



## KVALITET I TESTIRANJE SOFTVERA\*

## SOFTWARE QUALITY AND TESTING\*

mr Miladin Stefanović<sup>1</sup>, mr Milan Erić<sup>2</sup>, mr Slobodan Mitrović<sup>3</sup>

**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. Sa druge strane potrebno je definisati odgovarajuću metriku i izvršiti testiranje softvera da bi se izabrane karakteristike na neki način kvantifikovale. Tema ovog rada kvalitet i testiranje web servisa.*

**Ključne reči:** *kvalitet, testiranje, softver, web servisi*

**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. On the other hand software metrics and testing are important for quantification of defined parameters. In this paper we will present quality and testing of web services.*

**Key words:** *Quality Assurance, Software*

### 1. UVOD

Obezbeđenje kvaliteta (Quality Assurance) predstavlja jednu od vodećih paradigmi modernog poslovanja, sa druge strane imamo informacione tehnologije kao strateški alat unapređivanja poslovanja i svakog procesa. Kvalitet softvera 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 strategije i pristupe u validacionom testiranju. Sve ovo je potrebno da bi dobili softverski proizvod koji

---

1) Mr Miladin Stefanović, CIM centar, Mašinski fakultet Kragujevac

2) Mr Milan Erić, Mašinski fakultet Kragujevac

3) Mr Slobodan Mitrović, Mašinski fakultet Kragujevac

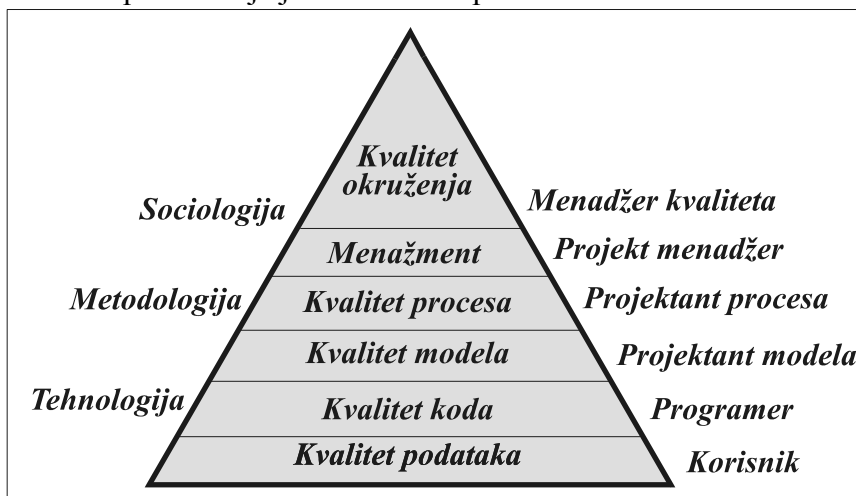
\* Rezultati deo projekta TR-6218A - "Razvoj softverskih rešenja u Internet-intranet okruženju za integrisani razvoj proizvoda i proces", Ugovor sa Ministarstvom za nauku i ekologiju

zadovoljava zahteve kupaca, koji je razvijen u skladu sa specifikacijom i koji ne sadrži greške. Svi ovi problemi se usložnjavaju kada je reč o web servisima koji su zasnovani na XML otvorenim standardima (SOAP, WSDL, UDDI) i imaju svojstvo interoperabilnosti preko različitih platformi, a pri tome su slabo povezani. Sa druge strane, veliki broj složenih servisa je sastavljen od jediničnih, baznih servisa, koje isporučuju različite komponente i zbog toga se kvaliteti i performanse servisa stavljaju u prvi plan. U okviru CIM centra Mašinskog fakulteta u Kragujevcu razvijen je softverski alat Q-WS 1.0 koji omogućava testiranje nekih atributa web servisa. U okviru ovog rada biće prezentovane osnove kvaliteta softvera, metrike i testiranja softvera i web servisa i biće predstavljen originalno razvijeni CASE alat.

## 2. 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 [1]:

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



*Slika 1 – Kontekst kvaliteta softvera [2]*

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

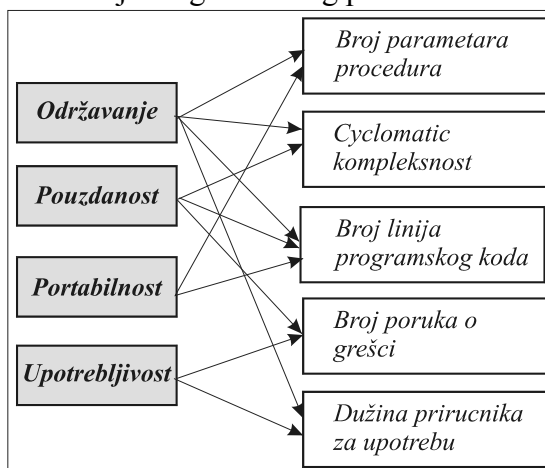
- Obezbeđivanju kvaliteta, koji obezbeđuje da organizacija prati ciljeve kvaliteta, što uključuje definisanje i izbor različitih standarda koji se uključuju u proces razvoja softvera ili sam softverski proizvod. Izabrani standardi mogu biti ISO 9000 ili JUS/ISO 12207 - Informaciona tehnologija - procesi 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 kvalitetno 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 1 se vidi složenost i multidimenzionalnost koncepta kvaliteta softvera. Svaki od slojeva prikazanih na slici poseduje sopstvene karakteristike koje ga definišu. Na primer kvalitet podataka se može specificirati kroz [2]: **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.

### 3. METRIKA I TESTIRANJE SOFTVERA

Metrika softvera je onaj tip merenja koji se odnosi na softverski sistem, proces ili pripadajuću dokumentaciju. Sama metrika se može podeliti na metriku kontrole i metriku predviđanja. Metrika kontrole se koristi od strane menadžmenta u cilju kontrole procesa vezanih za softvera. Primeri ove metrike su potreban rad i korišćenje prostora na disku. Metrika kontrole nam može dati informacije o kvalitetu procesa i kvalitetu samog proizvoda. Metrika predviđanja meri attribute proizvoda koji mogu biti korišćeni da predvide pripadajući kvalitet softvera. Na primer, čitljivost prateće softverske dokumentacije se može proceniti na osnovu Fog indeksa, ili jednostavnost održavanja softvera se može predvideti preko *cyclomatic* kompleksnosti (McNabe, 1976). Znači merenjem jednog internog atributa kao što je *cyclomatic* kompleksnost dobijamo vrednost jednog eksternog parametra kao što je održavanje.



Slika 2 – Veza između internih i eksternih akarakteristika

Pri merenju internih karakteristike prema [1], mora se voditi računa o tome da:

- Interni atributi moraju biti tačno izmereni.
- Mora postojati zavisnost između onoga što merimo i eksternih ponašanja atributa.
- Relacija se mora razumeti, validirati i iskazati preko formule ili modela.

Mnogi autori, (Sommerville, Kitchenham) ističu da je poslednji uslov najbitniji, kao i to da se on često zanemaruje. Joše jedan problem koji se javlja pri ovom merenju je suštinsko razumevanje šta izmereni podaci u biti znače.

Pri definisanju metrike softvera obično se srećemo sa:

1. Metrikom kvaliteta dizajna koja obuhvata: koheziju, povezanost, razumljivost, adaptibilnost i kompleksnost. Gde se kompleksnost može meriti kao  $Kompleksnost = Duzina(Fan - in \times Fan - out)^2$ , gde je *Fan-in* broj linija koje ulaze u komponentu, a *Fan-out*, broj linija koje izlaze iz komponente, a dužina je predstavljena kao broj linija koda.
2. Metrika kvaliteta programa koja obuhvata: dužinu koda, *cyclomatic* kompleksnost, dužinu identifikatora, dubinu uslovnog ugnježdavanja.
3. Metrika kvaliteta dokumentacije koja obuhvata najpoznatiji Fog indeks, koji predstavlja odnos dužine teksta i kompleksnih izraza i ukazuje na „čitljivost“ teksta.

Testiranje softvera je složen posao koji obuhvata testiranje jedinca, modula, pod-sistema, sistema i prijemno testiranje. Pri ovome postoji veliki broj strategija testiranja kao na primer: **top-down** strategija (gde se prvo testiraju viši nivoi sistemi pa se ide ka nižim), **bottom-up** testiranje (obrnuto u odnosu na prethodnu strategiju), **thread** testiranje (testiranje bazirano na događajima koji izazivaju akciju sistema), **stress** testiranje (testiranje sa specifičnim inputima) i **back-to-back** testiranje (kada se dve verzije sistema testiraju istim podacim, a kasnije se rezultati upoređuju).

U samom procesu testiranja naročito je bitno ono testiranje koje obezbeđuje da krajnji proizvod, tj. softver nema grešaka. Mayers (1979) je definisao kao dobar test onaj koji otkriva prisustvo greške u softveru koji se testira.

Pri ovom pristupu, validacionog testiranja, koji je od suštinskog značaja za kvalitet softvera potrebno je izvršiti:

- **Funkcionalno** ili **black-box** testiranje, gde su testovi definisani na osnovu specifikacije softvera. Pri ovome se vodi računa da se sistem testira kvalitetnim inputima i inputima koji su dobro raspoređeni. Tj. da se pri testiranju koriste i ispravni i neispravni, karakteristični i najčešće očekivani inputi.
- **Strukturno** ili **white-box** testiranje, gde se testovi formiraju na osnovu strukture i implementacije softvera. Pri čemu se često koristi strategije testiranje putanji, pri čemu se testiraju sve moguće putanje realizacije jedne komponente. Pri čemu se definiše *cyclomatic* kompleksnost kao  $CC = Broj(ivica) - Broj(cvorova) + 1$  (gde se broj čvorova i ivica dobija iz grafa raspoloživih putanja).
- **Testiranje interfejsa**, gde se testovi izvode iz specifikacije programa i znanja o interfejsima. Ovo testiranje je naročito važno za objektno-orijentisane sisteme.

#### 4. KVALITET I TESTIRANJE WEB SERVISA

Web servis model je zasnovan na XML otvorenim standardima (SOAP, WSDL, UDDI) i ima svojstvo interoperabilnosti preko različitih platformi. Web servisi su podržani od strane svih velikih kompanija kao jedan od strateških pravaca razvoja softvera.

Kod servisne orijentisane arhitekture:

- funkcije su definisane kao servisi; ovo uključuje poslovne funkcije, poslovne transakcije sastavljene od nižih funkcija i systemske servisne funkcije.
- svi servisi su nezavisni; svi servisi funkcionišu na principu "crne kutije".

- za servise je irelevantno da li su lokalni ili udaljeni (van matičnog sistema); one mogu biti deo aplikacije, ali se mogu pozivati preko Interneta i biti deo arhitekture poslovnog partnera.

**Tabela 1- Paramteri bitni za testiranje i kvalitet web servisa [3]**

<b>Atribut</b>	<b>Značenje</b>
Dostupnost	<i>Predstavlja procenat vremena u kome je servis dostupan.</i>
Pouzdanost	<i>Pouzdanost uspostavlja neke merljive veličine koje servis treba da poseduje.</i>
Integritet	<i>Uključuje integritet servisa i integritet podatka/poruke za vreme transakcije servisa.</i>
Skalabilnost	<i>Odnosi se na sposobnost servisa za povećanjem kapaciteta funkcionisanja kako se povećava zahtev za servisom.</i>
Performanse	<i>Ova osobina uključuje merenje brojčanih karakteristika kao što su prosečno vreme odgovora, kašnjenje mreže i sistema, maksimalna i minimalna opterećenja i sl.</i>
Vreme odziva	<i>Predstavlja vreme koje je potrebno servisu da odgovori na različite zahteve. Vreme odziva je funkcija vremena od intenziteta ulaza koji može biti meren kao nivo pristiglih zahteva za servisom.</i>
Protok	<i>Predstavlja nivo do koga servis može da procesira zahteve.</i>
Sigurnost	<i>Ova osobina zahteva postojanje neke vrste provere identiteta korisnika servisa kao i mogućnost da sistem odbije pružanje servisa.</i>
Autentizacija	<i>Provera identiteta i verifikacija korisnika, kao i provera podatka/poruke/servisa od strane servis provajdera.</i>
Privatnost	<i>Odnosi se na provatnost koju web servis ili web provajder garantuje klijentima, uključujući privatnost korisnika, privatnost podataka i servisa.</i>

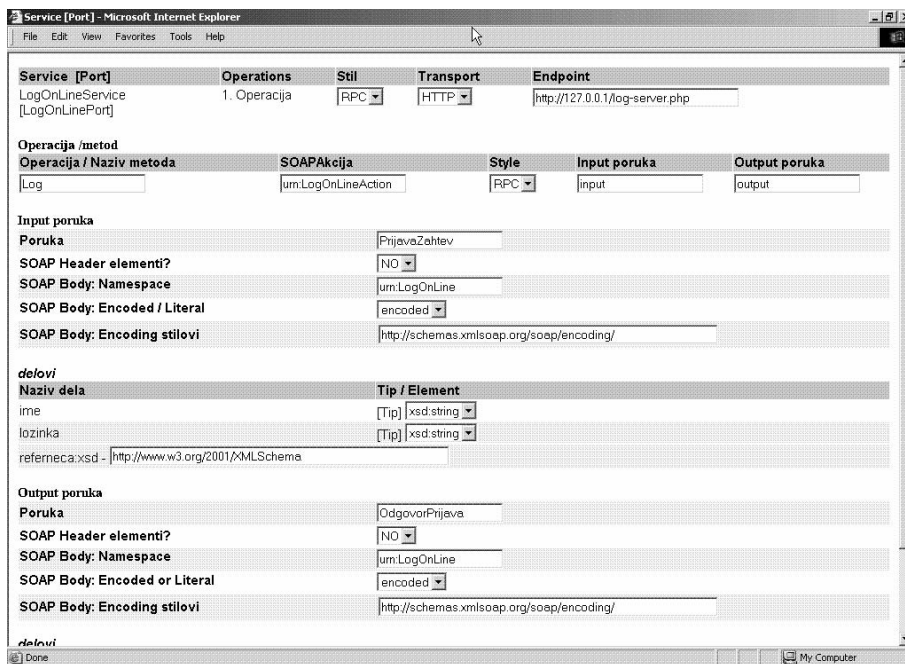
Zbog navedenih karakterisitka testiranje kvaliteta i funkcionalnosti web servisa ima svoje specifičnosti u poređenju sa drugim softverom. Testiranje web servisa može se svesti na testiranje svakog pojedinačnog servisa, testiranje integracije i testiranje celog sistema. Postoji veliki broj parametara kojie je potrebn testirati i pratiti u svrhu ocenjivanja kvaliteta web servisa (tabela 1). Dobra polazna osnova je da se u konceptu korišćenja web servisa svaka akcija na krajnjim tačkama upisuje u dnevnik aktivnosti, na taj način analziom tih zapisa može se dobiti slika o aktivnosti servisa.

Po perthodno navedenom, kvalitet web servisa se može odnositi na kvalitet specifikacije, kvalitet modeliranja arhitekture sistema i kvalitet korišćenje i registracije i pronalaženja servisa.

## **5. CASE ALAT ZA TESTIRANJE WEB SERVISA**

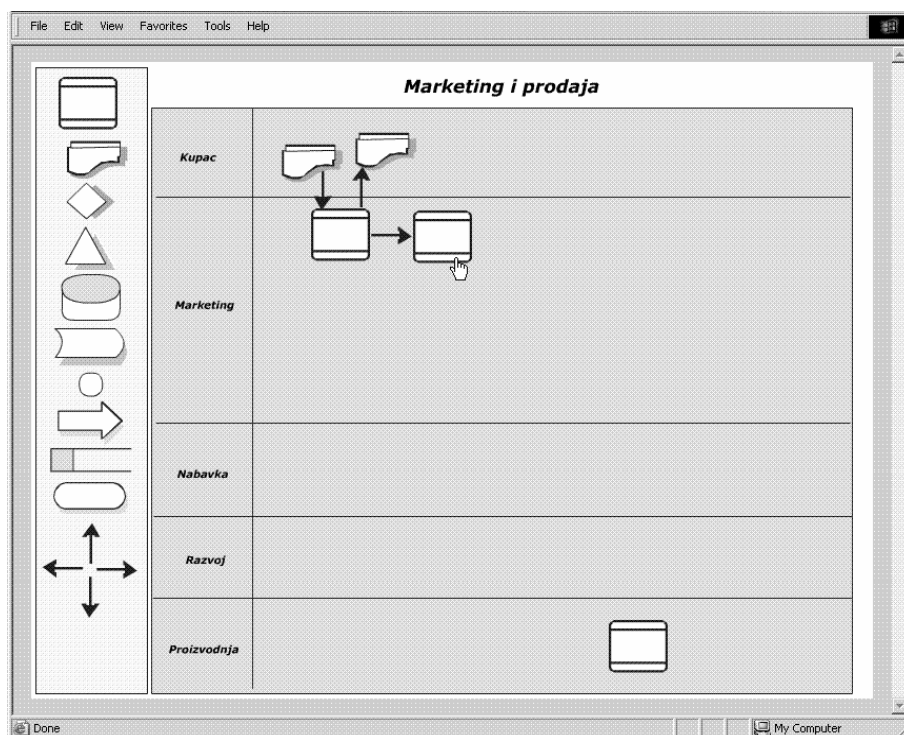
Pri svakom testiranju i određivanju atributa kvaliteta nekog softvera koriste se različiti alati. Svima njima je zajednička potreba da ispunjavaju neke od sledećih karakteristika:

1. Statička analiza proizvoda. Podrazumeva statičku analizu softvera i merenje različitih parametara. Alat može analizirati izvorni kod, dokumentaciju.
2. Dinamička analiza proizvoda. Beleži informacije o funkcionisanju softvera i mogućim greškama i otkazima.
3. Analiza procesnih podataka. Daje podatke menadžmentu o upotrebljivosti i funkcionalnosti softvera.



**Slika 3 – Specifikacija web servisa**

Postoje različite metode specifikacije i kreiranja web servisa. Metoda Q-WS koja je razvijena u CIM centru Mašinskog fakulteta u Kragujevcu bazira se na specifiaciji web servisa korišćenjem dokumentacije sistema kvaliteta. Kreiranje web servisa se odvija kroz sledeće korake: kreiranje matrice dokumentacije sistema kvaliteta, kreiranje matrice komunikacije, kreiranje matrice podataka i poslednji korak je implementacija i kreiranje sumarnog opisa web servisa.



**Slika 4 – Mapa web servisa, test i dnevnik testiranja**

Na osnovu ove metode kreiran je alat Q-WS 1.0 CASE tool koji omogućava specifikaciju web servisa kao i testiranje web servisa. Kao rezultat procesa definisanja dobija se kompletna specifiakcija web servisa koja je prikazana na slici 3.

Sama specifikacija web servisa nam omogućava kreiranje test fajla koji će se koristiti za testiranje web servisa. Naime, pošto su u prethodnom koraku definisane poruke i promenljive koje se razmenjuju u tim porukama, moguće je sastaviti test fajl koji će sadržati potrebne parametre i omogućiti black-box, funkcionalno testiranje web servisa.

Na slici 4 je prikazano postavljenje test fajla koji je prikazan sa strelicom usmerenom ka web servisu i dnevnik testiranja koji je pokazan kao dokument ka kome ide strelica od web servisa. Takođe je mogući izvršiti testiranje interfejsa, odnosno testiranje veze između dva web servisa koji funkcionišu u okviru jedne logičke celine.

U datom primeru testiran je web servis koji služi za logovanje korisnika na sistem za naručivanje. Razmenjena su dva parametra (login i password) i potrebni podaci su upisani u dnevnik testiranja web servisa.

Predloženi dnevnik dešavanja za testiranje web servisa tabela 2.

**Tabela 2 – Dnevnik testiranja web servisa**

Naziv	Obavezan / Neobavezan	Primer
ID poruke	O	2131231
Vreme	O	2004:08:16 22:40:41
Naziv servisa	O	log
Poruka (samo telo poruke)	N	password login
Smer (primanje, slanje poruke od strane servisa)	O	primanje
Prioritet poruke	O	1
Prima	O	log
Šalje	O	user

Takođe se prati:

- **utvrđeni tok** - razmena zahteva i odgovora; pozivanje operacija ili metoda mora sa izvršavati prema prethodno definisanom postupku.
- **slanje poruka** - provera pouzdanosti isporuke poruka; potrebno je ispitati vreme kako poruka stigne od pošiljaoca do odredišta, ukoliko poruka stigne van definisanog okvira smatra se da ona nije isporučena.
- **prioritet poruka** - ukoliko šalju poruke istog prioriteta; odnosno zahtevi istog prioriteta, odgovori moraju biti usklađeni sa vremenom pristizanja zahteva. Prvo se šalje odgovor na prvi pristiglu zahtev, ukoliko stiže zahtev višeg prioriteta potrebno je ispuniti zahtev višeg prioriteta.
- **performanse** - neke od merljivih performansi su broj poruka poslatih iz jedne tačke u jedinici vremena ili broj primljenih poruka u jedinici vremena.
- **pouzdanost** - broj pristiglih poruka u zadatom vremenskom okviru u odnosu na ukupan broj poruka.

## 6. 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 kvaliteta softvera od velikog je značaja određivanje metrike softvera i sprovođenje adekvatnog procesa testiranja. Pri ovome validaciono testiranje predstavlja bitan korak u procesu dobijanja proizvoda koji je usklađen sa specifikacijom i ne sadrži greške.
2. Web servisi kao softverske komponente imaju svojih specifičnosti, ali takođe moraju biti testirani u cilju postizanja adekvatnog nivoa kvaliteta. Postoji veliki broj parametara koje je potrebno testirati kada su u pitanju web servisi.
3. U okviru CIM centra mašinskog fakulteta u Kragujevcu razvojen je CASE alat Q-WS koji omogućava specifikaciju web servisa na osnovu dokumentacije sistema kvaliteta, čime se postiže da svi procesi i dokumentacija koja se razmenjuje bude bazirana na relevantnim standardima i koju sadrži modul za testiranje pojedinih parametara web servisa. Ovaj alat omogućava funkcionalno testiranje jednog seta karakteristika i formiranje dnevnika testiranja.

## LITERATURA

- [1] Sommerville J., "Software Engineering", Addison-Wesley, Fifth-edition, 2001
- [2] 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
- [3] Hu M., "Web Services Composition, Partition, and QoS in Distributed System Integration and Re-engineering", Police Information Technology Organisation, United Kingdom, 2003.
- [4] Unhelkar B., "Applying the UML to Enhance the Quality of Web Service", <http://www.methodscience.com>
- [5] Conti M., Kumar M., "Quality of Service in Web Service", IEEE Computer Magazine, 2001