

VERIFIKACIJA, FAZA U PROCESU OSIGURANJA KVALITETA SOFTVERA

Milorad Jovanović

Tehnički opitni centar VSCG, Vojvode Stepe 445, Beograd

Članak razmatra potrebu primene standardizovanih metoda razvoja i projektovanja softvera u cilju blagovremenog otkrivanja programskih grešaka i dobijanja kvalitetnog softvera. Izrada kvalitetnog softvera je složen proces koji zahteva poseban pristup u fazama upravljanja i kontrole. Primenom prihvaćenih metodologija i odgovarajućih standarda osigurava se zahtevani kvalitet i omogućuje dostizanje standarda kvaliteta čime se smanjuje rizik i ranjivost informacionih sistema na razne pokušaje zloupotrebe.

This article explain the need of appliance standardized software development and design methods to achieving high quality software without error and potential abuses. A software production is difficult process requesting specific approach in phases of management and control. Applyngi these these methodologies and standards requested quality can be assured and accepted standards of quality can be achieved which reduces risk and IT (information tehnology) systems vulnerability to diferent abuse attempts.

1. Uvod

Verifikacija softvera je proces potvrđivanja deklarisanog kvaliteta, performansi i sigurnosti. Ove karakteristike u tesnoj vezi su sa postojanjem grešaka u softveru. Nisu u pitanju materijali, komponente, starenje, radni uslovi i sl.

U pitanju su loša rešenja i pogrešne odluke planera, menadžera, programera i drugog osoblja koje na bilo koji način učestvuje procesu razvoja i projektovanja softvera.

Pored ovog, uzroci nedovoljne sigurnosti, informacionih sistema sa visokim rizikom su namerne greške odnosno programske zloupotrebe.

U analizi rizika, pri proceni verovatnoće nastajanja pretnje za funkcionisanje informacionog sistema, otkrivanje grešaka u fazama testiranja, namernih koji se mogu kategorisati u zloupotrebu i nenamernih koji su posledica nemara ili neznanja, tretira se kao jedan od osetljivih procesa u razvoju i projektovanju softvera.

U fazi tehničkog prijema pri verifikacija softvera, razmatraju se i sledeća dva aspekta: organizaciona i stručna sposobnost kompanije da osigura prihvatljiv nivo kvaliteta softvera. Razmatranje navedenih uslova osnov je za procenu rizika, verovatnoće manifestovanja grešaka i eventualne štete. U ove uslove u najširem smislu uključen je i aspekt bezbednosti.

2. Verifikacija – proces potvrđivanja kvaliteta

Proces verifikacije se izvodi kroz široka ispitivanja ispunjenja korisnikovih zahteva svrstanih u tri kategorije; kvalitet, performanse i sigurnost, koji se odnose na zahteve funkcionisanja i standardizovane karakteristike proizvoda.

Svaki od ovih zahteva predstavlja kompleks karakteristika skupa funkcija, attribute kvaliteta i elemente sigurnosti. Razmatranje mnoštva karakteristika iz navedenog, nameće zaključak da je verifikacija softvera u fazi tehničkog prijema obiman, usko specijalizovan i rizičan zadatak.

U slučajevima obimnih softvera za masovnu upotrebu, testiranje se vrši probnim eksploatacijama, distribuiranjem određenog broja primeraka reprezentativnom skupu test specijalista i potencijalnih korisnika.

Nameću se sledeća pitanja. Da li je to moguće u potpunosti izvršiti? Kako obezbediti prihvatljiv nivo kvaliteta i eliminisati odnosno na najmanju meru svesti rizik upotrebe?

Odgovori na navedna pitanja predstavljaju stručni izazov za IT menadžere, test-inženjere i specijaliste za IT sigurnost.

2.1 Elementi ocene kvaliteta

Kvalitet, performanse sigurnost

Prihvatljivi nivo kvaliteta podrazumeva zadovoljavajuću ocenu deklarisanog kvaliteta definisanih kroz karakteristike funkcionisanja, performanse i sigurnost softvera.

Postupak verifikacije i donošenje ocene zavisi od obima softvera, raznovrsnosti funkcija, širine domena ulaznih podataka i sl. U nekim slučajevima moguće je ispitati sva stanja softvera i u tom slučaju može se govoriti o potpunoj oceni završenosti softvera, uglavnom u sistemima u realnom vremenu, gde je broj funkcija i ulaznih podataka ograničen.[3]

U mnogim slučajevima se vrši i takozvano "beta" testiranje probnim eksploatacijama distribuiranjem softvera reprezentativnom skupu test specijalista i potencijalnih korisnika. Pored donošenja kvalitativnih i kvantitativnih ocena u procesu verifikacije zbog kompleksnosti softvera očekuje se otkrivanje preostalih grešaka, skrivenih mana i programskih zloupotreba.

Cilj verifikacije softvera je, dakle, konačna ocena o završenosti softvera, odnosno potvrđivanje sigurnosti sa određenim (procenjenim) rizikom.

Završenost softvera

Razvoj i projektovanje softvera treba da osiguraju zahtevani kvalitet tako da se nakon verifikacije može sa velikom verovatnoćom i minimalnim rizikom tvrditi da je softver završen odnosno da zadovoljava sve postavljene zahteve čime se potvrđuje da je raspoloživ u određenim uslovima i vremenu. Na taj način može se govoriti o stepenu **sigurnosti** softvera.

Greške i zloupotrebe u softveru

U zavisnosti od vrste, obima i načina funkcionisanja softvera ugrađuju se, manifestuju i otkrivaju nekoliko tipova softverskih grešaka.

Prema načinu ugradnje greške mogu biti :

- Aritmetičke
- Logičke
- Sintaksne i semantičke
- Greške u prenosu podataka

Osnovni predmet analize otkrivenih grešaka je njihov uzrok i moguće posledice. Da li su nenamerne ili namerne?

Ukoliko su nenamerne, obično su to grube greške i u većini se otkrivaju u procesu debugovanja. Namerne greške su vešto skrivene. Često imaju izgled nenamerne greške da u slučaju otkrivanja krivci ne budu optuženi.

Međutim, postoje i nenamerne greške koje opstaju i ne budu otkrivene do kritičnog trenutka, čak u odmaklim fazama eksploatacije. To je obično slučaj sa aritmetičkim greškama koje se manifestuju kada se složi nepovoljna kombinacija argumenata i tada dolazi do matematički neizvodljive operacije zbog čega, bez odgovarajućih mehanizama zaštite (rukovanje izuzecima, "watchdog timer"), IT sistem otkazuje.

Poznat je slučaj evropske svemirske rakete Ariana koja je eksplodirala 5 sekundi nakon lansiranja zbog aritmetičke greške prekoračenja vrednosti ("overflow") i zaokruživanja na celobrojnu vrednost.

Slučaj sa aritmetičkom greškom na Pentium procesoru, 1996. godine, prouzrokovao je velike finansijske štete +proizvođaču procesora. U još jednom poznatom slučaju kompjuterske greške, analize su pokazale da su uzroci aritmetičke prirode, kada je eksploziji američke "Patriot" rakete u Saudijskoj Arabijskoj rakete poginulo 26 vojnika.

Prema nameni i uslovima korišćenja, namerne greške se ispoljavaju u jednom od oblika klasifikovanih kao modus operandi (nadmudrivanje, "seckana salama", trojanski konj, vremenska bomba itd.)

Greške se po pravilu otkrivaju detaljnim ispitivanjem takozvanim dokazivanjem korektnosti programa ali se za otkrivanje namernih grešaka programerskih zloupotreba zahteva posebna pažnja i tehnika analize.

2.2 Ocena rizika

Verifikacija uključuje proces detaljnog testiranja koji se sprovodi dokazivanjem korektnosti, izvođenjem odabranog uzorka test primera a u cilju otkrivanja svih kategorija grešaka u softveru, u poslednjoj fazi razvoja i projektovanja, pre puštanja u eksploataciju.

Posledice manifestovanja grešaka u eksploataciji mnogo su skuplje od opsežnih ispitivanja u toku razvoja i projektovanja. "Otkrivanje" grešaka u eksploataciji može prouzrokovati štetu, gubitke u određenim primenama čak i tragične posledice.

Složenost i "apstraktnost" softvera čine verifikaciju zamašnim poslom. Postoji, potreba provere na nivou svake linije programa, na nivou instrukcije, da bi se izbegla pojava greške u bilo kojoj situaciji i stanju softvera. Detaljno ispitivanje omogućuje procenu verovatnoće pojave greške (f). Taj podatak sa podatkom o veličini moguće štete(g), kao proizvod, mera je procenjenog rizika ($r = f \cdot g$) [4].

2.3 Metode poboljšanja softvera

Praksa pokazuje, da bi se u dovoljnoj meri osigurao kvalitet i sigurnost softvera, odnosno rizik smanjio na prihvatljiv nivo, potrebno je preduzeti organizovane mere za sprečavanje ugradnje grešaka i zloupotreba i njihovo otkrivanje vršiti u svim fazama razvoja i projektovanja. Ove mere su propisane standardima za osiguranje kvaliteta softvera. U zavisnosti od namene softvera primenjuju se odgovarajući standardi ali ih karakteriše jedinstvena tehnologija projektovanja (sl.1)

3. Proces osiguranja kvaliteta softvera

Proces osiguranja kvaliteta softvera podrazumeva jasnu i prihvaćenu metodologiju razvoja i projektovanja a u skladu sa specifikacijom, opštim zahtevima kvaliteta i ugovorima. Ovaj proces sprovodi se planiranim aktivnostima prema principima i pravilima usvojenog modela projektovanja. Verifikacija i tehnički prijem je poslednja faza projektovanja, pre uvođenja u eksploataciju.

Metodološki pravilno razvoj i projektovanje stvara uslove za ostvarenje definisanih zahteva kvaliteta koje se realizuju po sledećim fazama:

- Specifikacija zahteva kvaliteta projektovanja, isporuke i održavanja
- Ugovoranje
- Projektovanje
- Prijemno ispitivanje
- Eksploatacija i održavanje

3.1 Specifikacija zahteva kvaliteta, isporuke i održavanja

Pri specifikaciji zahteva koji se ugovaraju za softver primarna su tri aspekta koji su definisani sledećim standardima:

- kvalitet (standard ISO/IEC 9126 - 1996),
- performanse ISO/IEC 14756 - 1999
- sigurnost - serija standarda ISO/IEC

Standardima za kvalitet softvera definisani su atributi softvera kao što su funkcionalnost, međuoperativnost, prenosivost, sigurnost itd. Razmatranje sigurnosti kao jednog od atributa kvaliteta softvera u pojedinim namenama ima poseban značaj.

3.2 Ugovaranje

Izbor modela projektovanja

Ugovorom pored zahteva kvaliteta, performansi i sigurnosti, treba definisati model projektovanja. To zavisi od vrste softvera koji se razvija. Ukoliko složenost softvera zahteva opsežna prijemna ispitivanja ili je namena osetljive prirode (finansijski, obaveštajni, odbrambeni) stvaranje uslova za učešće korisnika u svim fazama projektovanja bitan je uslov za izbor modela.

Standardi osiguranja kvaliteta

Izbor modela upravljanja projektima određen je standardima za osiguranja kvaliteta. Jedan od prvih standarda koji su propisali osiguranje kvaliteta softvera u razvoju i projektovanju su MIL-S-52779 A i DOD-STD-2167. Pored ovih standarda koji su vezani za odbrambene tehnologije, za civilnu upotrebu dugo nije bilo adekvatnih standarda koji bi propisali uslove kvaliteta, opšte i posebne zahteve za softver, raznih namena.

Od 1995. godine usvojeni su ISO/IEC standardi koji detaljnije regulišu upravljanje projektima za obezbeđenje kvaliteta softvera. Primarni značaj u projektovanju softvera ima standard ISO/IEC/IEA/IEEE 12207-1995. Sigurnosti informacionih sistema i softvera je regulisana posebnim standardima. U ovoj oblasti danas su uglavnom u upotrebi ISO/IEC standardi iz serije TR 13335, ISO/IEC 15408 i ISO/IEC 9000-3.

Tehnička dokumentacija

Značajan aspekt osiguranja kvaliteta softvera je dobro definisan ugovor kupca i isporučioaca softvera u pogledu tehničke dokumentacija. Osim prethodno pravilno definisanih zahteva i uslova isto tako važan aspekt je kompletnost i razumljivost tehničke dokumentacije softvera koja treba da obezbedi uslove za verifikaciju i održavanje softvera.

Standardi koji regulišu kvalitet dokumentacije softvera ISO/IEC 9127, ASTM E919-96 itd. polaze od principa otvorenosti koda, kompletnosti i razumljivosti dokumentacije u svim nivoima programskog sistema. To podrazumeva opis svih programskih zadataka, poslova i procesa. Time se programeri obavezuju na otvorenost, štite od svoje zaboravnosti i sprečavaju da u programska rešenja uvode neautorizovane "inovacije".

Tehnološki uslovi, kao što su principi projektovanja ("top-down", modularno), principi programiranja (objektno-orijentisano, struktuirano), izbor programskog jezika i sl., takođe su značajni za obezbeđenje kvaliteta softvera.

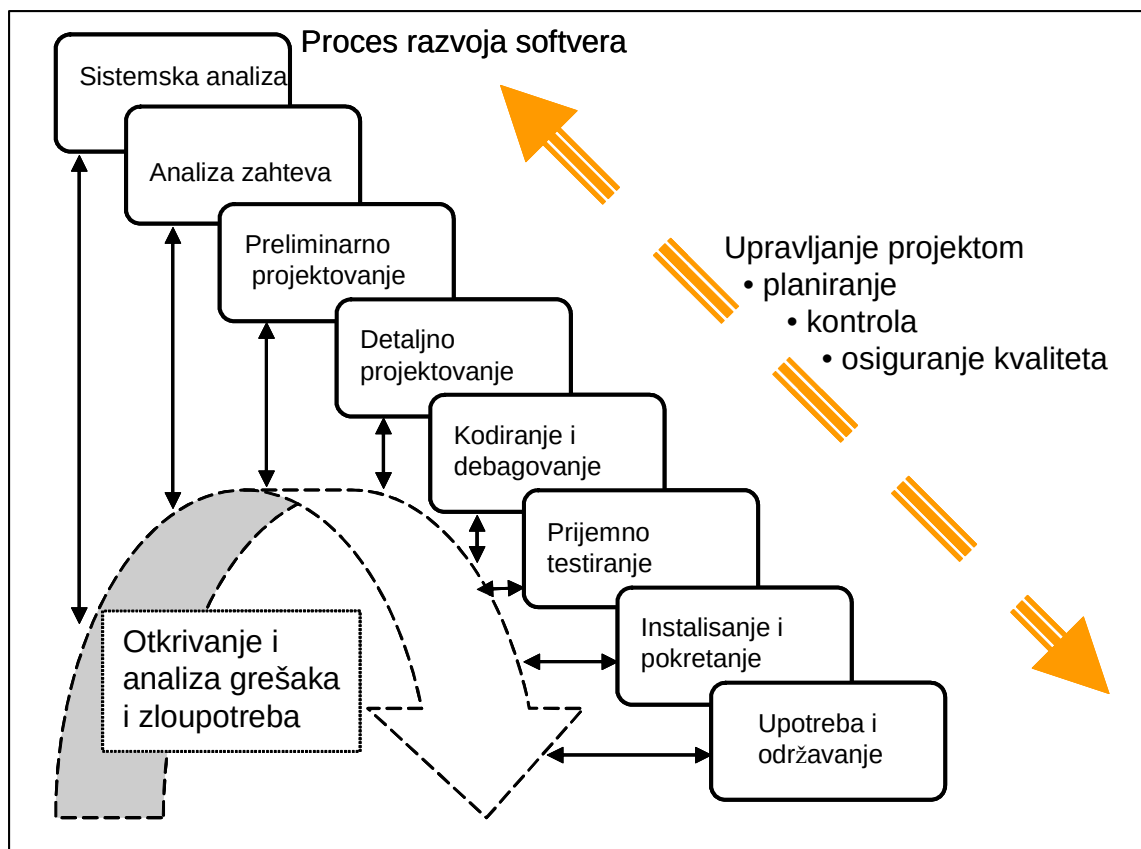
3.3 Projektovanje

Projektovanje se sprovodi prema principima softverskog inženjeringa u sledećim fazama:

- analiza zahteva za sistem
- definisanje sistemskih zahteva
- preliminarno projektovanje
- detaljno projektovanje
- kodiranje i testiranje modula
- integracija komponenata i testiranje
- integracija i testiranje podsistema
- integracija sistema i testiranje

Bez obzira na primenu automatizovane opreme za programiranje i testiranje u svim fazama projektovanja postoji velika verovatnoća "potkradanja" greške u softveru. Zbog toga postoji potreba za vrlo opsežnim i detaljnim testiranjem i stalnim preispitivanjem nalaza.

Na slici 1. prikazan je model koji predstavlja karakterističan model procesa razvoja softvera koji ukazuje na važnost testiranja koji je prisutno u svim fazama razvoja.



Sl. 1. Faze projektovanja softvera

3.4 Testiranje - dokazivanje korektnosti softvera

Menadžeri u projektovanju softvera i test inženjeri korišćenjem odgovarajuće opreme testiraju svako moguće stanje softvera kao automata i u skladu sa tim i svaki programski modul odnosno liniju koda.

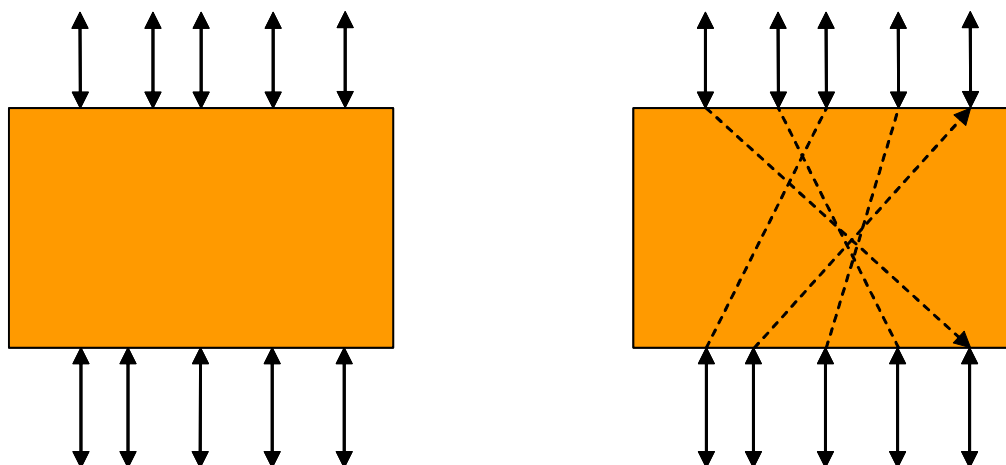
Pored semantičke i sintaksne analize, koji su predmet debugovanja vrši se provera funkcija na nižim nivoima u uslovima razvojnih sistema i emulacije.

Na taj način moguće je izvršiti analizu programskog koda i dobiti nalaze za ocenu završenosti softvera, i postojanja grešaka u softveru.

Izvođenje korektnosti dokaza od početnih izjava do izveštaja je vrlo složen proces. Današnje stanje metoda verifikacije softvera je na višem nivou razvoja u odnosu na programiranje.

Testiranje softvera prema prihvaćenim opštim metodama projektovanja predstavlja važnu aktivnost koja se realizuje u svim fazama. U toku razvoja primenjuju se principi dokazivanja korektnosti na nivou instrukcija, modula i programskih delova.

Tehnike pribavljanja dokaza o korektnosti povezane su sa fazom projektovanja počev od matematičkog modela i dijagrama toka. U narednim fazama, programiranje, kodiranje integracije i implementacije koriste druge metode i tehnike. [1]



Sl.2. Informaciono skrivanje i otkrivanje

Apstraktnost softvera povezana je i sa njegovom enkapsulacijom u okviru razvoja "nezavisnih" modula i izvornog koda. Test-inženjeri i analitičari moraju biti u mogućnosti da razumeju svaki programski modul i njegov uticaj na ostale module i softver u celini. Prihvatajući princip otvorenosti koda, standardi za kvalitet postavljaju principe koji se tumače i na nivou celine i delova, kao što su moduli, programi i potprogrami, rutine. Podrazumeva se da otvorenost koda bude obezbeđena detaljnim opisom funkcije svakog modula, podacima koji ulaze u obradu i koji se "vraćaju", opisom tipova podataka, konstanti, promenljivih itd.

Zbog toga je važno da ispitivanje softvera od strane ovlašćenih organa i organizacija započne već u ranim fazama projektovanja. Cilj testiranja ispitivanja na svim nivoima softvera od sistemskog do nivoa modula i programske linije.

Kvalitet tehničke dokumentacije

Ugovorom definisan kvalitet tehničke dokumentacije, podrazumeva pristup i pregled od strane korisnika odnosno ovlašćenih organa, kompletne softverske dokumentacije i opreme za razvoj i projektovanje.

Kvalitetnim dokumentovanjem, otkrivanjem sadržaja i opisa modula mora se obezbediti vidljivost svih organizacionih, tehničkih i programskih procesa, koje čini mogućim otkrivanje svih vrsta grešaka i zloupotreba i njihovo jednostavno ispravljanje odnosno kompletno izvršenje misije testiranja.

3.5 Uslovi održavanja u eksploataciji

Verifikacijom softvera ocenjuju se elementi integralno-tehničkog obezbeđenja odnosno osposobljenost nosioca održavanja da u toku celog životnog veka softvera obezbedi uslove za korekcije, dorade i ispravljanje naknadno utvrđenih grešaka ili potencijalnih zloupotreba. Ugovorom se obaveze u pogledu održavanja definišu na taj način da potrebna oprema i dokumenti budu raspoloživi u predviđenom veku softvera.

4. Zaključak

Potvrđivanje ugovorenih i opštih **zahteva za** softver je zadatak koji treba sprovesti. Verifikacija softvera koja se vrši u fazi tehničkog prijema, zbog njegove složenosti, obima i apstraktnosti predstavlja kompleksan proces koji zahteva obimne analize i ispitivanja. Kao deo procesa osiguranja kvaliteta deo verifikacionih aktivnosti bi trebalo planirati u ranim fazama razvoja i projektovanja softvera što bi smanjilo obim poslova u završnim ispitivanjima tehničkog prijema. To takođe omogućuje test-specijalistima potpuno

poznavanje softvera, rano otkrivanje grešaka i zloupotreba, povećanje kvaliteta i smanjenje rizika sigurnosti softvera, skraćanje vremena verifikacije u fazi tehničkog prijema itd.

Kvalitet i sigurnost softvera koji je verifikovan za datu namenu i određene uslove, sprovođenjem opisane metodologije, na taj način može se oceniti sa minimalnim rizikom.

Literatura:

1. Andersen, R. B.: *Proving Programs Corerect*, John Wiley & Sons, New York, 1979.
2. Gogates, G.G. *Software Validation Accreditaion Laboratories- A Practical Guide*, Fasor In, - www.fasor.com.
3. Jovanović M.: *Metoda automatskog merenja tačnosti mikroračunara laboratorijskom računarom*, JUREMA-1989, Zagreb-Plitvička jezera
4. Myers, G.I.: *The Art og Software testing*, John Wiley & Sons, New York, 1981.
5. Petrović R.S.: *Kompjuterski kriminal*, MUP Republike Srbije, 2000.
6. Shrivastava S.K.: *Reliable Computer Systems*, Springer-Verlag, Berlin, Heidelberg, 1985.
7. Youll, D.P.: *Achieving ultra high quality software*, *Quality Assurance*[115-120] Vol.14, No.3, 1988.