

# TESTIRANJE SOFTVERSKIH PROIZVODOV KAKO NAPREJ?

Marjan Pivka, Vojko Potočan  
Ekonomsko-poslovna fakulteta, Maribor

## Povzetek

Popularni modeli ocenjevanja softvera (CMM, BOOTSTRAP, SPICE, ISO 9000) ne upoštevajo vpliva certificiranja softverskih proizvodov na kakovost softvera. Prvi standard za kakovost softverskih proizvodov je nemški DIN 66285. Na njegovi osnovi je ISO razvil mednarodni standard ISO/IEC 12119, ki določa zahteve za kakovost in testne procedure za softverske pakete. V tekstu predstavljamo naše izkušnje z klasičnim testiranjem (na osnovi ISO/IEC 12119, DIN 66285) in možna izhodišča za učinkovitejše in uspešnejše testiranje. Realizacijo teh ciljev omogoča oblikovanje in uporaba kooperativnega modela testiranja, ki temelji na aktivnem sodelovanju in prenosu znanja med vsemi udeleženci testiranja.

## Abstract

*Popular models of software testing (CMM, BOOTSTRAP, SPICE, ISO 9000) do not take into account the impact of software certification upon software quality. The first standard for software products quality has been the German DIN 66285 standard. On its basis ISO developed the international standard ISO/IEC 12119, determining requirements for software packages quality and test procedures. In the paper we are presenting our experience with classical testing (following ISO/IEC 12119, DIN 66285) and possible guidelines for more efficient and effective testing. Realization of those objectives is made possible by designing and usage of the cooperative model of testing, introducing active cooperation and knowledge transfer among all the participants.*

**Ključne besede:** Kakovost softvera; Modeli testiranja; ISO/IEC 12119; Kooperativni model testiranja.



## 1 OPREDELITEV PROBLEMA

Ocenjevanje, vrednotenje in izboljševanje procesa izdelave softvera je danes znano po ameriškem CMM modelu, evropskem BOOTSTRAP modelu, ter mednarodnih SPICE in ISO 9000 modelih.

Vsi navedni modeli bazirajo na splošno znanem dejstvu, da je boljše kakovost proizvodov možno doseči le z izboljšavami v procesu izdelave tega proizvoda. Njihova uporaba pa ne zagotavlja, da bo programska oprema, izdelana v posameznem procesnem okolju, ki je na primer na tretjem zrelostnem nivoju po CMM modelu, skladna s specifičnimi zahtevami zakonodaje, varnostnimi zahtevami, ipd.

Uporabniki programskih proizvodov večkrat potrebujejo potrditev, da je programski proizvod (poslovni programski paket, varnostni softver, življensko pomembni softver, itd) skladen z določenimi predpisi ali z zakonodajo, da ustreza določenim varnostnim zahtevam, standardom ipd.

Testiranje delovanja programskega proizvoda na skladnost z zakonodajo ali drugimi specifičnimi zah-

tevami je možna le, če obstajajo standardi oziroma predpisi za njihovo testiranje. Vendar ocenitveni modeli ne dajejo neposrednega odgovora na vprašanje, ali je programski proizvod skladen s specifičnimi zahtevami uporabnika.

Poznanih je več kot dvesto standardov kakovosti, ki se nanašajo na softver. Večina njih je nacionalnih oz. mednarodnih (ANSI, BSI, DIN standardi) ali profesionalnih (IEEE standardi, obrambni standardi, ...). Namenjeni so za uporabo na področjih kot so: management, zagotavljanje kakovosti, upravljanje konfiguracije, varnost, opredelitev zahtev, verifikacija, validacija, itd.).

Za softverske proizvode obstajata le dva mednarodna standarda: ISO/IEC 12119:1994 Information Technology - Software Packages - Quality Requirements and Testing in ISO/IEC 9126:1991 Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use.

Testiranje programskega proizvoda na skladnost s standardi, z zakonodajo ali določenimi predpisi, ima



nedvomno vpliv na proces izdelave softvera in s tem dolgoročno na izboljšanje kakovosti celotnega dela proizvajalca softvera. V tem prispevku obravnavamo model testiranja programskih proizvodov, s katerim preverjamo ne le skladnost proizvoda, ampak dosežemo tudi povečanje kakovosti dela proizvajalca s prenosom znanja od testnega laboratorija na proizvajalca.

## 2 ISO/IEC 12119

ISO/IEC 12119 daje navodila za testiranje softverskih paketov kot so: urejevalniki besedil, preglednice, programi za vodenje baz podatkov, grafični paketi, programi za tehnične ali znanstvene funkcije, softverski paketi za poslovni informacijski sistem ipd.

Standard vključuje zahteve za kakovost softverskih proizvodov in navodila, kako moramo testirati softverske proizvode. Sistem kakovosti proizvajalca softverskega proizvoda ni predmet standarda ISO/IEC 12119.

Zahteve za kakovost, ki izhajajo iz standarda, so:

1. Opis proizvoda. Vsak softverski proizvod mora imeti opis, ki je sestavni del dokumentacije proizvoda. Opis proizvoda vsebuje njegovo kratko predstavitev: navedba funkcij, območja delovanja, zahtevano strojno in programsko okolje, primernost uporabe ipd.
2. Vsak softverski proizvod mora imeti celovito, korektno in konsistentno uporabniško dokumentacijo. Ta mora vključevati vse potrebne informacije o inštalaciji, uporabi in vzdrževanju softverskega proizvoda.
3. Programi in podatki. Zahteve opredeljujejo šest karakteristik kakovosti, ki so definirani v standardu ISO/IEC 9126. Za funkcionalnost lahko na primer opredelimo naslednje zahteve: inštalacija, celovitost funkcij (prisotnost vseh funkcij), korektnost in konsistentnost.

ISO/IEC 12119 daje navodila za testiranje programskega paketa. Opis programskega proizvoda in dokumentacija se testirata z metodo inšpekcije, testiranje programov in podatkov pa z metodo testiranja črne škatle.

Klasičen način takega testiranja predpostavlja, da je programski proizvod končan in da testni laboratorij testira proizvod, ki je pripravljen za tržišče, ali pa je že na tržišču. Rezultat takega testiranja je poročilo o testiranju, ki pove, ali je programski proizvod skladen, ali pa ne, z zahtevami standarda in vseh predpisov, ki se na ta proizvod nanašajo.

V praksi smo pri takem testiranju ugotovili naslednje:

- Testni laboratorij mora sam izdelati testne podatke za testiranje, zato je procedura testiranja časovno zahtevna in draga;
- Testni laboratorij mora poznati področje, ki ga

pokriva programski proizvod. Na primer: pakete za gradbeništvo lahko testirajo le timi, v katerih so tudi eksperti iz področja gradbeništva. To zagotavlja visoko raven kakovosti testiranja ne le programov in podatkov, ampak tudi uporabniških vmesnikov in dokumentacije;

- Testni laboratorij nima nobenega neposrednega vpliva na proces izdelave tega proizvoda;
- Omogočen je omejen prenos znanja od testnega laboratorija na dobavitelja;
- Tržišče (na strani uporabnikov in na strani dobaviteljev) je pod močnim vplivom ISO 9000 skupine standardov in drugih ocenitvenih modelov in zato ne pozna pomena certifikatov kakovosti programskih proizvodov;
- Raziskovalna sfera (univerze, raziskovalni laboratoriji) certificiranju programskih proizvodov ne posveča pozornosti.

Zaradi teh razlogov klasični model testiranja programskih paketov na skladnost z zahtevami ISO/IEC 12119, v praksi ni tako popularen, kakor omenjeni ocenitveni modeli. Razlogi za to so po našem mnenju v klasičnem modelu testiranja, ki zahteva, da že opravljeno delo (integralno testiranje, preverjanje skladnosti dokumentacije) izvedeta tako dobavitelj (predpostavka), kot testni laboratorij (obveza).

## 3 KOOPERATIVNI MODEL TESTIRANJA

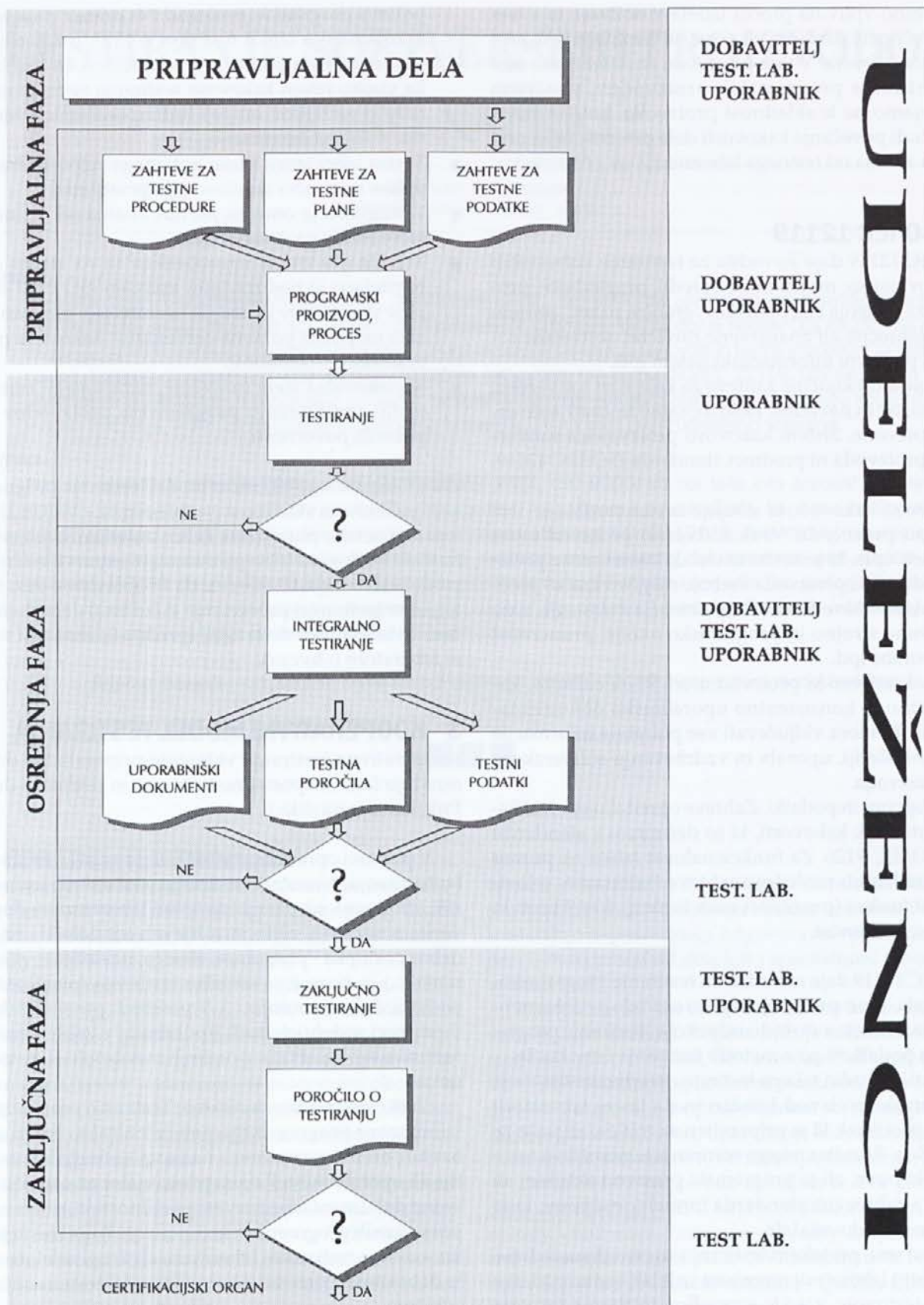
Kooperativno testiranje vključuje pripravljajna dela, osrednjo fazo - neposredno testiranje in zaključna dela. Prikazano je na sliki 1.

V pripravi opredelimo izhodišča in osnove za izvedbo testiranja. Naročnik testiranja (dobavitelj, uporabnik, strokovna združenja) in testni laboratorij se dogovorita o namenu, ciljih in zahtevah ter določita način (izbor postopka - procedure, obsega, udeležencev testiranja, ...) in pogoje za izvedbo testiranja (plani, testni podatki, dokumentacija, ...). V posebnih primerih lahko v pripravi sodelujejo tudi uporabniki z zahtevami za testiranje ter določijo posebne in željene ciljev testiranja.

Sledi osrednja faza testiranja. Testiranje posameznih komponent programskega paketa običajno izvaja dobavitelj brez neposrednega nadzora testnega laboratorija ali uporabnikov. V njem proizvajalec na osnovi razvojne dokumentacije preveri pravilnosti in ustreznost posameznih programskih komponent. V unit testu lahko sodeluje tudi testni laboratorij ali drugi udeleženci, ki dobavitelju posredujejo specifične zahteve, znanja in izkušnje.

Po končanem testiranju komponent nadaljujemo z integralnim testiranjem. Integralno testiranje izvaja





SLIKA 1: KOOPERATIVNI MODEL TESTIRANJA



dobavitelj na osnovi zahtev in pogojev testnega laboratorija. Testni laboratorij izvaja funkcijo kontrole nad delom proizvajalca neposredno z nadzorom pri izvajalcu ali posredno s preverjanjem dokumentacije testiranja. S prenosom znanja vpliva na izboljšanje dela proizvajalca, kar omogoča povečanje kakovosti obravnavanega in bodočih proizvodov. Drugi udeleženci lahko na integracijski test vplivajo s svojimi izkušnjami in predlogi.

Osrednjo fazo zaključimo s končnim testiranjem, ki ga izvede testni laboratorij na osnovi zahtev standardov in pogodbeno dogovorjenih obveznosti. Delo vključuje metodološko, vsebinsko in formalno presojo ustreznosti. Testirajo se naključno izbrane funkcije, mejne vrednosti, uporabniški vmesniki, uporabniška dokumentacija, robustnost, itd.. Testni laboratorij lahko v svoje delo vključi tudi uporabnike ali druge udeležence testiranja, da bi zagotovil skladnost delovanja softvera s potrebami in zahtevami okolja.

V zaključni fazi testni laboratorij izdela poročilo o testiranju, ki je osnova za odločanje certifikacijskega organa (podelitev certifikata, zavrnitev podelitve). Na posebno zahtevo proizvajalca lahko oblikuje še druge predloge in mnenja.

Predstavljene značilnosti kooperativnega modela testiranja lahko strnemo v naslednje ugotovitve.

Kooperativni model testiranja ni v nasprotju z zahtevami mednarodnega standarda ISO/IEC 12119. Uporabljamo ga za preverjanje skladnosti programskega paketa z zahtevami standardov in kot osnovo za vrednotenje - validacijo programskih paketov. Novo vrednost tega modela pa vidimo v vplivu na kakovost procesa izdelave programskega paketa proizvajalca.

Testiranje v testnem laboratoriju ni izolirana dejavnost, temveč je vključeno v proces izdelave programskega proizvoda. Testiranje torej pomeni eno od faz življenjskega cikla procesa izdelave programskega paketa, ki poteka od razvoja pri proizvajalcu do njegove uporabe.

Testiranje poteka v več fazah, kar zagotavlja hitro reagiranje na ugotovljena odstopanja. Z ažurnimi korekcijami skrajšamo potrebni čas in obseg dela, kar omogoča večjo racionalnost testiranja.

Pri njegovem izvajanju aktivno sodelujejo testni laboratorij, proizvajalec in uporabniki oz. drugi udeleženci, ki so zainteresirani za kakovost softvera (strokovna združenja, revizijske hiše, svetovalna podjetja, ...). Kakovost testiranja je odvisna od ravni organizacijske razvitosti, udeležencev testiranja in prenosa znanja med njimi.

Posamezni udeleženci testiranja razpolagajo s specifičnimi znanji. Testni laboratorij posreduje dobaviteljem in uporabnikom znanja o metodiki testiranja. Dobavitelj posreduje testnemu laboratoriju in uporab-

nikom znanja o tehnologiji in tehniki izdelave softvera. Uporabniki oz. drugi udeleženci testiranja posredujejo dobavitelju in testnemu laboratoriju specifična strokovna znanja in praktične delovne izkušnje.

Kooperativno sodelovanje udeležencev testiranja v procesu prenosa znanj zagotavlja sinergijske rezultate, ki vplivajo na ekonomiko in kakovost testiranja ter procesa izdelave softvera.

#### 4 SPOZNANJA O UPORABI KOOPERATIVNEGA MODELA TESTIRANJA

Naša spoznanja in praktične izkušnje z uporabo kooperativnega modela testiranja lahko uvrstimo v tri skupine.

##### Prvič: Ciljni programski produkt je v razvoju.

Realnost (točnost) planiranja testiranja je odvisna od organizacijske razvitosti dobavitelja. V procesu razvoja se programski produkt nenehno spreminja, kar zahteva stalno prilagajanje planov testiranja in testiranja, tako na strani dobavitelja, kakor pri testnem laboratoriju. Zaradi razvoja je dokumentacija nepopolna, na razpolago ni dovolj testnih podatkov ali pa so ti takšni, da ne omogočajo realnega vsebinskega testiranja produkta. Ugotovili smo, da nizka razvitost razvojnega okolja dobavitelja zahteva večjo angažiranost testnega laboratorija, ki mora s povečanim nadzorom in prenosom znanja zagotoviti ustrezno raven testiranja.

##### Drugič: Izvajanje testiranja.

Druga skupina spoznanj se nanaša na izvajanje testiranja in vodenje konfiguracije programskega paketa v razvoju. Pri praktičnem delu smo spoznali klasične probleme testiranja brez vodenja konfiguracije: testiranje že testiranih modulov, iste napake se pojavljajo v različnih verzijah modulov, dokumentacija ne sledi razvoju in testni podatki niso na razpolago v ustreznem obsegu in kakovosti.

##### Tretjič: sodelovanje in komuniciranje.

V tretjo skupino uvrščamo spoznanja o sodelovanju in komuniciranju med udeleženci testiranja. Med testnim laboratorijem in dobaviteljem mora vladati odnos medsebojnega spoštovanja in zaupanja. Le na ta način je reagiranje pri ugotovljenih odstopanjih ažurno, manj konfliktno ter ni odvisno od odvečnih formalnosti. Pri testiranju namreč lahko pride do situacije, ko programske napake ni mogoče povsem natančno dokumentirati. Reševanje takih situacij je mogoče le z medsebojnim sodelovanjem in zaupanjem. Zelo pomemben je tudi odnos med testnim laboratorijem in uporabniki (pogodbenimi, potencialnimi). Njihovo znanje, zahteve ali želje so pomemben prispevek h kakovosti testiranja.



## 5 SKLEPNE UGOTOVITVE

Opisani kooperativni model uteleša sistemski način razumevanja življenjskega cikla softvera kot celovitega procesa, ki vključuje vse dejavnosti od uvodnih raziskav od prenehanja njegove uporabe. Uporaba sistemskega pristopa omogoča spoznanje pomembnih soodvisnosti ter sinergijskega delovanja posameznih aktivnosti v procesu, kar velja tudi za testiranje.

V kooperativnem modelu temeljijo vloge posameznih udeležencev testiranja in njihovi medsebojni odnosi na kreativnem sodelovanju in medsebojni izmenjavi znanja.

Prednosti kooperativnega modela so v celoti vidne pri testiranju programskih proizvodov v razvoju. V praksi smo spoznali, da je najpomembnejši problem raven organizacijske in tehnološke razvitosti dobavitelja.

Organizacijska nerazvitost se kaže kot neustrezna definiranost življenjskega cikla dobavitelja. Zaradi tega dobavitelj ne more ustrezno izvajati in dokumentirati vseh razvojnih faz, zlasti pa ne faz testiranja.

Tehnološka razvitost je običajno višja od organizacijske (na primer uporaba 4GL), vendar brez ustrezne organizacijske podpore ne omogoča doseganja pričakovanih rezultatov.

### **Naša praktična spoznanja lahko tako strnemo v naslednje:**

Testni laboratorij mora v pripravljalni fazi oceniti raven razvitosti dobaviteljevega procesa razvoja programske opreme. To lahko ugotovi s presojo dobavitelja z izbranim modelom (ISO9001, CMM, BOOTSTRAP, AMI,...).

Način sodelovanja in nadzor nad dobaviteljem je odvisen od njegove ravni razvitosti. Če je dobavitelj na začetni ravni razvoja (CMM model), je potreben večji obseg prenosa znanja in dela testnega laboratorija. Prenos znanja se nanaša na življenjski cikel, vodenje projektov, vodenje konfiguracije, metode testiranja etc.

Uporabniki lahko svoje pripombe in predloge posredujejo dobavitelju ali testnemu laboratoriju. Če

gre za želje, zahteve ali mnenja, ki so opredeljeni v standardu, odloča o njihovem sprejetju testni laboratorij. Če pa gre za vsebinske zahteve, pa odloča dobavitelj. Ugotovili smo tudi, da je v nekaterih primerih možna rešitev le s konsenzom vseh treh strank.

Kooperativni model testiranja zahteva manjšo porabo virov za testiranje neglede na raven razvitosti dobavitelja. Večja učinkovitost in prenos znanja vplivata neposredno na kakovost procesa gradnje programske opreme in s tem na uspešnost kooperativnega modela testiranja.

## LITERATURA

DIN 66285:1990 Anwendungssoftware. Gütebedingungen und Prüfbestimmungen. Buch Verlag Berlin 1990.

ISO 9000-3:1991 Quality Management and Quality Assurance Standards - Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software.

ISO/IEC 9126: 1991 Information Technology - Software product evaluation - Quality characteristics and guidelines for their use.

ISO/IEC 12119:1995 Information Technology - Software packages - Quality requirements and testing.

Lindermeier, R. Softwarequalität und Softwareprüfung. R. Oldenbourg Verlag München Wien 1993.

Pivka, M. 1995 Software quality system in a small software house. Proceeding of Software Quality Management 1995 Vol.1. Computational Mechanics Publication.

Paulk, et. al: M. Paulk, Bill Curtis, M.B. Chriss&C Webber, Capability Maturity Model, Version 1.1. IEEE Software, July 1993, pp 18-27.

Root, P.T. SPICE: A Framework for Software Process Assessment Software Process. Improvement and Practices. August 1995. Pp 57-66.

*Dr. Marjan Pivka se z informatiko ukvarja več kot dvajset let. Bil je programer in sistemski programer, vodil je projektne time in računalniške centre. Od leta 1986 je zaposlen na Ekonomsko-poslovni fakulteti v Mariboru, kjer predava Informacijsko tehnologijo, Organiziranje podatkov in Podatkovne strukture. V zadnjih osmih letih se na raziskovalnem in aplikativnem področju ukvarja predvsem s problematiko kakovosti ne le v programskem inženirstvu, ampak tudi širše. V tem okviru raziskuje sisteme kakovosti (standardi družine ISO 9000) ter preskušanje in certificiranje programskih proizvodov. Deluje tudi kot presojevalec sistemov kakovosti pri Slovenskem inštitutu za kakovost in meroslovje.*

*Mag. Vojko Potočan se ukvarja z organizacijo in informatiko deset let. Študij na EPF je končal leta 1987, leta 1993 pa je magistriral iz področja Poslovne organizacije. Med leti 1987 in 1994 je delal v podjetjih kot sistemski analitik, organizator in vodja projektnih timov. Od leta 1994 je zaposlen na Ekonomsko-poslovni fakulteti kot raziskovalec. V zadnjih petih letih se na raziskovalnem in aplikativnem področju ukvarja s problematiko splošnega managementa, informacijskega managementa in metodoloških osnov za poslovno odločanje.*