

EVALUACIJA KVALITETA SOFTVERA PREMA ISO 9126 STANDARDU

EVALUATION OF THE SOFTWARE QUALITY BASED ON THE STANDARD ISO 9126

Saopštenje

DOI 10.7251/POS1208099T	COBISS.BH-ID 3063832	UDK 004.42:006.44
-------------------------	----------------------	-------------------

Mr Boris Todorović¹, Univerzitet za poslovne studije Banja Luka

Sažetak

Razvoj softvera u 21. vijeku postaje glavni fokus mnogih velikih korporacija, što za sebe vuče i čitav niz manjih i srednjih preduzeća. Softver, koji je stvoren da bi riješio neki poslovni problem, još nije toliko zreo koliko je to hardver. Tek krajem devedesetih godina prošlog i početkom 21. vijeka se pojavljuju prvi značajni standardi za razvoj softvera, modeli najbolje prakse i sl. Današnji trend je fokus na kvalitet, odnosno eliminaciju grešaka u projektima razvoja softvera. Projektni menadžment, dakle, ne samo da ima za fokus upravljanje sa tri promjenljive projektnog menadžmenta (troškovima, ciljevima i rokovima), već upravljanje sa kvalitetom softvera. U posljednje dvije decenije razvili su se i standardi za upravljanje kvalitetom u sferi informaciono-komunikacionih tehnologije (IKT), koji definišu konačni cilj, stanje kojem bi trebalo težiti. Cilj ovog rada je predstaviti ISO 9126 standard, koji u sebi nosi model po kojem se vrši evaluacija, tj. procjena kvaliteta softvera. Osnovna hipoteza ovog rada je da korišćenjem ovog standarda možemo kontinuirano kontrolisati, pratiti i korigovati nivo kvaliteta softvera. Eksperimentalno istraživanje je sprovedeno u kompaniji Component Sourcing Group, na projektu razvoja softverskog alata za praćenje inventara putem mobilnih platformi. Rezultati istraživanja podržavaju hipotezu, odnosno pokazuju da je mjerenjem nivoa kvaliteta u ranom periodu projekta moguće identifikovati probleme i izvršiti korektivne akcije, kojima će se optimizovati procesi i podići nivo kvaliteta softvera. Stvaranjem kvalitetnog softvera ćemo smanjiti vrijeme i troškove održavanja, jer će se eliminisati značajne greške (defekti).

Ključne riječi: procjena i evaluacija kvaliteta softvera, mjerenje nivoa kvaliteta, projektni menadžment, ISO 9126, PMBOK, metrike za kvalitet, proces mjerenja kvaliteta softvera

¹ Mr Boris Todorović: Mob. 066/717-323; E-mail: boris@todorovic.biz

Abstract

Software Development in the 21st century becomes the main focus of many large corporations, which in itself draws a range of small and medium enterprises. The software, which was created to solve a business problem, is still not as mature as hardware. Only in the late 90's of the last century and early 21st century appear the first significant standards for software development, models, best practices and the like. Today's trend is a focus on quality and the elimination of errors in the software development project. Project management is not only concerned with planning and controlling the three key variables of project management (costs, objectives and deadlines) but managing the quality of software as well. In the last two decades, the development of quality management standards within the sphere of information and communication technologies came to the fore, thus defining the ultimate objective or situation we should strive for. The aim of this paper is to present the ISO 9126 standard, which carries the model for evaluating ie. assessment of software quality. The main hypothesis of the paper is that, using this standard, we can continuously control, monitor and adjust the level of software quality. The experimental research was conducted at the company Component Sourcing Group, on the project of software development tools for tracking inventory via mobile platforms. The research results support the hypothesis. Namely, the results show that measuring the level of quality in the early period of the project can identify problems and perform corrective action, which will optimize all processes and increase the level of software quality. By creating quality software, we will reduce the time and maintenance costs because it will eliminate significant errors (defects).

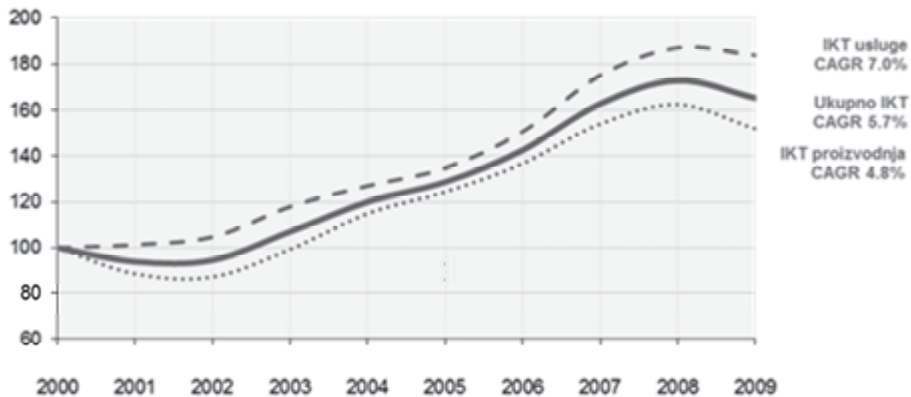
Keywords: *assessment and evaluation of software quality, measuring the level of quality, project management, ISO 9126, PMBOK, quality metrics, process of measuring software quality.*

1. UVOD

Razvoj softvera u 21. vijeku, dakle, postaje glavni fokus mnogih velikih korporacija, što za sebe vuče i čitav niz manjih i srednjih preduzeća. Softver, koji je stvoren da bi riješio neki poslovni problem još nije toliko zreo koliko je to hardver. Tek krajem devedesetih godina prošlog i početkom 21. vijeka se pojavljuju prvi značajni standardi za razvoj softvera, modeli, najbolje prakse i sl. To je jedan od dokaza da je softverska industrija tek započela značajnu standardizaciju, što je samo jedan od prvih koraka ka evoluciji i sazrijevanju.

Razvoj softvera danas je široko rasprostranjen i sve razvijene i privrede u razvoju se fokusiraju na stvaranju što većeg broja preduzeća baziranih na informacionim tehnologijama, koje ne iziskuju velike investicije i kapital, već isključivo znanje. Preduzeća koja se bave razvojem softvera su prisutna u sve većem broju i svakodnevno se otvaraju nova radna mjesta iz IKT industrije, koja bilježi konstantan rast profita i prihoda (grafikon 1). Kako razvoj softvera danas više nije haotičan, već jasno definisan proces, potrebno je primijeniti najbolju praksu, standarde, metode i alate; odnosno, potrebno je posjedovati i primijeniti određenu metodologiju razvoja softvera.

Grafikon 1 - Trend rasta indeksa prihoda najvećih IKT kompanija od 200 - 2009.



Izvor: OECD Information Technology Outlook 2010, www.oecd.org/sti/ito (OECD, 2010)

Fokus na kvalitet, odnosno eliminaciju grešaka u projektima razvoja softvera, dovodi kako do rješavanja neželjenih posljedica poput gubitka vremena i novca (uticaj na finansije preduzeća), a u najgorem slučaju i do ugrožavanja reputacije preduzeća, gubitka značajnih klijenata, sankcija od strane regulatornih organa i sl. Projektni menadžment, dakle, ne samo da ima za fokus upravljanje sa tri promjenljive (troškovima, ciljevima i rokovima), već upravljanje sa kvalitetom softvera.

2. PREGLED TEORIJE KVALITETA SOFTVERA, ISO 9126 STANDARDA I RELEVATNIH MEĐUNARODNIH PUBLIKACIJA

2.1. Kvalitet softvera kao važan faktor u upravljanju projektima

Svi projekti imaju definisane finansijske limite i rokove, tj. iznos definitivnog budžeta za projekat i dan kada sistem mora biti instaliran i potpuno operativan. Ovi limiti nisu nametnuti od strane projektnog tima, već od strane top menadžmenta, klijenata ili zainteresovanih strana i akcionara preduzeća za kojeg je softver i razvoja. Projektni menadžer mora da, pored poznavanja informacionih tehnologija i metoda razvoja softvera, radi i sa ovim netehničkim, tj. finansijskim aspektima projektima. Radni okvir za ovo je jasno definisan u PMBOK metodologiji², gdje se pojavljuje koncept *trougla projektnog menadžmenta*. Ovaj koncept podrazumijeva evaluaciju uspjeha projekta kroz tri varijable: **ciljeve** (engl. Scope), **troškovi** (engl. Cost) i **vremenski okvir** (engl. Time).

Varijabla *vremenski okvir* predstavlja vremenski raspored ili rokove koji su definisani za projekat, a u okviru kojih se projekat mora završiti. *Troškovi* predstavljaju budžet ili finansijske resurse koji su dodijeljeni za potrebe ljudskih resursa, hardver, softver i ostale implicirane troškove poput režija i troškova konsaltinga. Treća varijabla, *obim*, predstavlja ciljeve koji se moraju realizovati, da bi se smatralo da je projekat proizveo funkcionalan softver. Da bi projekat uvijek bio na pravom putu, projektni menadžeri moraju kontinuirano pratiti ove tri varijable i vršiti njihovu evaluaciju, usput modifikujući planove, raspoređujući resurse i aktivnosti projektnog tima.

Kako tvrde Schwalbe (2006, str. 12) i Haugan (2011, str. 89), tri varijable projektnog menadžmenta su zapravo međusobno povezane. Na primjer, povećavajući obim projekta izvršiće se direktni uticaj na vrijeme i resurse koji su potrebni da bi se završio dodatni

² PMBOK – Project Management Body of Knowledge, vodeća metodologija za upravljanje projektima

posao. Troškovi će rasti, a rokovi će se proširiti, što može imati negativan uticaj na položaj preduzeća i sam projekat.

Proizvod ovog trougla projektnog menadžmenta je *kvalitet*. Projekat koji je zadovoljio tri varijable se može smatrati da je uspješan i da je proizveo dovoljan nivo kvaliteta. Ovaj stav je, međutim, vrlo osporavan od strane projektnih menadžera i u teoriji projektnog menadžmenta i razvoja softvera od strane Haugan (2011, str. 10), Hamilton, Byatt, & Hodgkinson (2010, str. 1), Schwalbe (2006, str. 5) i mnogih drugih. Projekat, u stvari, može zadovoljavati tri varijable, ali kvalitet može biti upitan ili neprihvatljiv za klijenta. Zbog ovoga je u PMBOK metodologiji došlo do evolucije koncepta trougla projektnog menadžmenta, kako će biti opisano u sljedećoj sekciji.

Gledanje na uspješnost projekta kroz zadovoljavanje trougla projektnog menadžmenta, iako neophodno, nije dovoljno. Kao što smo prethodno sugerisali, održavanje trougla u konzistentnom obliku ne znači nužno proizvodnju kvalitetnog proizvoda (softvera). U modernoj teoriji projektnog menadžmenta (Project Management Institute, 2010, str. 165) na ovaj problem se pokušava odgovoriti uspostavljanjem novog koncepta **6 varijabli projektnog menadžmenta**. Za razliku od prethodne tri varijable (troškova, obima i rokova), uvedeni su rizici i resursi kao dvije nove varijable, dok je kvalitet postao šesta. Haugan (2011, str. 11) i Westfall (2010, str. 61) podržavaju ovaj koncept, jer kvalitet zapravo predstavlja varijablu kojom se može upravljati, a ne de facto rezultat procesa projektnog menadžmenta. Kako u teoriji projektnog menadžmenta već postoji šest procesnih modela (Schwalbe, 2006, str. 86) koji su adekvatni navedenim varijablama, logično je kontrolisati projekat kroz njih, kao što je predstavljeno na sljedećem dijagramu:

Slika 1 - Šest varijabli projektnog menadžmenta



Izvor: (Project Management Institute, 2010)

Prema novom modelu, kvalitet dobija veći značaj i nije tako apstraktno definisan kao "rezultat procesa", već kao varijabla, ravnopravna sa svim ostalima. Drugim riječima, projektni menadžer može dogovoriti nivo kvaliteta softvera sa klijentom i na osnovu toga vršiti balans svih varijabli. Na kraju, koncept šest varijabli je realan i u praksi projektnog menadžmenta se uvijek vrši monitoring i kontinuirana evaluacija svih aspekta projekta (Pressman, 2001). Zbog toga je i naglasak ovog rada na **upravljanju kvalitetom**, a ne posmatranje istog kao rezultata procesa ili konstante.

2.2. Generalni pojmovi upravljanja kvalitetom softvera i kvaliteta softvera

2.2.1. Pojam upravljanja kvalitetom softvera

Upravljanje kvalitetom softvera možemo posmatrati kroz jednu praktičnu definiciju kvaliteta softvera od strane Pressmana (2001), koji opisuje kvalitet softvera kao "*usklađenost softvera sa operativnim i*

efektivnim potrebama, koje su jasno iskazane, poštujući standarde proizvodnje i razvoja softvera i postojanje impliciranih karakteristika koje se očekuju od svakog profesionalno razvijenog softvera". U ovoj definiciji možemo identifikovati tri bitne stavke kvaliteta softvera, od kojih svaka govori o potpuno drugačijoj perspektivi, odnosno poimanju pojma kvaliteta softvera.

Kada posmatramo prvu konstataciju, "*kvalitet softvera podrazmijeva usklađenost sa operativnim i efektivnim potrebama*", faktički govorimo o poimanju kvaliteta softvera od strane njegovih budućih korisnika. Operativne i efektivne potrebe su, dakle, potrebe korisnika koje moraju da budu zadovoljene da bi neki softver nazvali kvalitetnim. S obzirom na to da je cilj procesa razvoja softvera direktno u vezi sa zadovoljavanjem potrebe klijenata i da je sama svrha proizvodnja softvera koji klijentima (korisnicima) rješava određene poslovne probleme, jasno je da je kvalitet vezan za proces razvoja softvera.

Kod druge konstatacije "*kvalitet softvera poštuje standarde proizvodnje i razvoja softvera*" uočava se takođe direktna veza između kvaliteta softvera i opšteprihvaćenih standarda razvoja softvera. Pressman jasno identifikuje ovu povratnu spregu, koja pored standarda proizvodnje iziskuje potrebu i za povezivanjem sa standardima kvaliteta softvera.

Treća konstatacija "*kvalitet softvera podrazmijeva postojanje impliciranih karakteristika koje se očekuju od svakog profesionalno razvijenog softvera*" otkriva jednu interesantnu tačku za razmatranje koja se pojavila kroz industrijsku praksu, a koja je izučena i kroz teoriju. Naime, krajnji korisnici softvera pri definisanju svojih zahtjeva za funkcionalnost softvera podrazmijevaju određene zahtjeve za kvalitet softvera. Ovakve vrste zahtjeva nazivamo *nefunkcionalnim zahtjevima* (Khayami, Towhidi, & Ziarati, 2008), jer se ne mogu funkcionalno specifikovati (npr. efikasnost softvera, sposobnost daljeg razvoja i nadogradnje, jednostavnost održavanja, sigurnost, itd).

Upravljanje kvalitetom softvera u najjednostavnijoj mogućoj formi je jedan proces kroz koji možemo definisati kvalitet softvera, ali i planirati, osigurati i provjeriti koliko je softver u razvoju u skladu sa inicijalnim zahtjevima.

S obzirom na to da ni kroz Pressmanove definicije kvaliteta softvera nismo obuhvatili njegove najvažnije aspekte, ali i široko prihvaćene standarde, kao što su ISO standardi, u sljedećim sekcijama ćemo detaljnije posvetiti pažnju samom pojmu kvaliteta softvera. Korektno postavljanje definicije kvaliteta softvera umnogome će pomoći daljem istraživanju o procesu njegovog upravljanja.

2.2.2. Definisane pojma kvaliteta softvera

Kvalitet softvera je definisan u teoriji na nekoliko različitih načina, od kojih je važno izdvojiti dvije definicije:

- *"Stepen u kojem sistem, komponente, ili procesi zadovoljavaju specifične zahtjeve, klijentove potrebe ili očekivanja"*, IEEE (Institute of Electrical and Electronics Engineers, Inc.)
- *"Slaganje sa eksplicitnim izraženim funkcionalnim zahtjevima i zahtjevima za performanse, eksplicitno dokumentovanim razvojnim standardima, i implicitnim karakteristikama koje se očekuju od svakog stručno razvijenog softvera"* (Pressman, 2001)

Kvalitet softvera se takođe može sagledati i kroz definicije klasične teorijske škole kvaliteta:

- *"Sveukupnost osobina proizvoda ili usluge koje mu omogućavaju da zadovolji specifikirane ili implicitne zahtjeve"*, ISO standard 8204
- *"Kvalitet znači slaganje (usaglašenost) sa zahtjevima"*, Crosby (1979)
- *"Kvalitet sadrži one karakteristike proizvoda koje ispunjavaju potrebe klijenta i na taj način obezbjeđuju zadovoljstvo proizvodom. Kvalitet čini sloboda od nedostataka"*, Juran (1988)

Iz definicija kvaliteta softvera koje smo postavili u prethodnim paragrafima može se izvući paralela sa trougлом projektnog menadžmenta, odnosno potvrditi da zaista postoji veza između njega i

ostvarivanja projektnih ciljeva, poštovanja rokova i budžeta. Ovu definiciju potvrđuje i definicija **obezbjedenja kvaliteta softvera** u ISO 9000-3 standardu, koja glasi:

"Obezbjedenje kvaliteta softvera je sistematski, planirani skup akcija da se obezbijedi adekvatna sigurnost da softverski proizvod odgovara uspostavljenim funkcionalno-tehničkim zahtjevima, kao i menadžerskim zahtjevima vremenskog raspoređivanja i budžeta."

S obzirom na to da postoji direktna veza između kvaliteta softvera i projektnog menadžmenta u informacionim tehnologijama, potrebno je dublje sagledati problematiku samog procesa razvoja softvera, u kojem upravo i nastaje nekvalitetan softver.

Pojam kvaliteta koji se pojavljuje u široko prihvaćenoj klasičnoj teoriji se pojavljuje i u radovima *Međunarodne organizacije za standardizaciju*³, koja je, možda i najviše od svih međunarodnih institucija, pridonijela standardizaciji ovog pojma i procesa upravljanja kvalitetom. U sljedećoj sekciji ćemo govoriti nešto više o ISO standardima za kvalitet softvera.

2.3. ISO 9126 standard - karakteristike i proces evaluacije kvaliteta softvera

Jedan od najvećih problema u softverskoj industriji nije definisanje pojma kvaliteta softvera, već zapravo način na koji mjeriti, odnosno *kvantifikovati kvalitet softvera*. Takođe, problem je i na koji način gledati na kvalitet softvera s obzirom na njegovu razliku od drugih materijalnih proizvoda sa kojim se industrija i nauka suočila u prethodnom vijeku. Da bi se standardizovao način mjerenja kvaliteta softvera, 1991. godine razvijen je ISO 9126 standard. Ovaj standard nosi naziv "*evaluacija softvera kao proizvoda - karakteristike kvaliteta softvera i smjernice za njihovo korišćenje*". U ovom standardu se definišu navedene karakteristike kvaliteta softvera, kao i sam proces

³ Engl. ISO – International Organization for Standardization, www.iso.org

evaluacije kvaliteta softvera (Abran, Khelifi, Suryan, & Seffah, 2003). Dakle, prema ovom standardu postoji **šest generalnih karateristika kvalitetnog softvera**:

1. **funkcionalnost** - postojanost funkcija softvera ili njenih naznačenih osobina
2. **pouzdanost** - sposobnost softvera da održava nivo performansi pod navedenim uslovima, u toku navedenog vremenskog perioda
3. **upotrebljivost** - pogodnost za korišćenje softvera, odnosno kompleksnost za krajnjeg korisnika
4. **efikasnost** - odnos između nivoa performansi softvera i količine resursa koji se koriste, pod navedenim uslovima
5. **održivost** - sposobnost softvera da jednostavno podrži buduće modifikacije
6. **prenosivost** - sposobnost softvera da bude prenesen iz jednog u drugo radno okruženje

Pored navedenih karakteristika kvaliteta softvera, ISO 9126 nudi i tri različite **perspektive**, odnosno tri načina poimanja kvaliteta od strane osoba u organizaciji. Naime, *menadžeri* su više zainteresovani za sveukupni kvalitet nego za pojedinačne karakteristike kvaliteta i zbog toga vrše njihovo rangiranje da bi ih usaglasili sa poslovnim zahtjevima. U njihovoj perspektivi je potrebno vršiti unapređenje kvaliteta u skladu sa vremenskim rokovima, troškovima i ljudskim resursima. *Korisnici* su najviše zainteresovani za lakoću korišćenja softvera, performanse i efikasnost i često procjenjuju kvalitet softvera bez poznavanja internih aspekata softvera i procesa razvoja softvera. *Razvojni tim*, s obzirom na to da ima zadatak stvaranja softvera, zainteresovan je za trenutni⁴ i konačni kvalitet softvera.

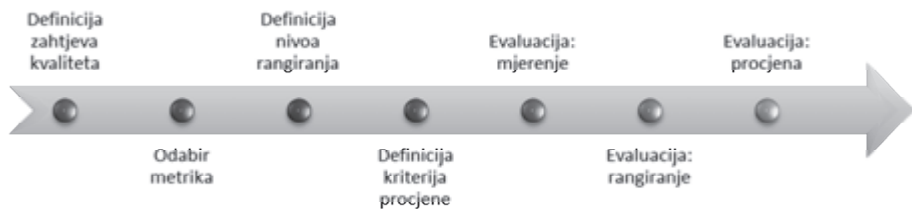
ISO 9126 takođe definiše i nešto što nazivamo **proces evaluacije kvaliteta softvera**. Naime, u toku procesa razvoja softvera

⁴ Trenutni kvalitet softvera se odnosi na kvalitet radnih verzija, odnosno *prototipa* softvera, koji se redovno pojavljuju u ciklusu razvoja softvera; radne verzije služe za evaluaciju napretka projekta, odnosno procjenu podobnosti proizvoda (softvera) u razvoju

se vrši izgradnja softvera, odnosno njegovo planiranje, analiza, dizajn, implementacija i testiranje funkcija. Paralelno sa testiranjem funkcija potrebno je vršiti i provjeru kvaliteta softvera, odnosno test kroz koji će se krajnji korisnici uvjeriti da dobijaju softver koji odgovara kako njihovim, tako i generalnim zahtjevima kvaliteta.

Proces evaluacije kvaliteta softvera započinje sa definicijom zahtjeva kvaliteta, odnosno navedenih i podrazumijevanih zahtjeva iz relevantne tehničke dokumentacije i samih ISO standarda, na osnovu čega se kreira specifikacija zahtjeva.

Slika 2 - Proces evaluacije kvaliteta softvera



Izvor: ISO 9126

Priprema evaluacije je drugi korak i uključuje odabir odgovarajućih metrika⁵ (kvantitativnih), definisanje nivoa rangiranja metrika (raspon vrijednosti koje se računaju kao zadovoljavajuće ili nezadovoljavajuće) i definiciju kriterijuma procjene (priprema procedura za sumiranje rezultata evaluacije).

Treći korak je procedura evaluacija, koja se sastoji iz tri koraka: mjerenja, u toku kojeg se odabrane metrike apliciraju na softver radi dobijanja određenih vrijednosti; rangiranja, u kojem se karakteristike softvera unose u raspon između zadovoljavajućeg ili nezadovoljavajućeg; i procjene, na osnovu koje se vrši sumiranje rezultata evaluacije. Konačni kvalitet softvera se poredi sa drugim aspektima projekta razvoja softvera, kao što su vremenski rokovi i

⁵ U softverskom inženjstvu riječ "metrika" se koristi u smislu procedure, odnosno predstavlja način ili opis mjerenja određenog objekta. ISO 9126 standard koristi riječ metrika u kontekstu formula koje se primjenjuju nad određenim modulima softvera da bi kao rezultat dobili kvantitativne rezultate koji nam govore o procentu ostvaranje ciljeva kvaliteta (Abran A. , 2010, str. 3).

troškovi, radi podrške daljim poslovnim odlukama.

Na samom kraju, neophodno je napomenuti da se ISO standard 9126 godinama razvijao, te da danas sadrži i nekoliko pridodatih standarda, kako je navedeno:

- ISO 9126-1: Model kvaliteta
- ISO TR 9126-2: Eksterne metrike
- ISO TR 9126-3: Interne metrike
- ISO TR 9126-4: Metrike kvaliteta u korišćenju

Iako u ovom radu nećemo detaljno izučavati sve podskupove ISO 9126 standarda, važno ih je predstaviti, posebno radi upoznavanja projektnih menadžera i drugih odgovornih ljudi u industriji sa relevantnim standardima kvaliteta softvera. Najveću pažnju ćemo posvetiti definisanim karakteristikama kvaliteta softvera, koje se dalje rafinišu kao što je opisano u sljedećoj sekciji.

2.3.1. Klasifikacija karakteristika kvaliteta softvera prema ISO 9126 standardu

Šest karakteristika kvaliteta softvera koje smo naveli u prethodnoj sekciji se prema ISO 9126 standardu dalje razlažu na **potkarakteristike** i **attribute** (dijagram ispod). Atributi treba da budu mjerljivi, odnosno moraju da budu pitanja na koje se može odgovoriti objektivno, kvantitativnim vrijednostima.

ISO 9126 standard definiše **generalne potkarakteristike** (tabela ispod), koje se mogu implementirati u gotovo svim vrstama projekata razvoja i evaluacije kvaliteta softvera. Iako ove potkarakteristike nisu sveobuhvatne, do sada su najrazvijeniji set potkarakteristika u teoriji (Abran, Khelifi, Suryin, & Seffah, 2003).

Tabela 1 - ISO 9126 karakteristike i potkarakteristike

Funkcionalnost	Pogodnost	Da li softver izvršava funkcije kao što je previđeno?
	Tačnost	Da li rezultati funkcija softvera tačni?
	Interoperabilnost	Da li softver može vršiti interakciju sa drugim sistemima?
	Sigurnost	Da li softver onemogućuje neautorizovani pristup?
Pouzdanost	Zrelost	Da li je većina značajnih grešaka eliminisana kroz korišćenje softvera?
	Tolerancija grešaka	Da li je softver sposoban tolerisati greške korisnika?
Upotrebljivost	Sposobnost oporavka	Da li je softver sposoban da se vrati u radno stanje nakon greške ili nestabilnosti?
	Razumljivost	Da li korisnik razumije kako da koristi softver?
	Lakoća učenja	Da li korisnik može relativno jednostavno da nauči koristiti softver?
Efikasnost	Jednostavnost	Da li korisnik može koristiti sistem uz minimum napora?
	Atraktivnost	Da li je korisnički interfejs atraktivan?
	Odziv	Da li softver brzo reaguje, tj. koliko dugo izvršava određene operacije?
Održivost	Iskorištenost resursa	Da li softver koristi resurse u srazmjeri sa funkcijom koju obavlja?
	Sposobnost analize	Da li se greške mogu lako analizirati?
	Promjenljivost	Da li se softver može lako mijenjati i dorađivati?
Prenosivost	Stabilnost	Da li softver funkcioniše pored promjena?
	Sposobnost testiranja	Da li se softver može jednostavno testirati?
	Adaptivnost	Da li se softver može prebaciti u drugo radno okruženje?
	Sposobnost instalacije	Da li se softver može relativno jednostavno instalirati?
Sve karakteristike	Sukladnost	Da li je softver u skladu sa standardima prenosivosti?
	Zamljenljivost	Da li softver može jednostavno zamijeniti drugi softver?
	Saglasnost	Da li su softver i njegove karakteristike u skladu sa zakonima i ostalom regulativom?

Izvor: (Abran A. , 2010)

Jedna od najvažnijih prednosti ISO 9126 modela je u definisanju kako internih, tako i eksternih karakteristika kvaliteta softvera. Sa druge strane, ovaj standard ne formuliše jasno način na koji se mogu vršiti mjerenja, kako tvrdi Maryoly (2002). Na individualnim projektnim timovima i organizacijama je da samostalno razviju i apliciranju mjerenja. Ovo za projektne timove predstavlja svojevrsni izazov, jer je potrebno izvršiti pažljiv odabir metrika, tako da one omogućavaju objektivno, a ne subjektivno mjerenje i procjenu nivoa kvaliteta softvera.

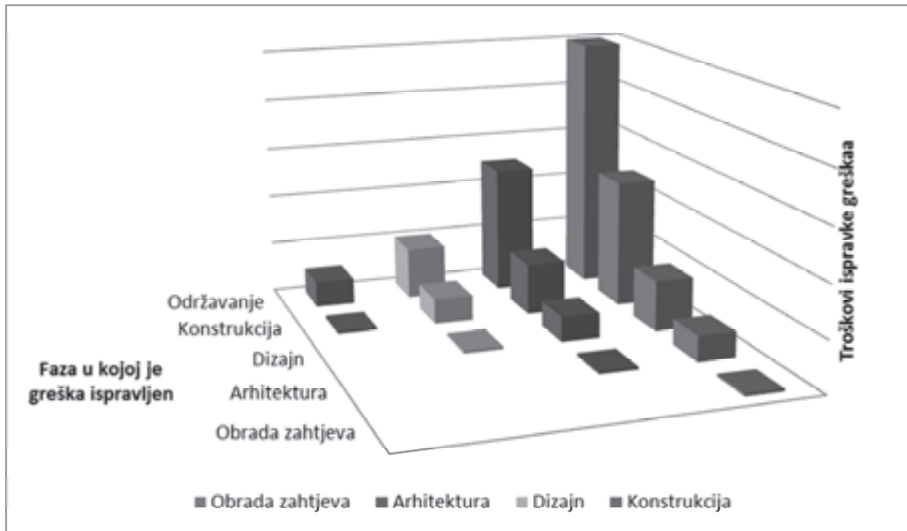
2.3.2. Statična i kontinuirana kontrola kvaliteta softvera

Testiranje softvera je u tradicionalnom pristupu razvoja softvera⁶ posljednja faza u razvoju u toku koje su se otklanjale posljednje greške, koje su mogle biti ustanovljene ovim putem. Međutim, moderni razvoj softvera podrazumijeva korišćenje metodologije koja omogućava integraciju testiranja u kompletan razvojni ciklus, ne samo pred samu distribuciju softvera krajnjim korisnicima.

Jedan od problema koji se rješava kontinuiranim testiranjem je eliminisanje *troškova otklanjanja grešaka*. Naime, greške koje se pojavljuju u posljednjim danima razvoja softvera najviše koštaju, jer su nastale mnogo ranije i zbog toga je potreban veći angažman ljudskih resursa za njihov ispravak. Greške se tako, prema McConnellu (1997), najjednostavnije i najekonomičnije ispravljaju u trenutku kada su nastale, odnosno u odgovarajućoj fazi razvoja softvera, kako je ilustrovano na sljedećem dijagramu.

⁶ Waterfall metodologija i slične

Slika 3 - Troškovi ispravke grešaka u procesu razvoja softvera u zavisnosti u kojoj fazi je greška identifikovan i ispravljena



Izvor: (McConnell S., 1997)

2.3.3. Troškovi kvaliteta softvera

Troškovi kvaliteta predstavljaju zbir **usaglašenih** i **neusaglašenih** troškova. Usaglašeni, odnosno standardni troškovi, porazumijevaju troškove koji su nastali razvojem softvera koji je u skladu sa zahtjevima klijenta i koji je podoban za korišćenje. Neusaglašeni troškovi podrazumijevaju troškove grešaka i neispunjenosti očekivanog nivoa kvaliteta.

Prema istraživanju RTI Internationala (2002), neusaglašeni troškovi kvaliteta u preduzećima u Sjedinjenim Američkim Državama je iznosio 59,6 milijardi američkih dolara, odnosno 0,6% BDP-a (bruto društveni proizvod). U istraživanju se sugerše da više od jedne trećine ovih troškova može biti eliminisano unapređenjem procesa testiranja, odnosno kontrole kvaliteta softvera i uklanjanjem grešaka.

Schwalbe (2006) definiše, pored usaglašenih i neusaglašenih troškova, pet najznačajnijih kategorija troškova kvaliteta:

1. **Trošak prevencije** - trošak planiranja i izvršenja projekta koji ne sadrži greške ili sadrži greške u prihvatljivom rasponu
2. **Trošak procjene** - trošak evaluacije procesa i izlaza radi osiguranja da projekat ne sadrži greške ili sadrži greške u prihvatljivom rasponu
3. **Trošak internog neuspjeha** - trošak identifikovanja i ispravljanja grešaka prije nego što je proizvod isporučen klijentu
4. **Trošak eksternog neuspjeha** - trošak koji se pojavljuje nakon isporuke proizvoda klijentu, odnosno troškovi u sklopu garantnog roka proizvoda i naknadne popravke
5. **Troškovi mjerenja i opreme za testiranje** - kapitalni troškovi opreme (ili softvera) koji se koristi za detekciju i prevenciju grešaka i aktivnosti procjene

Pomenutih pet kategorija troškova kvaliteta se mogu direktno povezati sa klasifikacijom na usaglašene i neusaglašene troškove na sljedeći način: troškovi prevencije, procjene, internog neuspjeha, mjerenja i opreme za testiranje dovodimo u vezu sa usaglašenim troškovima, dok trošak eksternog neuspjeha dovodimo u vezu sa neusaglašenim troškovima.

Top menadžment preduzeća je u mnogim slučajevima direktno odgovoran za visoke nestandardne troškove u informacionim tehnologijama, jer često požuruju sa razvojem novog softvera i plasiraju na tržište (RTI International, 2002). Brzim reakcijama na nove tržišne situacije i konkurenciju preskače se veći dio razvoja i testiranja. Rješavanje ovog problema jedino je moguće fokusiranjem top menadžmenta na korporativnu kulturu koja obuhvata kvalitet.

3. METODOLOGIJA ISTRAŽIVANJA

Eksperimentalno istraživanje će izvršiti evaluaciju kvaliteta softvera nad novim projektom razvoja softvera, odnosno pratiće se nivo kvaliteta softvera od prve do posljednje faze i vršiti monitoring i korektivne akcije. Istraživanje je sprovedeno u kompaniji "*Component Sourcing Group America, Laguna Hills, California*" na projektu razvoja softverskog alata za praćenje inventara putem mobilnih platformi, koji je autor ovoga rada vodio.

Radi evaluacije kvaliteta softvera u okviru projekta je sa klijentom postignut pisani dogovor o *minimalnom nivou kvaliteta softvera*. Ovaj dogovor je specifikovan kroz tabelarni prikaz po individualnim karakteristikama i potkarakteristikama kvaliteta softvera. Naime, specifikacija minimalnog nivoa kvaliteta softvera treba da bude numerički izražena za cjelokupan proizvod, ali i detaljno kroz minimalan nivo kvaliteta karakteristika i potkarakteristika softvera.

Odabir metrika je prvi korak u procesu evaluacije kvaliteta softvera. Cilj je od svih definisanih metrika u ISO 9126 standardu odabrati one koje su najpogodnije za mjerenje kvaliteta softvera u projektu.

Definicija nivoa rangiranja je proces u kojem se određuje koji skup vrijednosti se smatra kao zadovoljavajući rezultat mjerenja, a koji ne. Nivo rangiranja zavisi od tipa metrike, odnosno od rezultata koje ona proizvodi. Na primjer, metrika za evaluaciju pogodnosti, pod nazivom "adekvatnosti funkcije" (engl. functional adequacy) se izračunava putem sljedeće jednačine⁷:

$$X = 1 - A/B$$

U ovoj formuli A predstavlja broj funkcija u kojima su detektovani problemi, dok B predstavlja ukupan broj testiranih funkcija. Rezultati ove jednačine se nalaze između 0 i 1, odnosno matematički predstavljeno: $0 \leq X \leq 1$. Kada postavljamo nivoe

⁷ Jednačina za izračunavanje metrike "adekvatnosti funkcije" je definisana u ISO 9126-3 standardu (ISO/IEC, 2002, str. 14)

rangiranja, zapravo postavljamo minimalnu vrijednost ispod koje se smatra da metrika proizvodi negativan rezultat. U ovom primjeru, pozitivan rezultat može biti ispunjen samo ako je $X \geq 0,75$.

Definicija kriterijuma procjene opisuje kako će se rezultati mjerenja kroz različite metrike sumirati. Preporučeni način sumiranja mjerenja za potrebe ovog modela je računanje na osnovu *ponderisanog prosjeka*⁸. Naime, za svaku metriku će se dodati poseban rang, odnosno ponder.

Definicija nivoa rangiranja i kriterijuma procjene kao rezultat proizvodi osnovni tabelarni prikaz, kako slijedi:

Tabela 1 - Rezultat definicije nivoa rangiranja i kriterija procjene

Funktionalnost	Pogodnost	Eksterna	Adekvatnost funkcija	$X \geq 0,75$	R=5
			Tačnost implementacije funkcija	$X \geq 0,90$	R=10
	Interna	Adekvatnost funkcija	$X \geq 0,75$	R=5	
		Tačnost implementacije funkcija	$X \geq 0,75$	R=10	
		Stabilnost specifikacije funkcionalnosti	$X \geq 0,25$	R=2	

Izvor: autor

Mjerenja su primarni radni zadaci za kontrolore kvaliteta, koji se moraju adekvatno rasporediti imajući na umu da često obavljaju i druge poslove (aktivni su pod drugim projektnim ulogama). Slučaj dijeljenja projektnih uloga je najčešći u malim projektnim timovima, gdje mali broj ljudi mora obavljati veći obim poslova (Todorović, 2010, str. 4).

⁸ Prema preporučenim načinom sumiranja – ponderisani prosjek (Saphire, 2008, str. 13)

Kontrolori kvaliteta će, dakle, u toku projekta vršiti mjerenja, odnosno *kvantitativno izračunati i zabilježiti vrijednost pojedinog mjerenja*. Mjerenja se vrše po vremenskom rasporedu metodologije unifikovanog procesa, odnosno na kraju svake faze i na kraju svake iteracije.

4. PREZENTACIJA, ANALIZA I DISKUSIJA REZULTATA ISTRAŽIVANJA

Kako smo naveli, za odabrani projekat će se vršiti mjerenja nivoa kvaliteta softvera po pojedinim fazama projekta. Projekat je kao metodologiju razvoja softvera koristio unifikovani proces, koji se klasifikuje kao metodologija iterativnog pristupa razvoja softvera⁹. U sklopu ove metodologije postoje četiri faze razvoja softvera - priprema, razrada, konstrukcija i prelaz. Svaka od ovih faza se fokusira na posebne zadatke, tj. discipline i aktivnosti. U pripremi je tako fokus na pripremi projektne dokumentacije, korisničkih zahtjeva; u razradi na analizi problema i dizajnu rješenja; u konstrukciji na detaljnoj analizi i stvaranju softverskog rješenja (programiranju), kontroli kvaliteta; i u prelazu u finalnim programerskim zadacima, testiranju, finaliziranju dokumentacije i završenju postavljanja softvera u produkciono okruženje. Upravo na kraju svake od ovih faza će se vršiti evaluacija kvaliteta softvera, jer, prema iterativnom pristupu (Gibbs, 2007), a i samoj metodologiji unifikovanog procesa, softver se razvija od prve pa do posljednje faze. Dakle, postojaće radne verzije softvera, koje se vremenom dopunjuju sa funkcionalnošću i integrišu u jednu cjelinu.

⁹ Iterativni i postepeni pristup razvoju softvera (engl. iterative and incremental software development process) počiva na principima proizvodnje velikog broja radnih verzija softvera. Ovaj princip omogućava evaluacije softvera, tj. radnih modela aplikacija od strane klijenta, koji ima priliku da od samog početka projekta bude aktivno uključen u izradu finalne verzije softvera. Pored uključivanja krajnjeg korisnika u proces razvoja softvera, razvoj manjih, jednostavnijih modula softvera omogućava njihovo sazrijevanje i kvalitetniju doradu (Kroll, i drugi, 2005).

Tabela 2 - Rezultat evaluacije kvaliteta softvera kroz sve faze projekta

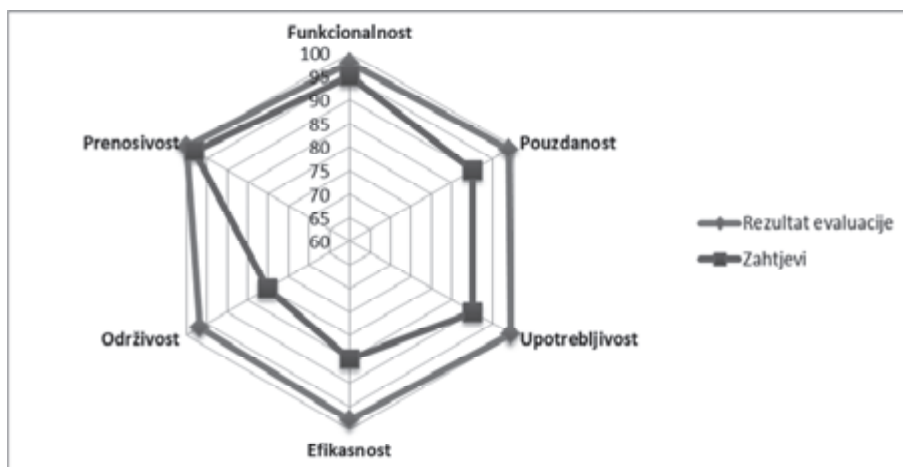
Funkcija sistema	Karakteristike kvaliteta softvera														
	Faza u procesu razvoja softvera ¹⁰						Faza u procesu izvođenja softvera ¹⁰								
	Funkcionalnost		Pouzdanost		Upotrebljivost		Efikasnost		Održivost		Promjenljivost				
P	R	K	Z	P	R	K	Z	P	R	K	Z	P	R	K	Z
Manufacturers data entry	20	65	100	100	80	100	100	100	100	100	100	100	100	100	100
Vendors data entry	50	100	100	100	80	100	100	100	100	100	100	100	100	100	100
Parts data entry	100	100	100	100	80	100	100	100	100	100	100	100	100	100	100
Part categories data entry	100	100	100	100	80	100	100	100	100	100	100	100	100	100	100
Part serial number generation	60	100	100	100	100	100	100	100	100	60	80	100	50	75	100
Part barcode generation	80	100	100	100	100	100	100	100	100	60	80	100	50	75	100
Asset data entry	40	80	100	100	80	100	100	100	100	100	100	100	50	75	100
Define asset parts and quantities	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Define purchasing information	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Bulk insert of barcodes	50	100	75	100	100	80	100	80	100	20	30	80	95	50	75
Locations data entry	80	100	100	100	80	100	100	100	100	80	100	100	100	100	100
Sub-locations data entry	70	100	100	100	80	100	100	100	100	100	100	100	100	100	100
Inventory add/move	50	100	100	100	60	90	100	100	100	100	100	80	100	100	100
Inventory delete	100	100	100	100	60	90	100	100	100	100	100	80	100	100	100
Inventory check out	45	80	70	95	60	90	100	100	70	100	100	100	80	100	100
Inventory check in	60	80	70	95	60	90	100	100	70	100	100	100	80	100	100
Inventory reserve	50	60	60	90	60	90	100	100	70	100	100	100	80	100	100
Mobile device check in	0	85	95	95	0	85	100	100	0	80	100	100	3.5	50	80
Mobile device check out	0	80	100	100	0	85	95	90	0	80	100	100	3.5	50	80
Mobile device location change	0	100	100	100	0	60	90	100	0	80	100	100	3.5	50	100
Mobile device data sync	0	40	65	90	0	30	60	90	0	100	100	100	0	40	40
Create inventory report	50	100	100	80	100	100	100	80	100	100	100	50	80	100	100
Create assets report	0	100	100	100	80	100	100	100	80	100	100	100	80	100	100
Export database	0	70	65	85	0	100	100	100	0	100	100	100	0	25	50
User management	0	80	100	100	0	100	100	100	0	100	100	100	0	80	100
Group management	0	80	100	100	0	100	100	100	0	100	100	100	0	80	100
Permission management	0	0	60	85	0	0	80	100	0	0	80	100	0	0	100
Database configuration	0	0	100	100	0	0	50	85	0	0	50	85	0	0	60
Inventory data consistency check	0	0	70	100	0	0	100	100	0	0	100	100	0	0	100
Manage asset documents	50	50	80	100	100	100	80	100	100	100	85	100	50	70	80
Manage asset photos	50	50	80	100	100	100	80	100	100	100	85	100	50	70	80

Izvor: autor

¹⁰ P – Priprema, R – Razrada, K – Konstrukcija, Z – Prelaz; faze metodologije unificiranog procesa

Na prethodnoj stranici je prikazan sumarni rezultat evaluacije kvaliteta softvera kroz četiri faze, funkcije i karakteristike kvaliteta softvera. Rezultate možemo sumirati kroz "radar" grafikon, kako slijedi:

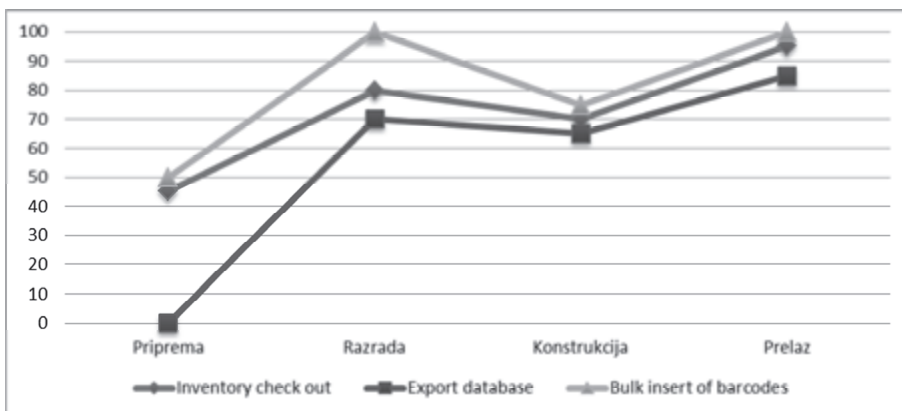
Grafikon 2 - Prikaz zahtjeva za kvalitet softvera i rezultata evaluacije kvaliteta softvera, na sumarnom nivou, za svaku od karakteristika kvaliteta softvera, na kraju projekta razvoja softvera



Izvor: autor

Iz grafikona se vidi da je softver zadovoljio zahtijevani nivo kvaliteta softvera. Kada se osvrnemo na rezultate, konačni rezultati nakon posljednje faze projekta su zadovoljavajući. Postavlja se pitanje da li bi ti rezultati bili zadovoljavajući i bez korišćenja ISO 9126 standarda i da li je ovaj standard uspio da doprinese kontinuiranom unapređenju kvaliteta softvera? Da bismo o ovome povelu diskusiju, pogledaćemo primjer mjerenja nivoa kvaliteta softvera kroz jednu karakteristiku kvaliteta softvera, ali kroz različite faze u projektu.

Grafikon 3 - Rezultati mjerenja nivoa kvaliteta softvera u karakteristici "funkcionalnost", za tri odabrane funkcije, kroz vremensku perspektivu četiri faze unifikovanog procesa

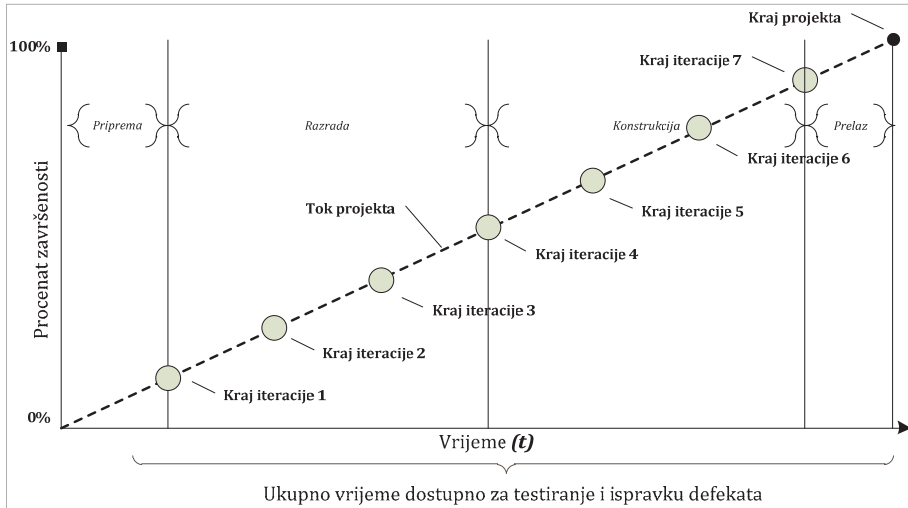


Izvor: autor

Kao što je vidljivo iz prethodnog grafikona, ISO 9126 standard je uticao na varijaciju u rezultatima mjerenja. Tačnije rečeno, s obzirom na to da promoviše kontinuirano testiranje, ISO 9126 projektne timove obavezuje da pronalaze greške i loš kvalitet softvera (Balla, 1995), te da vrše korektivne akcije. Ovo je upravo i vidljivo, jer se *varijacije ispoljavaju na pozitivan način*, tj. kvalitet softvera se kontroliše i kontinuirano unapređuje. Takođe, da kontrola kvaliteta nije kontinuirana, timovi ne bi uvidjeli pad kvaliteta, kao što se vidi da se dogodilo u fazi konstrukcije, te ne bi ispravili uočene greške do kraja projekta.

Jarzombek (1999), Schwalbe (2006), Kaner et al (2002) i mnogi drugi autori tvrde da kontrola kvaliteta treba da se vrši upravo na način koji sugeriše ISO 9126, a to je kontinuirano. U suprotnom, ako ne testiramo softver kontinuirano, kao što McConnell (1997) ukazuje, ukoliko se problem (greška) otkrije u kasnijoj fazi projekta, njen ispravak je eksponencijalno skuplji.

Slika 4 - Kontrola kvaliteta u iterativnim metodologijama, kakav je unifikovani proces



Izvor: (Gibbs, 2007, str. 178)

Diskutujući dalje o ovome možemo reći da je ovakav proces *efikasniji i efektivniji* od tradicionalnog pristupa razvoja softvera po pitanju unapređenja kvaliteta softvera. Kada kažemo "efektivniji", mislimo prvenstveno na mnogo značajnije efekte kontinuiranog testiranja, kako sugeriše Chemuturi (2010). Jedan od najvažnijih efekata je stvaranje korporativne kulture, odnosno ugrađivanje procesa kontrole kvaliteta softvera u projektni tim (Eisenberg, 2004). Sa druge strane, kada kažemo "efikasniji", govorimo o značajno boljem procesu kontrole kvaliteta softvera, koji identifikuje problematične i rizične elemente softvera u svakom trenutku (svim fazama), a ne samo na kraju životnog ciklusa razvoja softvera, kada je proizvod spreman za isporuku.

5. ZAKLJUČAK

Kvalitet softvera je u teoriji takođe vrlo raširen pojam, ali dosadašnji naponi domaćih i inostranih autora su donijeli jedino ISO 9126 standard po kojem bi se vršila evaluacija kvaliteta softvera. Međutim, to je samo standard koji definiše izlazne kriterijume, odnosno konačno stanje, i kao takav nije najbolje prilagođen za projektne timove i menadžment. Ovaj rad je predstavio strukturisani i sistematizovani proces putem kojeg se vrši kontrola kvaliteta softvera u IKT projektima.

Praktičnim istraživanjem je predstavljena primjena ISO 9126 modela na proces razvoja softvera i rezultati su pokazali da se dostiže vrlo visok nivo kvaliteta softvera kao krajnji rezultat. Takođe, ukazuje se i na značaj kontinuiranog testiranja, odnosno koliko taj pristup koristi projektnim timovima da prate nivo kvaliteta softvera i vrši korektivne akcije na njegovom održavanju i unapređenju. Istraživanjem je potvrđena osnovna hipoteza, a to je da je ISO 9126 standard odgovarajući za ovu namjenu, da donosi pomenute rezultate, tj. da je primjenljiv, efikasan i efektivan.

S obzirom na tendenciju približavanja Evropskoj uniji, naš region ima veliku šansu da formira outsourcing preduzeća koja će vršiti razvoj softvera za korporacije iz EU. Kako bude tekao razvoj domaćih preduzeća, biće potrebno ponuditi inostranim partnerima ne samo jeftinu radnu snagu za ove visokotehnološke poslove, već se i dokazati pred mnogo jačom internacionalnom konkurencijom. Ovo je moguće samo fokusirajući se na primarni cilj procesa razvoja softvera, a to je kvalitet. Kvalitet je taj faktor koji nas može diferencirati od drugih konkurenata, a putem njega možemo doprinijeti popularizaciji informacionih tehnologija, stvaranju novih radnih mjesta u sferi softverskog, pa i hardverskog inženjeriga. Fokus nam, dakle, mora biti kvalitet i konkurentnost, a modifikovani model unifikovanog procesa upravo može pomoći projektnim timovima svih veličina da usvoje najbolju praksu i koriste je za unapređenje svog poslovanja, a na samom kraju i na razvoj države i regije.

LITERATURA

1. **Abran, Alain. 2010.** *Software Metrics and Software Metrology.* Hoboken : John Wiley & Sons Inc., 2010. 978-0-470-59720-0.
2. **Balla, K. 1995.** *Software Quality Management : ISO 9000, but not only.* Eindhoven : Eindhoven University of Technology, 1995.
3. **Chemuturi, Murali. 2010.** *Mastering Software Quality Assurance: Best Practices, Tools and Technique for Software Developers.* s.l. : J. Ross Publishing, 2010. str. 358. 9781604270327.
4. **Eisenberg, Bart. 2004.** *Achieving Zero-Defects Software.* s.l. : Pacific Connection, 2004.
5. **Gibbs, R. Dennis. 2007.** *Project Management with the IBM® Rational Unified Process®: Lessons from the Trenches.* Upper Saddle River : Prentice Hall Professional, 2007. 9780321336392.
6. **Haugan, Gregory T. 2011.** *Project Management Fundamentals: Key Concepts and Methodology.* Vienna, VI : Management Concepts Inc., 2011. 9781567262810.
7. **ISO/IEC. 2002.** *ISO 9126-3: Software engineering – Product quality – Part 2 (Internal metrics).* s.l. : ISO/IEC, 2002.
8. **Jarzombek, J. 1999.** 1999. The 5th annual jaws s3 proceedings.
9. **Kaner, C., Bach, J. / Pettichord, B. 2002.** *Lessons Learned in Software Testing.* s.l. : John Wiley & Sons, 2002.
10. **Kroll, Per / Royce, Walker. 2005.** Key Principles for Business-Driven Development. *The Rational Edge.* 2005, October 2005.
11. **Maryoly, O., Perez, M.A. / Rojas, T. 2002.** *A systemic quality model for evaluating software products.* s.l. : Laboratorio de Investigcin en Sistemas de Informacin, 2002.
12. *Measurable Quality Charaeteristics of a Software System on Software Architecture Level.* **Khayami, S. R., Towhidi, A. / Ziarati, K. 2008.** 3, 2008, International Review on Computers and Software (I.RE.CO.S), T. 3, str. 234-239.

13. **OECD. 2010.** *OECD Information Technology Outlook 2010*. s.l. : OECD Publishing, 2010. 9789264084667.
14. **Pressman, Roger S. 2001.** *Software Engineering: A Practitioner's Approach*. Boston : McGraw-Hill, 2001.
15. **Project Management Institute. 2010.** *A Guide to the Project Management Body of Knowledge*. Newton Square, PA : Project Management Institute, 2010. 9780446526562.
16. **RTI International. 2002.** *Software Bugs Cost U.S. Economy \$59.6 Billion Annually*. s.l. : RTI International, 2002.
17. **Saphire. 2008.** Public Deliverables - Delivery of the Assessment Criteria for Software Validation. *Saphire - Intelligent Healthcare Monitoring*. [Na mreži] 2008. [Citirano: 21 09 2010.] <http://www.srdc.metu.edu.tr/webpage/projects/saphire/>.
18. **Schwalbe, Kathy. 2006.** *Information Technology Project Management*. Boston : Thomson Course Technology, 2006. 0-619-21528-3.
19. *The Unspoken Additional Constraint of Project Management*. **Hamilton, Gary, Byatt, Gareth / Hodgkinson, Jeff. 2010.** 2010, American Society for the Advancement of Project Management.
20. **Todorović, Boris. 2010.** Implementacija unificiranog procesa kao metodologije razvoja softvera u malim projektnim timovima. "Aktuelnosti", *Banjaluka College*. 12 2010.
21. *Upstream Decisions, Downstream Costs*. **McConnell, S. 1997.** 1997, Windows Tech Journal.
22. *Usability Meanings and Interpretations in ISO Standards*. **Abran, A., i drugi. 2003.** 4, 2003, Software Quality Journal, T. 11, str. 325-338.
23. **Westfall, Linda. 2010.** *The Certified Software Quality Engineer Handbook*. Milwaukee : ASQ Quality Press, 2010. 0873897307.

Rad je zaprimljen:14.06.2012
Rad je prihvaćen:18.06.2012.