstvu kupca /2/. Zbog toga se pri upravljanju kvalitetom softvera velika pozornost posvećuje procesu razvoja. U SAD je razvijena posebna metoda mjerenja zrelosti tog procesa nazvana SEI - Capability Maturity Model skraćeno SEI-CMM /3/, a i međunarodni standardi grupe ISO 9000 su orijentirani na proces proizvodnje, odnosno kada je softver u pitanju, na proces razvoja. Uporabom softverskog proizvoda kupac u većoj ili manjoj mjeri ostvaruje zacrtane ciljeve o čemu ovisi njegovo zadovoljstvo i to se naziva djelotvornost softverskog proizvoda. Pri ostvarenju tih ciljeva dolaze na vidjelo još neka svojstva proizvoda kao učinkovitost, tj. koliko se resursa troši za ispunjavanje zahtjeva, sigurnost itd. Ta svojstva čine kvalitetu proizvoda u uporabi /4/. Softverski proizvod se vremenom i uporabom ne troši poput hardverskog proizvoda. Ipak održavanje softverskog proizvoda je značajan element koji utječe na zadovoljstvo kupca.

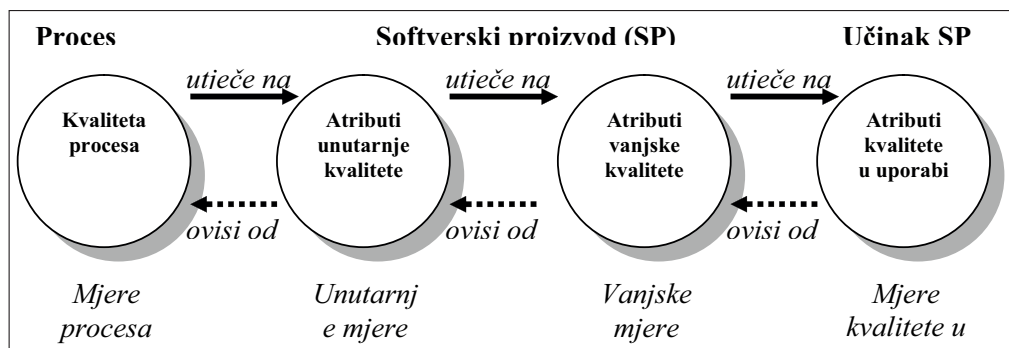
Kod hardverskih proizvoda materijal koji se prerađuje jedan je od osnovnih resursa, dok kod softvera toga nema. U poduzećima koje proizvode softver naglasak je na ljudima kao najvažnijem resursu a kod proizvodnje hardvera su bitni i strojevi za proizvodnju, koje treba uredno održavati i obnavljati. Jedan od bitnih resursa kojeg troši

proizvodnja softvera je zapravo vrijeme. Praksa je pokazala da je za realizaciju određenog softverskog proizvoda potrebno određeno vrijeme koje ovisi od veličine i kompleksnosti softvera i to se vrijeme ne može bitno skratiti bez obzira na povećanje broja ljudi koje rade na njegovoj realizaciji. Vrijeme je i inače kritičan činitelj razvoja softvera. Praksa je pokazala da korisnik vremenom mijenja svoje zahtjeve. Što je duže vrijeme razvoja softverskog proizvoda, mogu se očekivati veće izmjene u zahtjevima. Kod proizvoda čiji razvoj traje 9 do 12 mjeseci u prosjeku se promijeni oko 25% zahtjeva /5/, dakle zahtjevi se mijenjaju po stopi od oko 2% mjesečno. Uzevši u obzir presudan značaj ispunjenja zahtjeva za kvalitetu proizvoda, ova činjenica predstavlja ozbiljan problem. Pokazalo se da troškovi otklanjanja grešaka u softveru, a grešku predstavlja svaka nesukladnost sa zahtjevima, progresivno rastu s vremenom. Izmjena zahtjeva u tijeku razvoja softvera ima isti učinak kao i otklanjanje utvrđenih nesukladnosti sa zahtjevima. Kod softverskog proizvoda je kompatibilnost sa prethodnim verzijama proizvoda puno značajnija nego kod hardverskog proizvoda. Ta činjenica često utječe na neka svojstva softverskih proizvoda, jer proizvođači često traže kompromis između mogućnosti koje nudi nova tehnologija i

kompatibilnosti s prethodnim verzijama postojećeg softverskog proizvoda ili nekih drugih softverskih ili hardverskih elemenata koji su s njime povezani.

Iako je definicija kvalitete softverskog proizvoda potpuno u skladu s definicijom kvalitete za ostale proizvode, praksa je pokazala da primjena tako uopćenih definicija na softver često nije zadovoljavajuća. Razlog tome su navedene specifičnosti softvera koje utječu da u fazi iniciranja rada na izradi novog softverskog proizvoda, dakle kada treba utvrditi eksplicitne i implicitne zahtjeve na temelju kojih se utvrđuju značajne osobine tog proizvoda kao što su kvaliteta, rok izrade, cijena itd., ni kupac ni proizvođač nemaju točnu predodžbu konačnog proizvoda. Ta činjenica značajno utječe na već izneseni veliki postotak promašaja u softverskim projektima. Softverski proizvod mora zadovoljavati odgovarajuće zahtjeve i pri tome mora imati odgovarajuće značajke koje doprinose zadovoljavanju tih zahtjeva.

Slika 1. prikazuje kako se pogled na kvalitetu softverskog proizvoda mijenja tijekom njegovog životnog ciklusa /6/ tj. unutar, vanjska i kvaliteta u uporabi. Veliki dio značajki softverskog proizvoda koje se odnose na kvalitetu, nastaju u fazi njegovog razvoja.



Slika 1: Kvaliteta u životnom ciklusu softvera

Izvor: ISO/IEC 25000: Software engineering- Software product Quality Requirements and Evaluation (SQuaRe)- Guide to SQuaRe, 2005. str. 22

Na slici 1. je istaknuta neophodnost mjerenja koje s jedne strane omogućava utvrđivanje kvalitete proizvoda ali predstavlja i veliki izazov. Predodžba o kvaliteti softverskog proizvoda mijenja se ovisno o tome tko ga tijekom životnog ciklusa ocjenjuje, proizvođač, kupac ili netko treći. Kupac mijenja svoje zahtjeve u tijeku razvoja, pa s time i temeljne kriterije za ocjenu kvalitete. Kako količina promjena ovisi o vremenu, prirodna su nastojanja proizvođača da isporuči proizvod u što kraćem roku, odnosno kada se radi o velikim projektima, da ga isporučuje

u manjim dijelovima koji omogućavaju brzu isporuku.

### 3. Značajke kvalitete softvera prema ISO 25000

Međunarodna organizacija za standarde ISO objavljuje standarde već pedesetak godina. Informatička tehnologija je po količini objavljenih standarda jedna od vodećih gospodarskih grana za koju je dosad objavljeno preko 2600 standarda na oko 170.000 stranica /7/. Pregled svih standarda

koji se odnose na softver pa čak i na njegovu kvalitetu, ne može se prikazati u jednom članku. U prethodnom je tekstu navedeno kako kvaliteta softverskog proizvoda u velikom dijelu zavisi od procesa njegovog razvoja. Zbog toga ne iznenađuje da su nastojanja u osiguranju kvalitete softvera bila pretežno usmjerena na sam proces. Osim SEI CMM i ISO 9000 postoji još desetak drugih standarda odnosno metoda kojima se kvaliteta softverskog proizvoda nastoji postići putem kontrole procesa njegovog razvoja /8/. Izuzetak je međunarodni standard ISO 9126 koji je usmjeren na kvalitetu samog softverskog proizvoda, a objavljen je još 1991., a za tim njegovo revidirano izdanje 2001. godine.

Godine 1999. objavljena je grupa standarda ISO 14598: Information technology -- Software product evaluation koja je uređivala područje vrednovanja softvera. Konačno je 2005. objavljen međunarodni standard ISO/IEC 25000: Software engineering – Software product Quality Requirements

and Evaluation (SQuaRE) kao prvi u grupi međunarodnih standarda kojima se žele objediniti prethodno navedeni standardi i uvesti više reda u ovo područje. Već se u uvodu ovog dokumenta navodi da su ovi standardi usmjereni prema proizvodu, dok se upravljanje procesima prepušta grupi standarda ISO 9000 /9/.

Značajke i podznačajke kvalitete koje navodi ISO 25000 preuzete su još iz standarda ISO 9126 iz 1991. godine. Ove su značajke važne jer mogu kupcu softverskog proizvoda poslužiti kao podsjetnik i smjernica koje elementi utječu na kvalitetu softvera. Dakako, nijedan standard ne može biti zamjena za zahtjeve kupca, koji je potpuno autonoman u njihovom definiranju, ali se može očekivati da su neki od zahtjeva kupca navedeni ili dijelom sadržani u značajkama koje navodi standard. One ostale se može tretirati kao zahtjeve iz grupe 'obično se podrazumijeva ili je obvezatan'. Navode se šest glavnih značajki koje su podijeljene na ukupno 20 pod značajki koje su prikazane u tablici 1.

Tablica 1: Glavne značajke i pod značajke kvalitete softvera  
 Izvor : <http://www.sqa.net/iso9126.html> na dan 25.9.2007.

Značajka	Pod značajka	Opis
1.Funkcionalnost	1.1.Prikladnost	Svojstva softvera koja osiguravaju nazočnost i primjerenost skupa funkcija za specificirane zadace.
	1.2.Preciznost	Svojstva softvera koja doprinose dobivanju točnog ili dogovorenog rezultata.
	1.3.Međudjelova-nje	Svojstva softvera koja doprinose njegovoj sposobnosti da surađuje sa drugim specificiranim sustavima.
	1.4.Sukladnost	Svojstva softvera koja doprinose njegovoj usklađenosti sa standardima, dogovorima, zakonima i sličnim propisima.
	1.5.Sigurnost	Svojstva softvera koja doprinose sposobnosti da spriječi slučajni ili namjerni neovlašteni pristup programima i podacima
2.Pouzdanost	2.1.Zrelost	Svojstva softvera koja utječu na učestalost ispada kod nailaska na greške
	2.2.Tolerancija greške	Svojstva softvera koja doprinose njegovoj sposobnosti da zadrži određenu razinu performansi u slučaju nailaska na grešku .
	2.3.Sposobnost povratka	Svojstva softvera koja doprinose njegovoj sposobnosti da ponovno uspostavi prethodnu razinu performansi i obnovi podatke na koje je utjecala greška te vrijeme i napor koji su potrebni za to.
3.Prikladnost za uporabu	3.1.Razumljivost	Svojstva softvera koja utječu na napor korisnika da shvati logički koncept i mogućnosti primjene.
	3.2.Jednostavnost za učenje	Svojstva softvera koja utječu na napor korisnika da nauči njegovu primjenu.
	3.3.Operativnost	Svojstva softvera koja utječu na napor korisnika za njegovu redovnu uporabu i nadzor nad njom.
4.Učinkovitost	4.1.Ponašanje u odnosu na vrijeme	Svojstva softvera koja utječu na vrijeme odgovora i obrade te protoku podataka u izvođenju svojih funkcija.
	4.2.Ponašanje u odnosu na resurse	Svojstva softvera koja utječu na količinu korištenih resursa i vrijeme korištenja u izvođenju svojih funkcija.
5.Prikladnost za održavanje	5.1.Prikladnost za analizu	Svojstva softvera koja utječu na napor kod dijagnosticiranja slabih točaka, uzroka ispada ili identifikaciju dijela kojeg treba mijenjati .
	5.2.Jednostavnost izmjena	Svojstva softvera koja utječu na napor potreban da se izvrši izmjena, otkloni greška ili promijeni okolina.
	5.3.Stabilnost	Svojstva softvera koja utječu rizik od neočekivanog rezultata ili promjene ponašanja.
	5.4.Prikladnost za testiranje	Svojstva softvera koja utječu na napor potreban da se ovjeri izmijenjeni softver.
6.Prenosivost	6.1.Prilagodljivost	Svojstva softvera koja utječu na mogućnost prilagodbe različitim specificiranim okolinama bez da se poduzimaju dodatne aktivnosti.
	6.2.Prikladnost za ugradnju	Svojstva softvera koja utječu na napor potreban da ga se ugradi na specificiranu okolinu.
	6.3.Usklađenost	Svojstva softvera koja osiguravaju da njegovu usklađenost sa standardima i pravilima u odnosu na prenosivost.
	6.4.Prikladnost za zamjenu	Svojstva softvera koja utječu na mogućnosti i napor potreban da se zamijeni drugim softverom u okolini u kojoj djeluje.



Vidljivo je da navedene značajke i pod značajke koje utječu na kvalitetu softverskog proizvoda pretežno pokrivaju aspekte vanjske kvalitete i kvalitete u uporabi iako ima i elemenata koji se odnose na unutarnju kvalitetu, kao na primjer 1.4. Sukladnost i 2.1. Zrelost.

Pitanje je i koliku relativnu težinu treba dati pojedinoj značajki. Tu nema jedinstvenog odgovora jer relativna težina pojedinačne značajke ovisi o vrsti proizvoda, ali i o kupcu. Na primjer, ako se radi o softveru za upravljanje zrakoplovima pouzdanost ima zasigurno veliki značaj, dok je kod softvera za video igre ona manje značajna. Predstavnik kupca koji je na razini uprave će vjerojatno staviti naglasak na funkcionalnost i na one značajke koje utječu na visinu troškova nabavke i eksploatacije softvera, dok će neposredni korisnik tog istog softvera vjerojatno veći značaj davati prikladnosti za uporabu.

Kada je potencijalni kupac u prilici da nabavlja novi softverski proizvod on ima u načelu dvije različite mogućnosti:

1. Nabaviti proizvod koji se izrađuje 'po mjeri' prema njegovim zahtjevima.
2. Izabrati gotovi proizvod koji zadovoljava njegove zahtjeve.

U prvom slučaju on može utjecati na sam proces razvoja softvera, dok u drugom slučaju ne.

Procjena procesa kod potencijalnog dobavljača je zasigurno jedan od bitnih kriterija za izbor. Međutim

čak i kada nabavlja gotov proizvod, dakle kad kupac više ne može utjecati na sam proces, ocjena procesa potencijalnih dobavljača može biti važan element za procjenu kvalitete softvera. U oba slučaja mora biti definiran sustav mjerenja procesa ali i značajki i pod značajki softvera, kako bi se mogle izvršiti odgovarajuće usporedbe.

U tablici 2. je prikazan konkretan primjer koji ilustrira kako može izgledati sustav mjerenja i koliko sam proces može utjecati na neke elemente kvalitete softvera. Prema SEI- CMM metodi svi proizvođači softvera mogu svrstati u odnosu na zrelost procesa u 5 stupnjeva. Prvi je stupanj najniži i dodjeljuje se proizvođačima koji su u inicijalnoj fazi zrelosti tj proizvode u manje ili više kaotičnim uvjetima, dok je stupanj pet najviši. Taj je model razvijen u SAD gdje se najviše primjenjuje, i pokazalo se da velika većina proizvođača softvera ne prelazi stupanj zrelosti 3. Procjenjuje se da je proizvođač softvera koji ima certificiran sustav upravljanja kvalitetom prema zahtjevima ISO 9001 u pravilu blizu razine zrelosti 3. Ipak ne postoje podaci o stupnju zrelosti proizvođača softvera u Europi ili Hrvatskoj. U Italiji su 2001. godine procijenili da barem 85% proizvođača softvera ne prelazi stupanj 1/10/. Nema razloga vjerovati da je u Hrvatskoj stanje bitno bolje.

CMM stupanj zrelosti	Trajanje projekta (mjeseci)	Broj angažiranih osoba*mjeseci	Isporučeni broj grešaka	Prosječni ostvareni troškovi
1	30	600	61	5,5
2	18,5	143	12	1,3
3	15	80	7	0,73

Tablica 2: CMM statistika za projekte od 200.000 LOC

Izvor: <http://www.so.e.ucsc.edu/~yiz/ism58/files/lecture-3.pdf> na dan 26.09.07.

Uobičajena mjera za veličinu softvera je LOC (Lines Of Code) odnosno broj naredbi u softveru. U tablici su uspoređeni podaci za proizvode iste veličine (ne govori se o kompleksnosti koja također može imati značajan utjecaj). Dakle, proizvođači čiji je proces ocijenjen sa stupnjem zrelosti 3 propustili su skoro devet puta manje grešaka nego oni u stupnju 1 (7 u odnosu na 61), a ukupni su troškovi manji za gotovo 8 puta, dok je trajanje projekta prepolovljeno. Razlike su ogromne i umjesno je pitanje zbog čega se proizvođači ne trude da postignu veći stupanj zrelosti. Odgovor je vjerojatno

u tržištu, na kojem nisu razvijeni odgovarajući mehanizmi 'prisile'.

#### 4. Sustav znanja u potpori identificiranju važnosti značajki kvalitete softverskog proizvoda

U cilju otkrivanja pravila između značajki kvalitete softverskog proizvoda, kreiran je upitnik (Prilog 1.) u kojem se nalaze osnovne značajke iz Tablice 1. sa svojim vrijednostima. Provedeno je istraživanje na uzorku razvojnog tima proizvođača softvera koji podržava rad medicinskih ustanova i njihovih

korisnika. Riječ je o predstavniku proizvođača softverskog rješenja koji više od petnaest godina uspješno posluje na cijelom području Hrvatske. Veličina uzorka je relativno mala i prikupljeno je 13 odgovora korisnika i 8 mišljenja članova razvojnog tima, ali sasvim dovoljna za potrebe istraživanja. I tako mali broj odgovora mogao je poslužiti kao ulaz u sustav znanja i pokazalo se, pružiti korisne odgovore menadžmentu poduzeća. Popunjeni upitnici su poslužili kao ulazni podaci (slučajevi) u bazu znanja slučajeva na čijem temelju je bilo moguće izvesti induktivno zaključivanje. Korištena je ljuska znanja Doctus /11/ koja se koristi simboličkom umjetnom inteligencijom prilikom izvođenja tri vrste zaključivanja: dedukcije, indukcije i redukcije. U ovisnosti o znanju i iskustvu donositelja odluka, omogućeno je manipuliranje i vizualizacija implicitnih i/ili prešutnih pravila u obliku "Ako... Tada" kao temelja pronalaženja zadovoljavajućeg rješenja problema odlučivanja. Ljuska omogućuje tri vrste zaključivanja koje kao rezultat pružaju korisniku i grafički prikaz objašnjenja ponuđene alternative. U slučajevima postojanja kvantitativnih ulaznih podataka iz baze podataka (ili skladišta podataka) omogućeno je automatsko povezivanje slučajeva s podacima u cilju rudarenja po podacima. Na taj način je omogućeno prepoznavanje skrivenih i do tada nepoznatih veza u podacima i njihovo uključivanje u model odlučivanja. Integracijom baze znanja s inteligentnim portalom, omogućeno je njeno publiciranje na Web-u i njeno stalno punjenje novim slučajevima odnosno pravilima /12/. U ovom slučaju nisu poznata pravila između vrijednosti atributa, odnosno značajki kvalitete softvera i zaključivanjem se dobivaju induksijska stabla koja opisuju percipirane značajke kvalitete softvera proizvođača i korisnika. Drugim riječima, ukoliko se korisnik izjasnio da je softverski proizvod "kvalitetan" željelo se odgovoriti na pitanje zašto je odgovor takav kakav jeste, odnosno postoje li pravila ili uzorak kojim je moguće objasniti takav odgovor.

Induksijsko zaključivanje kod Doctus ljuske vrši se primjenom modificiranog C4.5 algoritma koji predstavlja unaprijeđenu verziju originalnog ID3 algoritma. Za izgradnju induksijskog stabla koristi se algoritam opisan sljedećim pseudokodom:

Funkcioniranje ID3 algoritma ( $\beta$ : je skup nekategorijskih atributa,  $C_i$  je kategorijski atribut,  $C$ : skup podataka za trening); Početak: Ako je  $C$  prazan, prikaži jedan čvor s vrijednošću "Neuspjeh"; Ako se  $C$  sastoji od slogova podataka s istom vrijednošću kategorijskog atributa, prikaži jedan čvor s njegovom vrijednošću;

Ako je  $\beta$  prazan skup, za rezultat prikazati jedan čvor s vrijednošću najfrekventnije vrijednosti

kategorijskog atributa pronađenog u skupu slogova  $C$ ; Neka  $C_i$  bude atribut s najvišom informacijskom vrijednošću  $Informativity I_b(C_i, C)$  između atributa u  $\beta$ ; Neka  $\{w_i | i=1,2, \dots, m\}$  budu vrijednosti atributa  $C_i$ ; Neka  $\{C_i | i=1,2, \dots, m\}$  bude podskup od  $C$  koji se sastoji od slogova s vrijednošću  $w_i$  za atribut  $C_i$ ; Prikazati stablo s korijenom označenim s  $C_i$  i stablima označenim  $a_1, a_2, \dots, a_m$  dok se ne prikaže stablo  $ID3(\beta - \{C_i\}, C_i, C1)$ ,  $ID3(\beta - \{C_i\}, C_i, C2)$ , ...,  $ID3(\beta - \{C_i\}, C_i, C_m)$ ; Kraj.

Odrediti pokazatelj  $I_b$  za svaki atribut. Pokazatelj opisuje informacijsku dobit ili smanjivanje entropije dijeljenjem. Pokazatelj  $Informativity (I_b)$  se izračunava na sljedeći način: Neka je  $C$  skup slučajeva u čvoru, a označava  $a_1 \dots a_n$  njihovih vrijednosti, i  $w_{a_1} \dots w_{a_n}$  ( $\sum_i w_{a_i} = 1$ ) njihovo ponavljanje u skupu  $C$ . Tada se entropija grananja skupa  $C$  može napisati kao:  $E_C = -\sum_i w_{a_i} \log_n w_{a_i}$ .

Neka su  $b_1 \dots b_n$  vrijednosti atributa  $b$ ,  $\beta$  je skup tih vrijednosti.  $\beta$  se dijeli u neprazan podskup nazvan  $\beta_1 \dots \beta_m$  i tada je  $\sum_i \beta_i = \beta$ . Skup  $C$  se dijeli u podskup nazvan  $C_1 \dots C_m$  i postaje atribut od  $b$  za sve elemente  $C_i$  u  $\beta_i$  za svaki  $i$ . Neka  $w_i$  označava važnost  $C_i$  u  $C$ . ( $\sum_i w_i = 1$ ). Tada  $I_b = E_C - \sum_i w_i E_{C_i}$  predstavlja porast entropije (odnosno smanjenje informacijske vrijednosti) koja je rezultirala dijeljenjem  $\beta_1 \dots \beta_m$ . Dijeljenje skupa se zaustavlja pri pronalaženju optimalne vrijednosti odnosno pri utvrđivanju  $I_{bmax}$ . Riječ je o strojnom učenju u kojem se rezultat dobiva u obliku stabla odlučivanja i koje primjenjuje neinkrementalno nenadzirano strojno učenje. Kod ove skupine metoda strojnog učenja za generiranje rezultata (induksijskog stabla) potrebno je obraditi cijeli skup ulaznih podataka, te je prilikom dodavanja nove vrijednosti potrebno ponoviti cjelokupnu obradu početnog skupa /13/.

Atributi značajki kvalitete softverskog proizvoda sa svojim vrijednostima su prikazane tablicom 3.

Odgovori korisnika su uneseni u zasebnu bazu slučajeva, a odgovori članova razvojnog tima u drugu. Kako bi omogućili usporedbu rezultata zaključivanja, obje su baze znanja utemeljene na istim atributima iz tablica 3., odnosno obje skupine su odgovarale na ista pitanja. Koristeći skalu od pet simboličkih vrijednosti (neodgovarajući, nekvalitetan, zadovoljavajući, kvalitetan i vrhunski) korisnici su softverski proizvod ocijenili kao kvalitetan (69,2%) i vrhunski (30,8%), dok su predstavnici razvojnog tima bili samokritičniji i kvalitetu svog softverskog proizvoda ocijenili s ocjenom zadovoljava (50%) i kvalitetan (50%). Nakon primjene induksijskog zaključivanja nad ove dvije skupine odgovora dobivena su induksijska stabla prikazana slikom 2. Desni dio zaslona prikazan slikom 2. prikazuje značajke kvalitete softverskog proizvoda onako kako ga percipiraju

Name	Value 1	Value 2	Value 3	Value 4	Value 5
Kvaliteta softvera	Neodgovarajući	Nekvalitetan	Zadovoljavajući	Kvalitetan	Vrhunski
Prikladnost	F-ije nisu nazočne	F-ije često neodgovarajuće	F-ije ponekad neodgovarajuće	F-ije primjeren	
Preciznost	Ne doprinosi dobivanju rez	Doprinosi točnosti rezultata	Točan ili dogovoren rezultat		
Međudjelovanje (suradnja s ostalim alatima)	Moguće i otežano	Jednostavno, uz stručnjaka	Svakodnevno		
Sukladnost (sa standardima)	Nije uskladen uopće	U nekim elementima	Kontinuirano		
Sigurnost	Slobodan pristup	Samo ovlastene osobe			
Zrelost (Sustav je van funkcije)	Često van funkcije	Povremeno van funkcije	Nikada nije izvan f-ije		
Tolerancija greške (Reakcija na grešku)	Pad sustava	Odredjena razina performan			
Sposobnost povratka	Znacajan napor	Ne zahtijeva napor/vrijeme			
Razumljivost koncepta i primjene	Uz napor korisnika	Ne zahtijeva poseban napor			
Jednostavnost učenja	Posebna obuka i učenje	Jednostavno/intuitivno			
Operativnost (nadzor rezultata)	Zahtijeva dodatni napor	Jednostavno/lako			
Ponasanje u odnosu na vrijeme	Sporo i neucinkovito	Brzo i ucinkovito			
Ponasanje u odnosu na resurse	Velika količina resursa/vre	Ne zahtijeva veliku količinu r			
Prikladnost za analizu	Uz značajane napore	Brzo i jednostavno (prikladn			
Jednostavnost izmjena	Komplicirano/dugotrajno	Jednostavno/brzo			
Stabilnost (Rizik dobivanja neočekivanog rez	visok	postoji/nije značajan	zanemariv		
Prikladnost za testiranje (Uvodjenje novih ver	uz dodatna odobrenja	bez posebnih zahtjeva			
Prilagodljivost specifičnim okolinama	tesko	uz nekoliko dodatnih aktivnc	lako		
Prikladnost za ugradnju u spec. okolinu	nemoguće	izvodivo, ali zahtjevno	jednostavno		
Uskladenost sa standardima prenosivosti	Ne postoji	Postoji			
Prikladnost za zamjenu	Otežana/ nemoguća	izvodiva			

Tablica 3: Vrijednosti atributa značajki softverskog proizvoda sustava znanja  
Izvor: Pripremili autori

članovi tima proizvođača. Na vrhu su prikazane značajke koje imaju najvišu informacijsku vrijednost i pomoću kojih se može iskazati pravilo da bez obzira na vrijednosti ostalih značajki softverskog proizvoda da "Ako značajka "Jednostavnost učenja" ima vrijednost "Jednostavno ili intuitivno" tada je softver "Kvalitetan" ili "Ako je atribut "Jednostavnost učenja" s vrijednošću "Zahtijeva posebnu obuku i učenje" tada je softver "Zadovoljavajući". U ovom slučaju proizvođač softvera promatra značajke kvalitete softvera s aspekta korisnika (aspekti vanjske kvalitete), no kako je riječ o softverskom proizvodu koji se nalazi u svojoj zreloj fazi razvoja, ujedno je riječ i o pokazatelju kvalitete u uporabi. Usporedbom opisanog induksijskog stabla proizvođača s onim na lijevoj strani zaslona (Slika 2.) evidentna je razlika u percepciji važnosti značajki kvalitete softvera korisnika i proizvođača. Odgovori korisnika nisu bili toliko konzistentni što je rezultiralo "razgranatijim" induksijskim stablom.

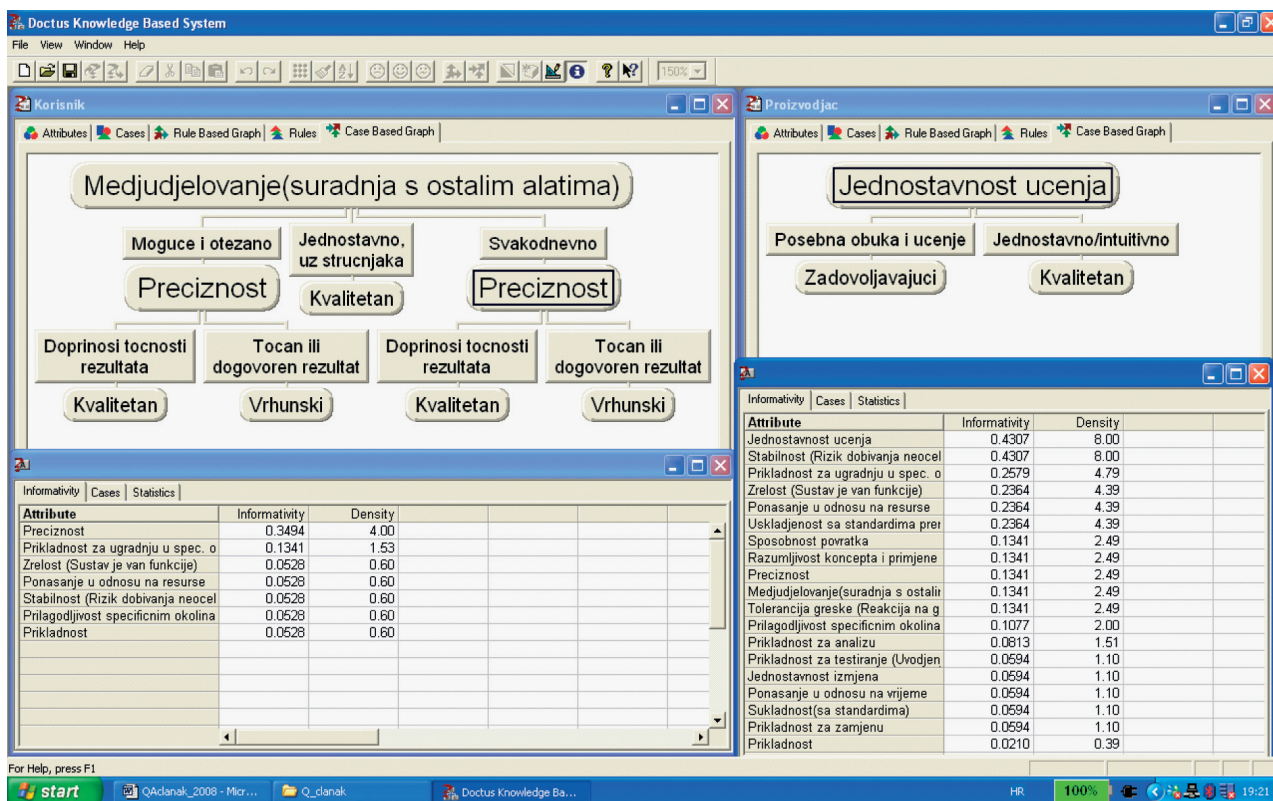
Korisnici između ostalog smatraju da: ako značajka "Međudjelovanje (suradnja s ostalim alatima)" ima vrijednost "Jednostavno, ali uz prisutnost stručnjaka", tada je softver "Kvalitetan" ili ako značajka "Međudjelovanje (suradnja s ostalim alatima)

ima vrijednost "Svakodnevno" i značajka "Preciznost" ima vrijednost "Točan ili dogovoren rezultat" tada je softver "Vrhunski". Ovim su utvrđene značajke kvalitetnog softvera korisnika da što jednostavnije, bez pomoći stručnjaka i svakodnevno mogu integrirati softver s ostalim potrebnim alatima, kao i postojanje napora proizvođača softvera na jednom sasvim drugom planu: unapređenju potpore i omogućavanju jednostavnosti učenja. Nakon dodatnih upita proizvođaču softvera, utvrđeno je da se različitost percepcije značajki ne može objasniti već učinjenim naporima proizvođača na rješavanju "Međudjelovanja (suradnje s ostalim alatima)" i potrebi naknadne edukacije korisnika kako koristiti nove funkcionalnosti, te da je ovim putem signalizirana potreba poduzimanja dodatnih aktivnosti.

## 5. Zaključak

Suvremeno poslovanje postavlja jasne i stroge zahtjeve proizvođačima poslovnih softverskih rješenja za isporukom kvalitetnog proizvoda, što je brže moguće i u okviru utvrđenog budžeta, kako bi se adekvatno odgovorilo pritiscima konkurencije i promjenjivim zahtjevima tržišta. Tri su osnovna ograničenja ispunjenju tih zahtjeva: troškovi,





Slika 2: Značajke kvalitete softverskog proizvoda korisnika i članova razvojnog tima proizvođača softvera (Izvor: Pripremili autori)

kvaliteta i vrijeme. I bez dodatnog opterećenja zahtjevima upravljanja kvalitetom, većina današnjih projekata razvoja softvera premašuje planirane budžete i vrijeme razvoja. Jedno od rješenja jeste izvođenje aktivnosti upravljanja kvalitetom kontinuirano, upravljano i automatizirano tijekom cijelog životnog ciklusa softverskog proizvoda. Smatra se poželjnim s ovim aktivnostima započeti što ranije u životnom ciklusu softvera, izvoditi ih iterativno i kontinuirano, utvrditi metriku kojom je moguće pratiti kvalitetu, te pravodobno poduzimati korektivne aktivnosti.

Izgradnjom i primjenom sustava znanja, utvrđena je snimka stanja postojanja različite percepcije softverske kvalitete korisnika i proizvođača, i inducirana pravila na temelju kojih je procijenjena kvaliteta u danom trenutku. Time je omogućeno otkrivanje i dijagnosticiranje problema, stvoren je temelj za poduzimanje korektivnih aktivnosti kao i za jednostavnu komunikaciju korisnika i proizvođača.

## Bilješke

- 1/ Procjenjuje se da su svjetski ukupni troškovi vezani za proizvodnju softvera veći od ukupnih troškova proizvodnje automobila!
- 2/ Iako su pojmovi kupac i korisnik u načelu sinonimi, uobičajeno je da se pod pojmom kupac podrazumijevaju sve zainteresirane strane za proizvod, dok pojam korisnik asocira na onog tko ga neposredno koristi.
- 3/ SEI je skraćenica od Software Engineering Institute. Više o tome može se vidjeti na primjer <http://www.sei.cmu.edu/pub/documents/95.reports/pdf/mm003.95.pdf>
- 4/ Vidjeti slika 1.
- 5/ Preuzeto sa <http://codebetter.com/blogs/darrell.norton/articles/50341.aspx> na dan 23.09.07.
- 6/ Životni ciklus softvera je vremensko razdoblje koje započinje sa zamisli da se izradi određeni softverski proizvod i završava s prestankom korištenja tog proizvoda.
- 7/ Vidjeti [www.iso.org](http://www.iso.org)
- 8/ Više o tome može se vidjeti u Sinković, Đ.: Quality Management in Software Production, Proceedings of the 14<sup>th</sup> International Scientific Conference on Information and Intelligent Systems, Varaždin, 2003.
- 9/ Više o tome može se vidjeti u Sinković G-Bevanda V: Standardi informacijsko/ komunikacijske tehnologije (IKT), The 14th International scientific conference "Society and Technology 2007" (The Position and Role of Electronic Media - Convergence of the Media), Split, June 28-30 2007, Informatologia Separat Speciale 11, 2007, 1-97.
- 10/ Preuzeto sa [http://www.tecnoteca.it/sezioni/qualita/norme/sei\\_cmm](http://www.tecnoteca.it/sezioni/qualita/norme/sei_cmm) na dan 1.10.2007.



/11/demo verzija alata je dostupna na <http://www.doctus.info>

/12/ Baracska, Z.; Bevanda, V., Dorfler, V. (2002) Intelligent Customer Relationship Management (iCRM) by eFlow Intelligent Portal, 13th International Conference on Information and Intelligent Systems, Conference Proceedings, Varaždin, 173-180.

/13/Martínez-Enríquez, A.M. & Eschalada-Imaz, G. (1998). The revision of inductive learning theory within incomplete and imprecise observations. *Expert Systems with Application*, 15, pp. 357-366. Available from: <http://elsevier.com/locate/eswa> . Accessed: 2000-06-03.

## Literatura

1. Baracska, Z.; Bevanda, V., Dorfler, V. (2002) Intelligent Customer Relationship Management (iCRM) by eFlow Intelligent Portal, 13th International Conference on Information
2. HR EN ISO 9001: Sustavi upravljanja kvalitetom – Zahtjevi
3. Intelligent Systems, Conference Proceedings, Varaždin, 173-180.
4. ISO/IEC 90003: Software engineering- Guidelines for the application of ISO 9001:2000 to computer softver
5. ISO/IEC 25000: Software engineering- Software product Quality Requirements and Evaluation (SQuaRe)- Guide to SQuaRe, 2005.

6. Juran J.M.-Gryna F.M: Planiranje i analiza kvalitete (treće izdanje) MATE, Zagreb,1999.

7. Krasner H. Using cost of quality approach for software <http://www.compaid.com/caiInternet/casestudies/krasner-CoSQ-talk.pdf> str 4. na dan 25.9.2007

8. Martínez-Enríquez, A.M. & Eschalada-Imaz, G. (1998). The revision of inductive learning theory within incomplete and imprecise observations. *Expert Systems with Application*, 15, pp. 357-366. Available from: <http://elsevier.com/locate/eswa> . Accessed: 2000-06-03.

9. Petrasch R: The definition of 'Software Quality' A practical Approach

<http://www.chillarege.com/fastabstracts/issre99/99124.pdf> na dan 17.09.07.

10. Sinković, Đ. : Quality Management in Software Production, Proceedings of the 14<sup>th</sup> International Scientific Conference on Information and Intelligent Systems, Varaždin, 2003.

11. Sinković G-Bevanda V: Standardi informacijsko-komunikacijske tehnologije (IKT), The 14th International scientific conference "Society and Technology 2007" (The Position and Role of Electronic Media - Convergence of the Media), Split, June 28-30 2007, Informatologia Separat Speciale 11, 2007, 1-97.

12. Quinlan, J. R. (1992) C 4.5: Programs for Machine Learning. *Morgan Kauffman*, ISBN 1558602380.

13. Webster M: Leveraging Static Analysis for Multidimensional View of Software Quality and Security: Klocwork's Solution, White paper, 2005 ([www.idc.com](http://www.idc.com))

