# OBEZBEĐENJE KVALITETA SOFTVERA

# SOFTWARE QUALITY ASSURANCE

dr Miladin Stefanović*, dr Zora Arsovski**, dr Milan Matijević*
* Mašinski fakultet, Kragujevac
** Ekonomski fakultet, Kragujevac

**Rezime:** *Kvalitet softvera je više dimenzionalni koncept koji se ne može jednostavno definisati. Potrebno je pratiti različite parametre i obezbediti kvalitet softvera, kreirati planove vezane za kvalitet i ostvariti potrebnu kontrolu. Potrebno je razviti i implementirati standarde sistema i dokumentaciju sistema kvaliteta kvalieta koja će biti primenjena na softverske proizvode. Tema ovog rada je obezbeđenje kvaliteta softvera.*
**Ključne reči:** *obezbeđenje kvaliteta, softver, dokumentacija sistama kvaliteta*

**Abstract:** *Quality of software is a multidimensional concept that is not easy to define in a simple way. In order to achieve quality we have to take care about quality assurance, quality planning and quality control. It is neccessarz to develop and implement qualitz standards and socumentation of qualitz szstem for software products. In this paper we will present road to software qaulity assurance.*
**Key words:** *Quality Assurance, Software, Quality System*

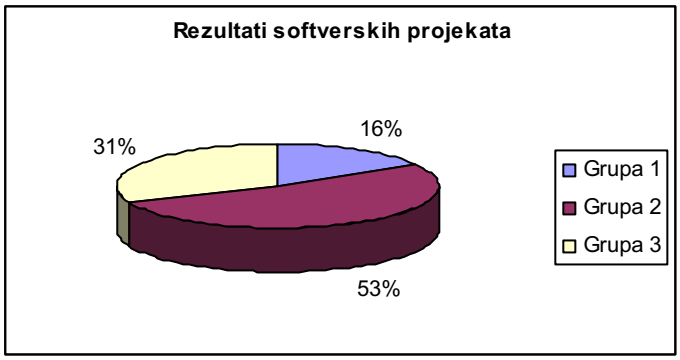
## 1. Uvod

Softveri različitih veličina i namena imaju ključnu ulogu u upravljanju uređajima i procesima u modernom poslovnom i životnom okruženju. Počev od softvera koji sadrže nekoliko linija koda pa do sftvera koji sadrži dva miliona linija koda i upravlja sa četiri hiljade procesora i zazlužan je za funkcionisanje Boinga 777 [1]. Sama pomisao da kvalitet softvera u ovakovom sistemu može biti nizak je prilično zabrinjavajuća ideja.
Postavlja se pitanje kakav je nivo kvaliteta i pouzdanosti softvera u softverskoj i drugoj industriji izuzev, naravno, ovako ozbiljnih sistema gde ljudski životi zavise od sigurnosti, pouzdanosti i kvaliteta softvera.
Softverska industrija je pod pritiskom zahteva tržišta da softverski proizvodi budu kvalitetni i pouzdani, ona troši oko 50% svog budžeta na aktivnosti koje imaju za cilj povećanje kvaliteta softvera. Nažalost, najveći broj ovih kompanija troše ove resurse u fazi testiranja, i ispravljanju onih defekta koji su se pojavili u naprednijim fazama razvoja softvera.
Studija koju je sprovela Jet Propulsion Laboratory (JPL) [2] pokazuje da cena detekcije i uklanjanja grešaka raste po stopi od 10 puta kako se softverski proizvod kreće kroz svoj životni ciklus (definisanje zahteva, dizajn, implementacija, testiranje, puštanje u rad). Ova studija je pokazala da je cena detektovanja i ispravljanja greške u softverskom proizvodu stoji u razmeri 1:10:100:1000 (zahtevi : dizajn : implementacija/testiranje : eksploatacija).
Iz svega ovoga se vidi koliko detekcija i ispravljanje softverskih grečaka može biti skupa aktivnost. Ako svemu ovome dodamo rastuće zahteve za kvalitetom i pouzdanošću softvera koje potrošači postavljaju pred softversku industriju, jasno je da samo implementacija sistema kvaliteta i obezbeđenje kvaliteta softvera može doneti pozitivne rezultate.
Ovaj rad se bavi definisanjem puta za obezbeđivanje kvaliteta softvera kroz definisanje modela kvaliteta softvera, prateće dokumentacije sistema kvaliteta i prezentacije softverskog rešenje za podršku obezbeđivanju kvaliteta softvera.

## 2. Softverski projekti i kvalitet softvera

U svetu se godišnje u softverske projekte uloži više milijardi dolara. Prema Standish Grupi [3] godišenje u USA se realizuje preko 175 000 softverskih projekata. Veliki projekti imaju prosečnu cenu $2,322,000; projekti u srednjim kompanijama $1,331,000; a projekti u malim kompanijama u proseku vrede $434,000. Međutim, veliki broj ovih projekata se završi neuspehom. Na slici 1 su prikazani rezultati softverskih projekata, gde:

- Grupu 1 - čine uspešni projekti, koji su završeni u skaldu sa predviđenim rokom, u okviru planiranog bužeta i sa svim funkcijama i osobinama koje su inicijalno planirane.
- Grupu 2 - čine projekti koji su kompletirani, ali su premašili budžet i predviđene rokove, i koji nude manji broj funkcija i osobina nego što je to inicijalno specificirano.
- Grupu 3 - čine neuspešni projekti, odnosno oni projekti koji su obustavljeni u nekoj tački razvojnog ciklusa.




***Slika 1 – Rezultati softverskih projekata***

Takođe je zanimljivo pogredati broj i strukturu projekata koji su prekoračili definisane vremenske rokove i definisana finansijska sredstva [3].

| Prekoračenje budžeta | % Projekata |
|---|---|
| Ispod 20% | 15.5% |
| 21 - 50% | 31.5% |
| 51 - 100% | 29.6% |
| 101 - 200% | 10.2% |
| 201 - 400% | 8.8% |
| Preko 400% | 4.4% |

***Tabla 1- Prekoračenje budžeta kod softverskih projekata***

| Prekoračenje vremena | % Projekata |
|---|---|
| Under 20% | 13.9% |
| 21 - 50% | 18.3% |
| 51 - 100% | 20.0% |
| 101 - 200% | 35.5% |
| 201 - 400% | 11.2% |
| Preko 400% | 1.1% |

***Tabela 2 – Prekoračenje vremena kod softverskih projekata***

Slika 1 i tabele 1 i 2 ukazuju da se značajna finansijska sredstva gube usled prekoračenja vremena i budžeta kod softverskih projekata, a da se veliki broj softverskih projekata nikad i ne završi.

Kao ključna posledica javlja se veliki broj otkaza i grešaka kod gotovih softverskih rešenja (tabela 3).

***Tabela 3 – Slučajevi pojave otkaza kod gotovih softverskih proizvoda.***

| | Rezultati iz 2000 | Rezulati iz 1995 |
|---|---|---|
| Značajan broj grešaka | 27% | 17% |
| Više grešaka | 21% | 29% |
| Bez promene | 11% | 23% |
| Manje grešaka | 19% | 23% |
| Značajno manje grešaka | 22% | 8% |

Uprkos činjenici da su tehnologije, alati za razvoj softvera uznapredovali i dalje je značlajan broj grešaka i otkaza koji se javljaju kod gotovih softverskih proizvoda.
Pri samom razvoju softverskih proizvoda javljaju se greške, koje variraju od zemlje do zemlje tabela 4.

***Tabela 4 - Broj gre{aka na 1000 linija izdatog koda u softveru po zemljama***

| rb | Zemlje | Broj grešaka | rb | Zemlje | Broj grešaka |
|---|---|---|---|---|---|
| 1 | USA | 1.6 | 8 | Švajcarska | 2.5 |
| 2 | Japan | 1.8 | 9 | Indija | 2.6 |
| 3 | Engleska | 2.1 | 10 | Grčka | 2.8 |
| 4 | Nemačka | 2.2 | 11 | Norveška | 3.6 |
| 5 | Izrael | 2.3 | 12 | Irska | 3.7 |
| 6 | Italija | 2.5 | 13 | Kanada | 3.9 |
| 7 | Francuska | 2.5 | | Srbija i Crna Gora | 4.6 |

Bolja realizacija projekata razvoja softvera i kvalitetniji softverski proizvodi mogu se dobiti kroz:

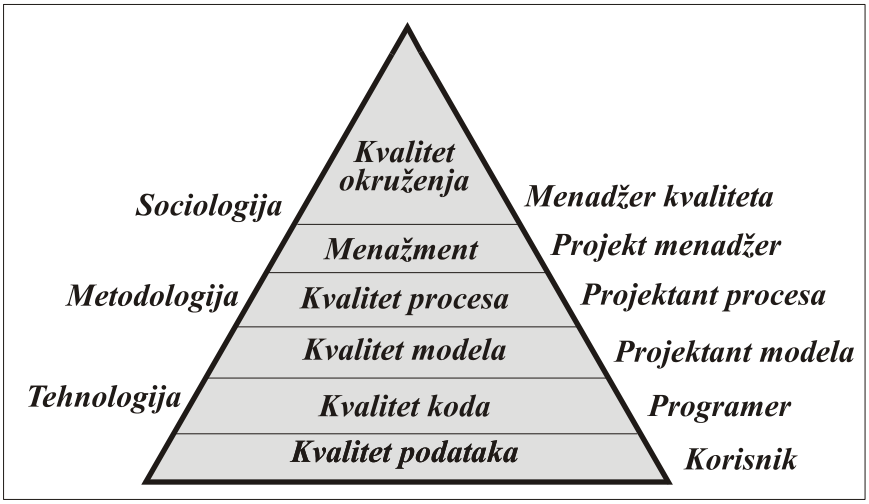
- Unapređenje metodologija upravljanja projektima i implementacija starih i unapređenih rešenja na probleme upravljanja softverskim projektima.
- Implementacija i razvijanje standarda sistema kvaliteta koji će biti primenjeni na softverske proizvode.

Jedino se korišćenjem ova dva pristupa može osigurati bolje vođenje softverskih projekata i dobijanje kvalitetnijih softverskih proizvoda.
Obezbeđenje kvaliteta (*Quality Assurance*) predstavlja jednu od vodeđih paradigmi modernog poslovanja, sa druge strane imamo informacione tehnologije kao strateški alat unapređivanja poslovanja i svakog procesa. Obezbeđenje kvaliteta softvera (*Software Quality Assurance* SQA) je više dimenzionalni koncept koji se ne može jednostavno definisati. Pri određivanju kvaliteta softvera potrebno je meriti više parametara. Poseban problem predstavlja definisanje metrike softvera, odnosno onog tipa merenja koji se odnosi na softverski sistem, proces ili pripadajuću dokumentaciju. Pri tome je neophodno izvršiti izbor parametara koji se mere i obezbediti testiranje softverskog sistema koristeći potrebne startegije i pristupe u validacionom testiranju. Sve ovo je potrebno da bi dobili softverski proizvod koji zadovoljava zahteve kupaca, koji je razvijen u skladu sa specifikacijom i koji ne sadrži greške. Naročito je bitno da se razvoj softvera odvija u sakladu sa unapred definisanim standardno propisanim procedurama da bi se umanjile intervencija na popravci i modifikaciji softverskih rešenja u kasnijim fazama u toku životnog ciklusa softverskog proizvoda.

## 3. Kvalitet softvera

U principu jedna od ideja kvaliteta proizvoda je težnja da taj proizvod bude izrađen u skladu sa specifikacijom (Crosby 1979). Međutim postoji nekoliko specifičnosti koje kvalitet softvera odvajaju od kvaliteta ostalih "klasičnih" proizvoda [4]:

- Specifikacija svakog proizvoda ide ka ispunjavanju želja i potreba kupca. Međutim, kod razvoja softvera često postoje zahtevi koji nisu izričito specificirani (na priemr zahtev za stabilnošću softvera).
- Ne postoji pouzdan način za kvantifikaciju pojedinih karakteristika softvera (na primer: stabilnost, pouzdanost).
- Softverska specifikacija je često nekompletna.

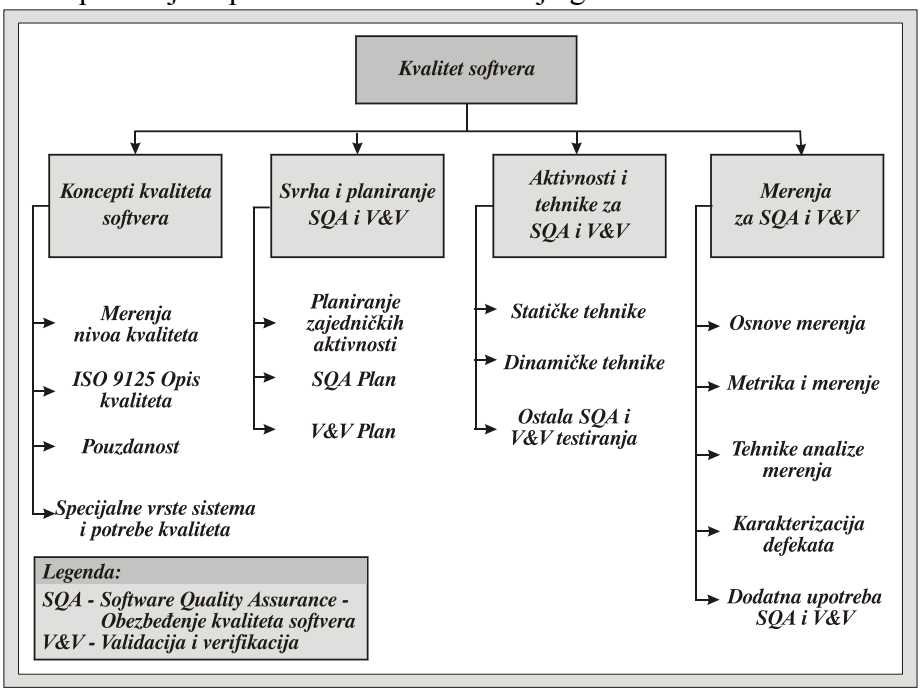



***Slika 2 – Kontekst kvaliteta softvera [10]***

S toga je kod menadžmenta kvalitetom softvera nepohodno voditi računa o sledećim aktivnostima:

- Obezbeđivanju kvaliteta, koji obezbeđuje da organizacija prati ciljeve kvaliteta, što uključuje definisanje i izbor različitih standarda koji se uključuju u proces razvoja softvera ili sam softverski proizvod. Izabrani standardi mogu biti ISO 9000 ili JUS/ISO 12207 - Informaciona tehnologija - procesi životnog ciklusa softvera; i/ili neki nacionalni standardi, kao na primer BS 5750. Svako preduzeće prema izabranom sistemu kvaliteta kreira svoj poslovnik o kvalitetu iz koga proističe dokumentacija sistema kvaliteta koja, pak, obezbeđuje kvalitet odvijanja odgovarajućih procesa u organizaciji i služi za kvaliteno vođenje različitih projekata, među kojima su i projekti razvoja softvera.
- Kontrola kvaliteta obezbeđuje da se proces razvoja softvera odvija prema definisanim i izabranim standardima.
- Planiranju kvaliteta koji treba da počne u ranim fazama razvoja softvera. Planovi kvaliteta treba da definišu željeni set karakteristika koje softverski proizvod treba da ima.

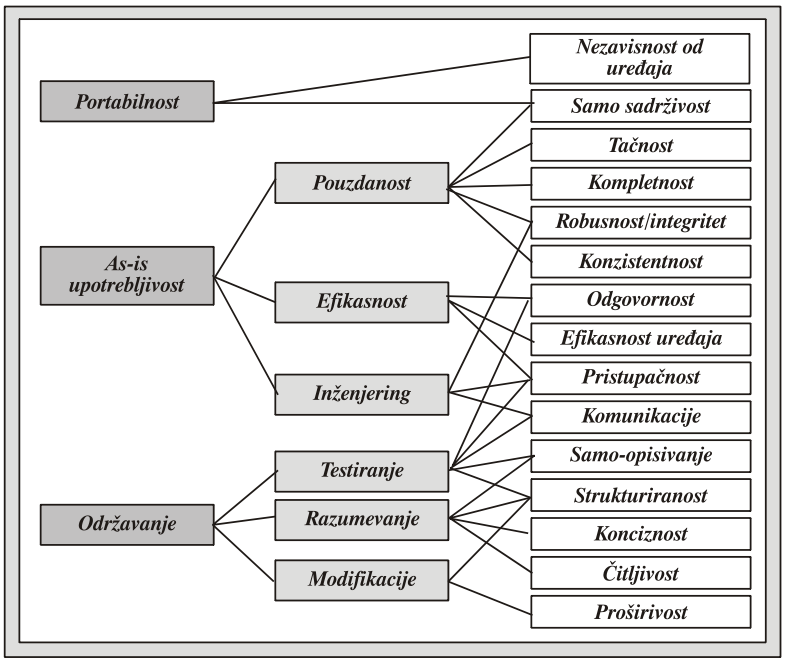
Sa slike 2 se vidi složenost i multidimenzionalnost konteksta kvaliteta softvera. Svaki od slojeva prikazanih na slici poseduje sopstvene karakteistike koje ga definišu.




***Slika 3– Koncept kvaliteta softvera***

Ukoliko se posmatra u užem smislu softver je "alat" koji se koristi unutar nekog sistema, ali sa druge strane softver značajno utiče na funkcionisanje tog sistema. Pošto softver može značajno uticati na rad i fukcionisanje poslovnog ili nekog drugog sistema, potrebno je da softver zadovoljava mnogobrojne atribute kvaliteta i da bude kvalitetan u celini. Softver mora da zadovolji, na prvom mestu, zahteve korisnika, i da poseduje određen nivo kvaliteta, a ne samo funkcionalnost. Na primer kvalitet podataka se može specificirati kroz [5]: tačnost (mera odsutpanja neke vrednosti *v* od neke druge vrednosti v' koja se smatra tačnom); komletnost (stepen do koga je specifična vrednost uključena u kolekciju podataka), konzistencija (koja se definiše na nivou prezentacije, vrednosti i izgleda), interpretacija (format u kome je podatak specificiran i jasno definisan), pouzdanost (koja se definiše kroz pouzdanost podatka i pozdanosti izvora), pravovremenost (mera koliko je podatak pravovremen i odgovarajući za neku operaciju). Kao što je potrebno meriti i vrednovati ove karakteristike da bi se definisao kvalitet podataka potrebno je meriti i mnoge druge karakteristike koje nam omogućavaju utvđivanje kvaliteta softvera, pri čemu metrika softvera ima značajnu ulogu.
Zato se moraju definisati karakteristike, koje često nisu tako eksplicitne, ali zančajno utiču na performanse i kvalitet sofvera. Postoje brojni, razlilčiti pristupi u definisanju seta karakteristika koje je potrebno meriti da bi se utvrdio kvalitet softvera, odnosno definisao modela kvalitea, počev od klasika Boehem [6], McCall [7], pa do novijih pristupa ISO 9126 [8] i JUS/ISO 12207 - Informaciona tehnologija - procesi životnog ciklusa softvera.
Slika 3 pokazuje koji aspekte kvaliteta softvera koji obuhvataju korišćenje različitih koncepata, planiranje, aktivnosti, tehnike i merenje koje utiču na obezebeđenje kaliteta softvera. Sa druge strane postoje modeli koji opisuju kvalitet softvera.
Boehem [6] je predložio model kvaliteta prikazan na slici 4, koji nastoji da uključi opšte karakteristike kvaliteta u kriterijume kvaliteta kroz tri ugla posmatranja, prema vrsti i položaju korisnika:

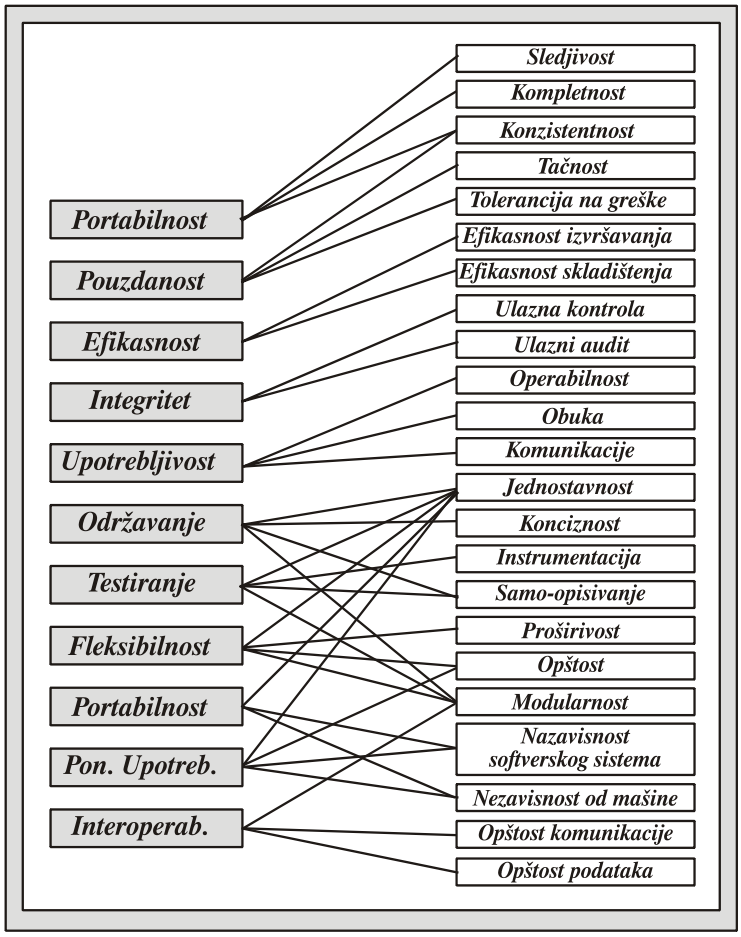
- Krajnji korisnici (as-is upotrebljivost)
- Potencijalnih korisnika, na drugim mestima (portabilnost)
- Potencijalnih korisnika, u nekom drugom vremenu (mogućnost održavanja).




***Slika 4 - Boehm-ov model kvaliteta softvera***

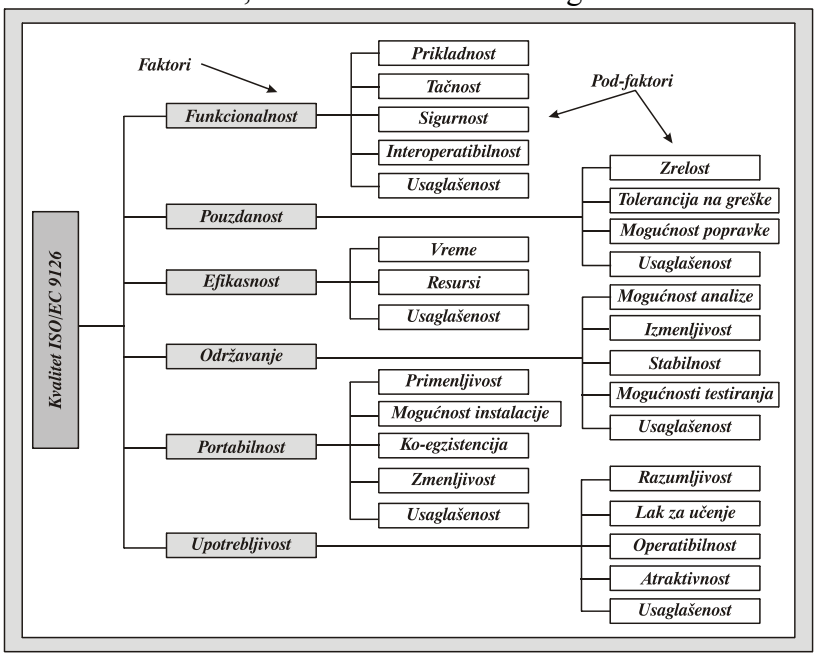
McCall je 1970 [7] definisao "okvir kvaliteta" sa kriterijumima za kvalitet koji se oslanjaju na tri tačke (slika 5):
· Operacija (korišćenje).

· Revizija (menjanje).
· Tranzicija (prenošenje).



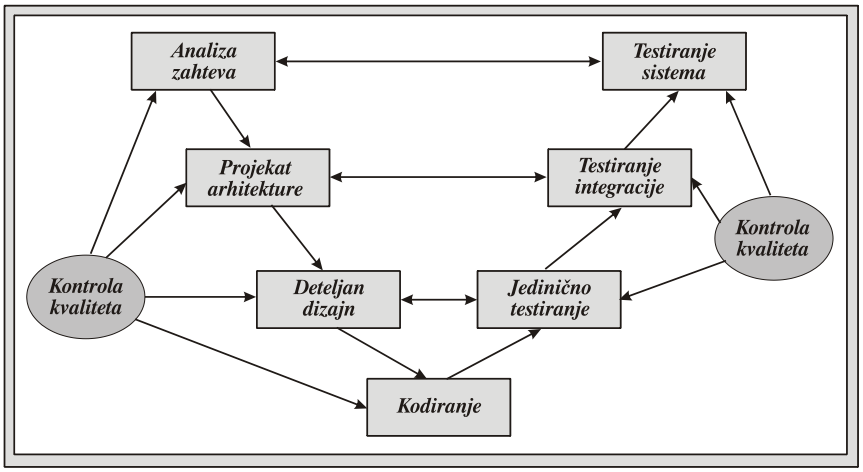

***Slika 5 - McCall -ov model kvaliteta softvera***

Dalji razvoj modela kvlaitea je doveo do standardizacije predložene ako ISO 9126 [8]. Ovaj model nudi šemu u dva nivoa sa karakteristikama i podkarakteristikama (slika 6). Jedna od ključnih razlika je što ovaj model kvaliteta dopušta da svaka pod karakteristika utiče samo na jednu karakteristiku na višem nivou, za razliku od većine drugih modela.



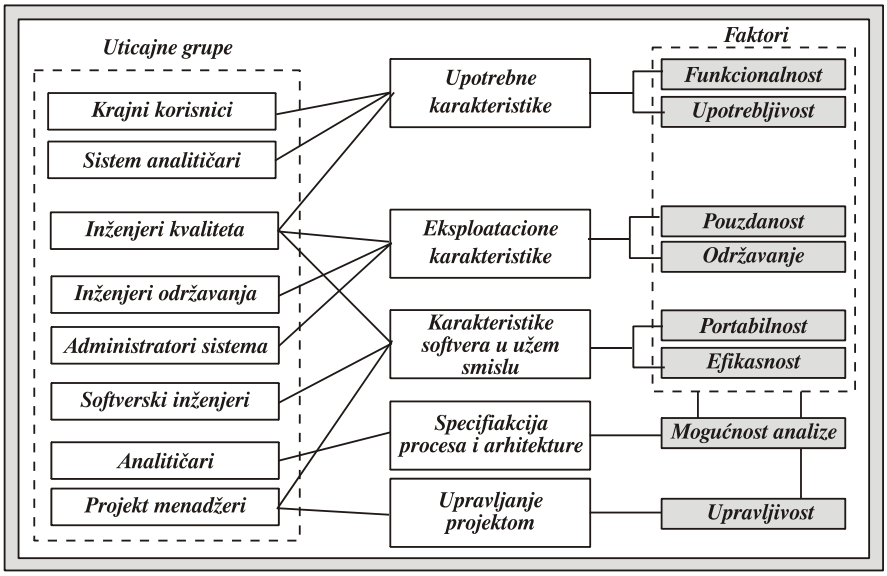

***Slika 6 - Model ISO/EC 9126 kvaliteta softvera***

Stnadard JUS/ISO 12207 - Informaciona tehnologija - procesi životnog ciklusa softvera - predstavlja dalji rad na definisanju na unapredjenju ovog modela.
Na slici 7 su prikazani osnovni elementi V modela kvaliteta softvera [9]. Pravougaoni elementi na slici predstavljaju tradicionalne faze u razvoju sofvera (zahtevi i projektovanje sa leve strane i testiranje sa desne strane). Naročito treba obratiti pažnju na veze između leve i desne strane koje su prikazane strelicama koji na primer povezuju analizu zahteva i testiranje sistema. Strelica koja pokazuje na levo označava dinamičko testiranje softvera. Strelica koja ukazuje na desno označava planiranje koje se odvija uporedo sa fazom razvijanja softvera. Kontrola kvaliteta označava da je u svakoj fazi životnog ciklusa razvoja softvera prisutna kontrola.




**Slika 7 – V- Model kvaliteta softvera**

Polazeći od ovih modela, a naročito modela ISO/EC 9126 i JUS/ISO 12207 Centar za kvalitet Mašinskog fakulteta u Kragujevcu je razvio svoj modifikovani model za obezbeđenje kvaliteta softvera (slika 8).




**Slika 8 –Model obezbeđenja kvaliteta softvera - CIM**

Ovaj model pokušava posmatra proces razvoja softvera kroz tri koraka. Prvi korak je klasičan razvoji softvera, odnosno pisanje programskog koda iz koga treba da proisteknu neke karakteristike softvera u užem smislu (kvalitet kodiranja, veličina razvijenog koda i sl.) Sledeći nivo je samo ponašanje softverskog proizvoda u procesu eksploatacije, tj. pouzdanost i održavanje i poslednji nivo su upotrebne karakteristike softvera, tj. one vrednosti koje softver ima za krajnjeg korisnika. Ovi definisani nivo imaju jedan opštiji značaj i ovaj model se može upotrebiti i za obezbeđenje kvlateta i drugih proizvoda, a ne samo softverskih, pri čemu bi se modifikovali izvesni faktori ili podfaktori.

Takođe ovaj model je proširen sa aktivnostima i karakteristikama koje nosi specifiakcija procesa i upravljanje projektnim aktivnostima.
I na samom kraju model uvodi i uticajne grupe, stejkholdere, odnosno one koji ocenjuju određene karakteristike i faktore, tj. one na koje ti faktori ponajviše utiču.
Ovako modifikovan pristup je omogućio osnovu za razvijanje dokumentacije sistema kvaliteta koja treba da obezbedi kvalitet softvera. Definisani su svi kroraci, potrebni dokumenti, standardi i testiranja kao i odgovornosti za sprovođenje pojedinih aktivnosti.

## 4. Formiranje dokumentacije sistema kvalitea za obezbeđenje kvaliteta softvera

Jedan od mogućih pristupa u obezbeđenju kvalita softvera je i definisanje potrebne dokumentacije sistema kvaliteta koja bi se odnosila na softver. Ova dokumentacija bi trebalo da obezbedi da softverski proizvod koji se isporučuje kupcu ili koristi u preduzeću zadovoljava ustanovljene i ugovorene tehničke zahteve. Ova dokumentacija bi takođe morala da obezbedi prijem softverskog projekta i do obezbedi da se sve politike, standardi, iskustva iz prakse, procedure i procesi primenjuju u toku trajanja projekta po unapred utvrđenom redosledu.
Prema preporuci Centra za kvalitet, Mašinskog fakulteta u Kragujevcu procedura za obezbeđenje kvaliteta softvera trebala bi da sadrži korake i akcije koje su prikazani na slici 9.
Neki od relevantnih IEEE standarda su korišćeni da bi se „pokrila“ sva značajna mesta u toku životnog ciklusa razvoja proizvoda i da bi se učinilo da preporučena dokumentacija za obezbeživanje kvaliteta softvera bude što potpunja:

- IEEE 730, Standard za planove obezbeđenja kvaliteta softvera, propisuje minimum od:
  - Specifikacija zahteve softvera - Software Requirements Specification (SRS)
  - Opis projekta softvera -Software Design Description (SDD)
  - Plan softverske verifikacije i validacije - Software Verification and Validation Plan (SVVP)
  - Izveštaj o softverskoj verifikaciji i validaciji - Software Verification and Validation Report (SVVR)
  - Korisnička dokumentacija - User Documentation
  - Plan upravljanja softverskom konfiguracijom -Software Configuration Management Plan (SCMP)
- IEEE 828, Standard za planove upravljanja dokumentacijom
- IEEE 829, Standard za dokumentaciju za softversko testiranje
- IEEE 830, Preporučena praksa za zahteve softverske specifikacije
- IEEE 1008, Standard za jedinično (unit) testiranje softvera
- IEEE 1012, Standard za planove softverske verifikacije i validacije
- IEEE 1016, Vodič za opis softverskog projekta
- IEEE 1028, Standard za softverske kontroel i audite
- IEEE 1042, Vodič za planove upravljanja softverskim konfiguracijama
- IEEE 1044, Standard klasifiakcije softverskih anomalija
- IEEE 1045, Standard za metriku softverske produktivnosti
- IEEE 1058.1, Standard za planove upravljanje softverskim projektima
- IEEE 1059, Vodič za planove softverske verifiakcije i validacije
- IEEE 1061, Standardi za metriku metodologije kvaliteta softvera
- IEEE 1063, Standard za korisničku dokumentaicju softvera
- IEEE 1074, Standard za razvoj SDLC procesa
- IEEE 1219, Standard za softversko održavanje
- IEEE 1233, Vodič za razvoj specifikacije zahteva sistema

| | | |
|---|---|---|
| | **Obezbeđenje kvaliteta softvera** | ***QS.01.01*** |

| | | | |
|---|---|---|---|
| **Standard** | | **tačka** | |

## Sadržaj:



| Kopija: _____ | Izradio | Kontrolisao | Odobrio |
|---|---|---|---|
| Ime i prezime | | | |
| Potpis | | | |
| Broj strana: | Putanja: | Izdanje: prvo | Datum: |

**Lista izmena dokumenta**

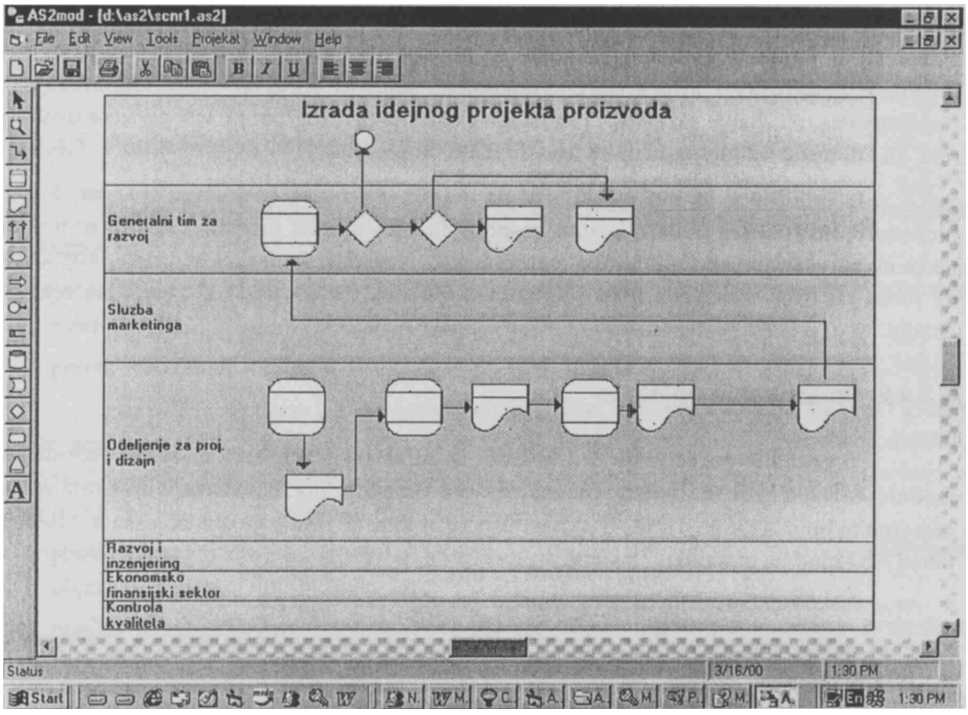
| Broj izmene | Datum izmene | Promenjene strane | Broj izmene | Datum izmene | Promenjene strane |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |

***Slika 9 – Procedura za Obezbeđenje kvaliteta softvera***

## 5. CASE alat za upravljanje dokumentacijom za obezbeđenje kavaliteta softvera

Softverski alat razvijen u CIM centru Mašinskog fakulteta u Kragujevcu namenjen je softverskim inženjerima i inženjerima sistema kvaliteta koji se bave analizom i ispitivanjem procesa razvoja softverskog proizvoda (ili bilo kog drugog proizvoda). Osnovna zamisao vezana za ovaj softver je da inženjer na jednostavan i intuitivan način za veoma kratko vreme može da putem veoma upotrebljivog grafičkog interfejsa (slika 10) prikaže proces razvoja novog softverskog proizvoda, analizira ga i u potpunosti dokumentuje. U slučaju razvoja softverskog proizvoda može se koristiti prethodno definisani predlog potrebne softverske dokumentacije definisane u okviru prezentovane procedure. U slučaju kompleksnih rešenja, data procedura se može podeliti u više međusobno povezanih procedura i instrukcija. Naravno, sve te procedure prate odgovarajući obrasci i zapisi. Notacija korišćenja u ovom softveru se bazira na AS2 modelu [12], pri čemu osnovna dokumentacija koja se direktno vezuje za aktivnost (aktivnost/proceduru) može biti: procedura, instrukcija ili uputstvo, što direktno zavisi od kompleksnosti aktivnosti i organizacionog nivoa na kome se aktivnosti sprovode i definišu. Ove entitete možemo nazvati „postupci“. Iz jednog postupka može da proistekne jedan ili više obrazaca. Postupci i obrasci su delovi jednog scenarija. Kako jedan scenario može da posluži kao osnova za razvoj više konkretnih proizvoda, tako i jedan obrazac može da ima više svojih pojavnih vidova kroz postojanje zapisa. Za sprovođenje svake procedure, instrukcije i uputstva definiše se odgovorna funkcija i odgovorna osoba, što na neki način korespondira sa ulogama uticajnih grupa koje su predstavljene na modelu obezbeđenja kvaliteta softvera.

Sam softver sastoji od nekoliko podcelina i svaka od njih je zadužena za određenu vrstu transakcija. Tako postoji podcelina za izradu postavke problema, podcelina vezana za dokumentovanje aktivnosti (u kojoj se na nekoliko standardnih *Windows* načina mogu unositi podaci o instrukcijama ili procedurama koje opisuju pojedine aktivnosti), podcelina za podešavanje načina funkcionisanja programa (*Options*), podceline za izradu elektronskih obrazaca, podceline za omogućavanje pristupa Interentu i Intranetu.




***Slika 10 – Softver za upravljanje dokumentacijom sistema kvaliteta za***

Sam softver u okviru svog kontrolnog centra omogućava podelu procesa na faze, definisanje funkcija koje učestvuju u procesu, kao i raspodelu funkcija po pojedinim fazama sa aspekta

prikazivanja i analizre procesa razvoja novog softverskog proizvoida. Samo okruženje ovog softverskog alata je standarno *Windows* okruženje pa je vrlo lako za korišćenje.
U okviru glavne forme u prvoj fazi (slika 10) moguće je postavljati objekte (procedure, instrukcije, zapise) i povezivati objekte prema prethodno utrđenim pravilima primenjene AS2 notacije.
U drugom koraku moguće je definisati opšte podatke o objektima (oznake, nazive ...), definisati izdanja, istorijat statusa i dokumentovati sve aktivnosti. Takođe je omogućeno pravljenje izmena na svim postupcima i dokumentovanje tih izmena.
Softver omogućava i kreiranje i pregled obrazaca, kao i potpunu manipulaciju obrascima (izmene, štampa...).
Razvijeni softver može poslužiti kao vrlo koristan alat namenjen inženjerima sistema kvaliteta i može pružiti značajnu podršku procesu obezebeđenja kvaliteta softvera.

## Zaključak

Na osnovu svega navedenog mogu se izvući sledeći zaključci:

1. Kvalitet softvera je više dimenzionalni koncept koji se ne može jednostavno definisati. Za ostvarivanje potrebnog kvalitea softvera od velikog je značaja određivanje modela kvaliteta softvera i definisanje dokumentacije sistema kvaliteta.
2. CIM centar Mašinskog fakulteta u Kragujevcu je modifikovao model kvlaiteta softvera uvodeći karakteristike pojedinih faza razvoja softvera, kao i podršku projekt menadžmenta. Sem toga u navedeni model su unete i interesne grupe, odnosno svi koji koji utiču na ostvarivanje određenih faktora ili karakteristike, tj. svi oni za koje pojedine grupe faktora imaju dominantni značaj. Na ovaj način stvorena je osnova za definisanje kvalitetne dokumentacije sistema kvaliteta koja treba da omogući obezbeđenja kvaliteta softvera.
3. Razvijen je predlog dokumentacije sistema kvaliteta za podršku obezbeđenju kvaliteta softvera. Sama dokumentacija se oslanja na relevantne ISO i IEEE standarde, odnosno ugradila je sve preporuke tih standarda u predlog sopstvene dokumentacije.Dokumentacija sadrži u sebi sledeće tačke: 1. Svrha; 2. Odgovornost za primenu; 3. Referentna dokumentacija, definicije i skraćenice; 4. Upravljanje projektom; 5. Dokumenti; 6. Standardi; 7. Kontrole i auditi; 8. Testiranje; 9. Izveštavanje o problemima i korektivne akcije; 10. Alati, tehnike i metodologijie; 11. Obuka; 12. Upravljanje rizikom
4. U okviru CIM centra Mašinskog fakulteta u Kragujevcu razvojen je CASE alat koji omogućava upravljanje dokumentacijom sistema kvaliteta. Ovaj alat se može sasvim uspešno primeniti na upravaljanje dokumentacijom sistema kvaliteta koja se tiče obezbeđenja kvaliteta softvera.

**Zahvalnost**

Ovaj rad je nastao kao rezultat istraživanja na projektu TR-6218A: "Razvoj softverskih rešenja u Internet/Intranet okruženju za integrisani razvoj proizvoda i procesa", koga finansira Ministarstvo nauke i zaštite životne sredine Republike Srbije.

## Literatura

1. Hilburn T., and Towhidnejad M., "Software Quality Across The Curriculum", 32nd, ASEE/IEEE Frontiers in Education Conference, 2002 IEEE, November 6 - 9, 2002, Boston, MA
2. Rothman, Johanna, "What Does It cost to Fix a Defect?", Column Archive, StickyMinds.Com, http://www.stickyminds.com/.
3. The Standish Group "The Chaose Report", http://www.standishgroup.com/sample_research/chaos_1994_1.php

4. Unhelkar B., "Applying the UML to Enhance the Quality of Web Service", http:///www.methodscience.com
5. Cappiello C., Francalanci C., Pernici B., Plebani P., Scannapieco M., "Data Quality Assurance in Cooperative Information Systems: a Multi-dimension Quality Certificate", IASI-CNR, Rome, Italy, 2002
6. McCall J. A. "Factors in Software Quality", General Electrics, June 1997
7. Boehm B. W. et. All. "Characteristics of Software Quality", TRW serise on Software Technologies, Vol. 1, North Holland, 1978.
8. Kan S. H. "Metrics and Models in Software Quality Engineering", Addison – Wesley Publishing, Co. 1995.
9. Wallace D., Reeker L., "Software Quality", IEEE Version 0.95. May 2001.
10. Cappiello C., Francalanci C., Pernici B., Plebani P., Scannapieco M., "Data Quality Assurance in Cooperative Information Systems: a Multi-dimension Quality Certificate", IASI-CNR, Rome, Italy, 2002
11. Arsovski Z., "Inforamcioni sistemi", CIM centar, 2001
12. Somervile J., "Software Engineering", Addison-Wesley, Fifth-edition, 2001