



ULOGA I OSNOVNE KARAKTERISTIKE KVALITETA SOFTVERA

Mirjana Bošković¹, Olgica Nikolin²

Rezime: Dokument "Uloga i osnovne karakteristike kvaliteta softvera" predstavlja opis kvaliteta softvera, posmatrajući ga iz različitih perspektiva. Sadrži veći broj definicija, ISO standard 8204, ISO standard 9126, faze kvaliteta softvera, opštu klasifikaciju atributa kvaliteta, veze između atributa kvaliteta, CMM (Capability Maturity Model), i kontrolu kvaliteta kroz testiranje.

Ideja vodilja je bila sakupiti definicije različitih autora, sa akcentom na ulogu i značaj koji kvalitet ima u izgradnji softvera. Ovaj dokument je pisan, u cilju da bude vodič ili podsetnik za već iskusne programere. Želja autora je da na jednom mestu budu sakupljene sve esencijalne informacije o kvalitetu softvera, te da bude inspiracija za izradu isključivo kvalitetnog softverskog proizvoda, koji će poštovati standarde i posedovati sve glavne attribute koji čine kvalitet, što vodi ka implementaciji pouzdanog, upotrebljivog, lako korišćenog, bezbednog i efikasnog (što su samo neke od osobina kvaliteta) softverskog proizvoda. Dovoljno pažnje posvećeno ovom segmentu u životnom ciklusu razvoja softverskog proizvoda značajno će smanjiti broj pojave nepovoljnih događaja na relaciji korisnik-softverski proizvod. Ako je dovoljno energije usmereno na ovu problematiku svakako neće izostati lakoća korišćenja i zadovoljstvo korisnika.

Ključne reči: kvalitet, softver

INTERPRETATION AND MAIN CHARACTERISTICS OF SOFTWARE QUALITY

Summary: Document "Interpretation and main characteristics of software quality" present description of software quality. It is consist of definitions, ISO 8204 and 9126 standards, phases of software quality, general classification of quality attributes, the relations between quality attribute, CMM (Capability Maturity Model), and quality control with testing.

The idea was to select the definition of different authors with with accentuation on interpretation and importance which quality has in software realisation. This document was written as guide for expert programers. The main goal was to collect information for realisation quality software using standards to get good, reliable and easy for use software product. Here is describe software life cycle development and how to remove all problems in software realisation.

¹ Mirjana Bošković, Tehnički fakultet "Mihajlo Pupin" Zrenjanin, E-mail: gogaa@neobee.net

² Olgica Nikolin, Tehnički fakultet "Mihajlo Pupin" Zrenjanin,

Key words: *quality, software*

1. UVOD

Razvoj softvera zahteva zadovoljenje određenih ciljeva i zahteva. Jedan od zahteva je i kvalitet. Kvalitet se može posmatrati iz različitih uglova. Iz svakog od tih uglova definisan je na drugi način.

Softversko inženjerstvo je grana informatike koja se bavi proučavanjem standarda pri razvoju softvera. Svrha softverskog inženjerstva je naći put izgradnje kvalitetnog softvera. Na temu kvaliteta i kvaliteta softvera postoji pregršt definicija i nekoliko standarda. Neki od standarda su ISO 8204, ISO 9126, IEEE standard.

Pre nego što je kvalitet počeo da igra bitnu ulogu u svetu informacionih tehnologija i u softverskoj industriji, često je povezivan sa fizičkim objektima, ili sistemima.

Kvalitet se povezuje sa proizvodnim procesom. Fokus je na sigurnosti. Treba zadovoljiti zahtev da proizvod odgovara specifikaciji.

Otkad su mnoge tačke u specifikaciji proizvoda određene, kao i uslovi i tolerancija greške, smanjenje varijacija u proizvodnji je postalo ključna tačka statističke kontrole kvaliteta.

Sa razvojem uslužne industrije, pojavio se nov pogled na kvalitet (Reichheld Jr. Sasser, 1990). On glasi: "Kvalitet je poslovna potreba za regulisanjem dinamične promene očekivanja kupca, sa fokusom na promenu kontrole kvaliteta od nula nedostataka u razvoju proizvoda do nula nedostataka kad kupac preuzme proizvod". [4]

Kvalitet softvera je moguće izmeriti samo onda kada su zahtevi definisani zajedno sa karakteristikama koje su relevantne za kvalitet [7].

2. DEFINICIJE

Sa stanovišta softverskog inženjerstva, kvalitet je jedan od nekoliko važnih faktora, uključujući cenu, planiranje i funkcionalnost. Neki od istraživača koji su proučavali ovu temu su: Blum, 1992; Humphrey, 1989; Ghezzi, 2003; von Mayrhauser, 1990.

Svaka definicija o softverskom kvalitetu trebala bi da bude orijentisana ka potrebama korisnika. Po Crosby-ju kvalitet je "prilagođavanje zahtevima". Prilikom definicije kvaliteta treba posmatrati softver iz korisnikove perspektive, [8]. Ključna pitanja su: ko su korisnici, šta im je važno i kako njihovi prioriteti utiču na izradu softvera ?

Kvalitet softvera je planirani i sistematski skup aktivnosti koji obezbeđuje kvalitet ugrađen u softver. Sadrži pouzdanost, kontrolu i inženjering kvaliteta softvera. Kao atribut, kvalitet softvera je (1) stepen po kojem sistem, komponenta, ili proces ispunjavaju određene zahteve; (2) stepen po kojem sistem, komponenta, ili proces ispunjavaju potrebe ili očekivanja korisnika, [2].

Prema Prahald-u i Krishnan-u (1999) visoko-kvalitetan softver se može definisati pomoću tri osnovna elementa: koegzistencija, prilagodljivost i inovacija.

Uvek kada postavljamo pitanje "Šta je kvalitet softvera?", otvaraju se nova pitanja, kao što je "Koje su karakteristike visoko-kvalitetnog softvera?" i "Koje su osnovne karakteristike kvaliteta softvera ?".

Kvalitet možemo posmatrati iz različitih uglova. Sa stanovišta Kitchenham-a i Pfleegera (1996,2002) postoji pet pogleda na kvalitet. To su: transcendentalan, korisnički, pogled

zasnovan na proizvodnji, pogled zasnovan na proizvodu, i pogled zasnovan na vrednostima.

- Kod transcendentnog pogleda, kvalitet je teško definisati ili opisati u kratkim crtama, ali se može prepoznati ako je prisutan.
- Sa stanovišta korisničkog pogleda, kvalitet je sposobnost pronalaženja svrhe ili sposobnost primećivanja potreba korisnika.
- U pogledu proizvodnje, kvalitet znači poštovanje standarda.
- U pogledu proizvoda, kvalitet je fokus na nerazdvojive karakteristike u proizvodu, u nadi da kontrola tim faktorima rezultira unapređenjem proizvoda tokom korišćenja.
- U pogledu zasnovanom na vrednostima, kvalitet je spremnost kupca da plati softver.

Prema ISO standardu 8204 definicija kvaliteta softvera glasi: " Karakteristike jednog entiteta, koje imaju sposobnost da zadovolje formulisane i sadržane potrebe čine kvalitet softvera ", [7]. Kvalitetan softverski proizvod mora imati određene karakteristike koje su povezane sa zahtevima (korisnika) i zadovoljavaju te zahteve.

3. ISO STANDARD 9126

ISO standard 9126 [3] je danas najuticajniji u softverskom inženjerstvu. On obezbeđuje hijerarhijski okvir za definiciju kvaliteta, organizovanu kao karakteristike i podkarakteristike kvaliteta. Postoji šest karakteristika kvaliteta sa svojim podkarakteristikama:

- Funkcionalnost: Skup atributa koji se odnose na postojanje skupa funkcionalnosti i specifikiranih osobina. Funkcije su te koje zadovoljavaju određene ili sadržane potrebe. Podkarakteristike:
 - Podesnost
 - Preciznost
 - Interoperativnost
 - Bezbednost
- Pouzdanost: Skup atributa koji se odnose na sposobnost softvera da ostvari svoje nivoe performansi pod određenim uslovima za određeno vreme. Podkarakteristike:
 - Raspoloživost
 - Tolerisanje greške
 - Mogućnost oporavka
- Upotrebljivost: Skup osobina koje se odnose na korišćenje, na osnovu individualne procene korišćenja od strane korisnika. Podkarakteristike:
 - Razumljivost
 - Učivost
 - Operativnost
- Efikasnost: Skup atributa koji se odnose na vezu između nivoa performanse softvera i sume korišćenih resursa, u određenim uslovima. Podkarakteristike:
 - Ponašanje u vremenu
 - Ponašanje resursa

- Održavanje: Skup atributa koji se odnose na trud koji je potreban da se specifikuju modifikacije. Podkarakteristike:
 - Analiziranje
 - Promenljivost
 - Stabilnost
 - Testiranje
- Prenosivost: Skup atributa koji se odnose na sposobnost softvera da se prenosi iz jednog u drugo okruženje. Podkarakteristike:
 - Adaptivnost
 - Mogućnost instalacije
 - Koegzistencija
 - Zamenljivost

4. FAZE KVALITETA SOFTVERA

Iz perspektive razvoja softvera, kvalitet čine pet faza (stages), [9]:

Faza 1 (Stage 1) predstavlja osnovni kod kvaliteta: sintaksno i kodno konstruisanje.

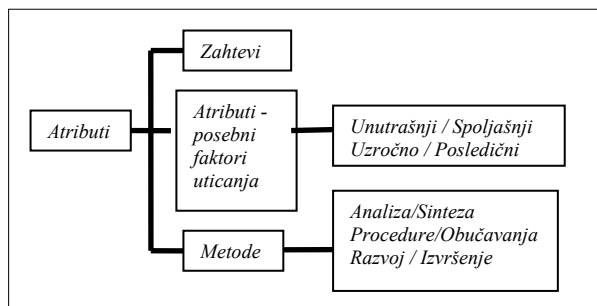
Faza 2 (Stage 2) predstavlja detaljan plan, opis posla: logičko konstruisanje programa i akcija potrebnih da bi ovi programi izvodili funkcije (ciljeve) radi kojih su i konstruisani.

Faza 3 (Stage 3) predstavlja plan na višem nivou: sistemska pitanja oko interfejsa, kompatibilnosti, učinka, bezbednosti i sigurnosti.

Faza 4 (Stage 4) predstavlja fokusiranje na zahtevima: određivanje i definisanje zahteva sa posebnim osvrtom na zaključivanje šta se traži između redova.

Faza 5 (Stage 5) predstavlja osvrt na korisnika i šta mora da se učini da bi im se obezbedio zaista kvalitetan proizvod. Iako je navedena kao poslednja faza, potrebe korisnika moraju imati najveći prioritet.

5. OPŠTA KLASIFIKACIJA ATRIBUTA KVALITETA



Slika 1: Opšta klasifikacija atributa kvaliteta, [6]

Klasifikacija atributa kvaliteta glasi, [6]:

Zahtevi – parametri po kojima se atributi sistema donose, specifikuju i ocenjuju.

Atributi kao posebni faktori – osobine sistema (kao što su polise i mehanizmi ugrađeni u sistem) i njihovo okruženje koje ima uticaj na zahteve. U zavisnosti od atributa, atributi

kao posebni faktori su unutrašnje ili spoljašnje odluke koje utiču na zahteve. Faktori ne moraju biti nezavisni i mogu da imaju uzročno – posledične veze. Faktori i njihove međusobne veze trebali bi da budu uvršteni u arhitekturu sistema. Među njima su:

1. Faktori učinka – aspekti sistema koji doprinose učinku. Ovo uključuje zahteve iz okoline i odgovor sistema na iste.
2. Oštećenja pouzdanosti – aspekti koji doprinose (nedostatku) pouzdanosti. Postoji uzročni lanac između nedostataka unutar sistema i propusta posmatranih u okolini. Nedostaci uzrokuju nastanak grešaka; greška je stanje sistema koje može dovesti do propusta ako se ne ispravi.
3. Faktori zaštite – aspekti sistema koji doprinose zaštiti. U te karakteristike spadaju sistem/oklina, interfejs, karakteristike i unutrašnje karakteristike kao što je temeljnost (kernelisation).
4. Oštećenja sigurnosti – aspekti sistema koji doprinose (nedostatku) sigurnosti. Rizici su uslovi ili stanja sistema koji mogu da vode do nezgoda ili nesrećnih slučajeva. Nezgode i nesrećni slučajevi su neplanirani događaji sa nepoželjnim posledicama.

Metode – kako se rukuje sa zahtevima: procesi analize i sinteze tokom razvoja sistema, procedure i obučavanja za korisnike i operatore. Metode mogu biti za analizu i/ili sintezu, procedure i/ili obučavanja, ili procedure koje se koriste u toku razvoja ili izvršenja.

6. VEZE IZMEĐU ATRIBUTA KVALITETA

Sistem često ne uspe da ispuni korisničke zahteve (nedostatak kvaliteta). To se dešava kada se dizajner usko fokusira na ispunjavanje određenih zahteva ne uzimajući u obzir druge zahteve ili uzimajući, ali prekasno u procesu razvoja. Na primer, moguće je da se istovremeno ne može ispuniti više zahteva. Ovo nije novi problem i ljudi koji se bave razvojem softvera pokušavaju sa njim da se izbore već duže vreme. O tome je pisao i Boehm: “ Najzad smo došli do zaključka da proračunavanje i razumevanje vrednosti jedne celokupne metrike za kvalitet softvera može stvarati više problema nego što to zaslužuje. Najveći problem je taj što su mnoge pojedinačne karakteristike kvaliteta u konfliktu; dodatna efikasnost često ima istu cenu kao prenosivost, tačnost, razumljivost, održivost; dodatna tačnost je često u sukobu sa prenosivošću preko zavisnosti iskazane veličine; konciznost može biti u konfliktu sa jasnoćom. Korisnici uglavnom smatraju da je teško odrediti prioritete u tako konfliktnim situacijama. “, [5].

7. CMM (CAPABILITY MATURITY MODEL)

Jedan od najpoznatijih modela za obezbeđenje kvaliteta je CMM (Capability Maturity Model). Ovaj model je nastao na institutu za softversko inženjerstvo na Carnegie Mellon univerzitetu, na inicijativu Ministarstva odbrane Sjedinjenih Američkih Država. Iako se alati i platforme tokom vremena menjaju, CMM je model koji još uvek ne odoleva vremenu. CMM se sastoji od 5 nivoa koji opisuju prednosti procesa u softverskom inženjerstvu. Četvrti nivo CMM modela se fokusira na kvalitet softvera.

1. U prvom nivou nema realnih procesa u celini.
2. Implementiranje procesa - Ovaj nivo uključuje zahteve menadžmenta, planiranje i praćenje softverskih projekata i formiranje menadžerskog tima.

3. Standardizacija procesa – Ovaj nivo zahteva upravljanje projektom, planiranje i praćenje izvršenja projekata pomoću organizacije. Prioritet se daje organizacionom procesu, infrastrukturi, obuci, itd.
4. Četvrti nivo je fokusiran na proizvod i proces kvaliteta softvera, sa mogućnošću merenja kvaliteta pri implementaciji projekta sa povratnom informacijom vođama timova koje im omogućavaju da vrše intervencije sa ciljem unapređenja procesa kvaliteta u organizaciji.
5. Unapređenje procesa - Na ovom nivou se sprečavaju greške u upravljanju softversko-tehnološkim promenama i promenama u procesu upravljanja. Najpre se po mogućstvu preventivno prave promene u vlastitom inženjerskosoftverskom okruženju, radi postizanja višeg nivoa efektivnosti i efikasnosti.

8. KONTROLA KVALITETA KROZ TESTIRANJE

Kvalitet softvera najviše zavisi od projekatata i programera, tj. vođa razvojnih timova. Oni definišu politiku izgradnje i testiranja određenog nivoa kvaliteta u različitim fazama razvoja softvera. Kontrola kvaliteta softvera je neophodna faza u životnom ciklusu softvera i treba je shvatiti kao preventivu. Pojam testiranja danas se razlikuje od vremena sedamdestih godina kada je to značilo isključivo traženje grešaka. Sada ono obuhvata niz aktivnosti od planiranja, dizajniranja, izgradnje, održavanja, pa sve do sprovođenja testova. Potrebno je definisati načine merenja pojedinačnih kvaliteta i definisati kriterijume prelaznosti. Prva faza u određivanju kontrole kvaliteta je faza projektovanja. Za sprovođenje već donete politike testiranja kvaliteta softvera postoje razni metodološki prilazi (struktuirani, objektno-orjentisani ili formalno-jezički orjentisani praćeni odgovarajućim case alatima). Testiranje je najefikasniji način za određivanje nivoa kvaliteta softverskog proizvoda. Bitno je da se testiranje posmatra kao jedna od faza u procesu razvoju softvera. Često se na testiranje gleda kao na sasvim odvojen, samostalan proces, što je greška. Najveći deo testiranja se obavlja od strane programera koji je razvio softver, ali da bi testiranje bilo potpuno, potrebno je obezbediti i testiranje od strane budućih korisnika. Samo na taj način se moguće greške najbrže pronalaze. Korisnik koji nije upoznat sa načinom izrade softvera neposrednom upotrebom će ukazati programeru na moguće greške. Prilikom testiranja nije moguće ispitati sve moguće akcije korisnika, nego se treba fokusirati na manji deo akcija. Zato je veoma bitno odrediti politiku testiranja: šta se testira, kojom brzinom će se vršiti testiranje, koliko različitih podataka treba unositi i sl. Od pomoći mogu biti i alati za testiranje određenih delova softvera, ali još nije pronađen alat za testiranje celokupnog softvera. Kvalitet korisničkog uputstva i uputstva za održavanje je mera u kojoj uputstva odražavaju stvarno funkcionisanje softvera, kao i mera u kojoj su informacije prikazane na razumljiv i upotrebljiv način. Ciljevi testiranja su različiti. Ogleđaju se kroz demonstraciju-proveru korektnosti implementacije softvera, destrukciju-detekciju grešaka na nivou programiranja, evaluaciju-otkrivanju grešaka u specifikaciji zahteva, dizajnu i implementaciji i prevenciju-sprećavanju grešaka kroz faze razvoja softvera. Iako testiranje kvaliteta softvera (softverskih komponenti) daje povratnu informaciju o upotrebljivosti softvera u realnim sistemima, kvalitet istog se ne može garantovati u potpunosti zbog delimićne nepoznanice koju sistem predstavlja kada se koristi u realnim sistemima. Jedan od načina provera kvaliteta softvera su i "beta tasteri", ljudi posebno obučeni za testiranje preliminarnih verzija programa.

9. ZAKLJUČAK

Kvalitet je veoma bitna faza u projektovanju i izradi proizvoda. Danas postoji pregršt različitih proizvoda na tržištu koji zadovoljavaju širok raspon potreba potrošača. Kako bi se dostigao željeni kvalitet pri izradi, najpre je potrebno informisati se o postojećim standardima za potrebnu oblast delovanja, treba ih poštovati, a onda i analizirati tržište. Uvođenje standarda znači postavljanje novih ciljeva, strategija i programa. Treba imati u vidu da kvalitet, sam po sebi nije dovoljan uslov za zagaranovanu implementaciju proizvoda u svakodnevicu potrošača, već da i marketing predstavlja potreban uslov da bi se potrošačima na najbolji način prezentovao proizvod kao konačan rezultat životnog ciklusa razvoja. Treba imati u vidu da se tehničke promene često dešavaju, pogotovo u eri digitalizacije i brzog prenosa informacija, pa se ponašati u skladu sa tim, i stalno poboljšavati kvalitet proizvoda. Kako ništa u prirodi nije savršeno, često se pronalaze inovacije ili nove mogućnosti za poboljšanje kvaliteta.

Kvalitet je trodimenzionalan fenomen. Prva dimenzija čini objektivni nadzor specifičnih zahteva, pomoću metoda i tehnika. Druga dimenzija je uslovljena odnosom proizvođač-korisnik. U ovoj dimenziji se dodaju subjektivni elementi (npr. zadovoljstvo korisnika). Treću dimenziju čini čovek sam, sa svojom sposobnošću da poboljša kvalitet. Karakteristike ove dimenzije su osmišljen rad, čovekovo samoostvarenje, [10].

Pri projektovanju i izradi softvera bitno je uvek imati na umu kvalitet. Kako on čini sintezu više faktora, svaki treba proučiti blagovremeno i temeljno.

Nije lako dostići kvalitet, pogotovo pri izradi softvera. Potrebno je dosta iskustva, truda, rada i zalaganja. Kvalitet softvera je kompleksno pitanje, koje se ne sme zaobići. Treba stalno postavljati pitanje: “ Kako poboljšati kvalitet ? “ .

Često je moguće ne postići očekivan uspeh. Kako bi se uspeh dostigao, potrebno je i mnogo vremena, pronicljivog razmišljanja, strpljenja i inspiracije. Bitan je individualan odnos prema kvalitetu, uspešno rešavanje zadataka, motivacija, inspiracija, postavljen sistem vrednosti, ideja i vizija.

Iako razvoj nekog softvera zahteva godine truda, to je ono na šta će programer biti ponosan, i zato treba težiti izgradnji, iznad svega, kvalitetnog softvera.

10. LITERATURA

- [1] Dragica Radosav, Softversko inženjerstvo I, Tehnički fakultet “Mihajlo Pupin“, Zrenjanin, 2005.
- [2] IEEE 610.12, IEEE Standard Glossary of Software Engineering Terminology, IEEE 1990.
- [3] ISO-9126, ISO standard, 2001.
- [4] Jeff Tian, Software quality engineering, IEEE Computer Society, New Jersey, 2005.
- [5] Mario R. Barbacci, Analyzing Quality Attributes, The Software Engineering Institute (SEI), Carnegie Mellon University, 1999.
- [6] Mario Barbacci Mark H. Klein Thomas A. Longstaff Charles B, Quality Attributes, Weinstock, December 1995.
- [7] Roland Petrasch, The Definition of Software Quality: A Practical Approach, FastAbstract ISSRE, 1999.
- [8] Watts S. Humphrey, A Personal Commitment to Software Quality, The Software

- Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA, 1994.
- [9] Watts S. Humphrey, A Personal Quality Strategy, The Software Engineering Institute (SEI),
 - [10] Carnegie Mellon University, 2005.
 - [11] Živoslav Adamović, Teorija sistema, Tehnički fakultet "Mihajlo Pupin", Zrenjanin, 2002.
 - [12] [http:// www.ekonomist.co.yu/magazin](http://www.ekonomist.co.yu/magazin)
 - [13] <http://www.hq.nasa.gov>
 - [14] [http:// www.chillarege.com](http://www.chillarege.com)
 - [15] <http://www.sei.cmu.edu>