Vzdrževanje programja – mit ali resničnost?

Niko Schlamberger, Slovensko društvo INFORMATIKA
niko.schlamberger@gmail.com

Izvleček

Prispevek obravnava problematiko, ki je med informatiki znana kot vzdrževanje programja. Sporni so izrazito nasprotujoči si pogledi ponudnikov in uporabnikov programskih produktov. V končni posledici gre za kredibilnost ponudnikov in za finančne učinke na eni in na drugi strani. Argumentacija je odvisna od gledišča, torej ponudniškega ali uporabniškega. Namen prispevka je izpostaviti dilemo, ali je definicija vzdrževanja programja primerna, in potem priti do kolikor toliko objektivnega pogleda na problematiko. Priporočilo je, naj se namesto izraza vzdrževanje programja raje uporabi izraz obratovanje programja, ki je vsebinsko bolj korekten in vsestransko manj zavajajoč.

Ključne besede: vzdrževanje programja, upravljanje programja, obratovanje informacijskega sistema.

Abstract

Software Maintenance – Myth or Reality?

The paper addresses the issue professionally known as software maintenance. The views of software providers and those of users tend to be contradictory, and the key antagonists are the credibility of providers on the one hand and the financial resources needed for such services on the other hand. The debate depends on the perspective where users and providers find it hard to accept each other's arguments. The purpose of the paper is, firstly, to discuss the suitability of the current definition of software maintenance as a professional activity, and secondly to provide a relatively impartial view of the problem. The recommendation is to use the terms software management or software *operation* instead, because both seem closer to the intended meaning and less misleading.

Key words: software maintenance, software management, information system management.

1 UVOD

Zamisel za ta prispevek se je porodila že pred časom. V teku izvedbe nekega javnega razpisa[1,2] se je izkazalo, da vzdrževanja programja naročniki in ponudniki ne razumejo enako. Ponudniki si ga razlagajo v svoje dobro in predvsem kot utemeljitev pavšalnega plačila, ki ga zaračunavajo naročnikom programja, naročniki pa pojmujejo vzdrževanje kot storitev, ki mora imeti opredeljene veličine (predmet storitve, cena, rok izvedbe, rezultat), da je mogoče sploh ugotavljati, ali so za plačilo dobili, kar so naročili. Navadno je posledica različnega razumevanja vzdrževanja programja slab občutek na obeh straneh: ponudniki so mnenja, da njihovo delo ni primerno cenjeno, naročniki pa mislijo, da plačujejo storitev, ki ni opravljena. Ta prispevek naj predvsem pripomore k razjasnitvi pojmov ter poenotenju razumevanja in s tem pomaga ustvariti izhodišče za korektne poslovne odnose.

Avtor ne skriva svojega stališča, da pri programju v klasičnem pomenu pojava ni kaj vzdrževati (in posledično tudi ne zaračunavati). To tezo skuša zagovarjati in dokazati, razen tega pa ponuditi še jasno in bolj sprejemljivo podlago za urejanje tovrstnih odnosov med naročniki in ponudniki vzdrževanja programja. Seveda to ne pomeni, da ponudnik ali razvijalec po opravljeni storitvi (dobavi programske rešitve ali končanem razvoju) nima s tem nobenega dela več. Nasprotno, dela in posledično storitev, ki jih je mogoče in tudi treba zaračunati, je še na pretek, bistveno pa je, da uporabnik ali naročnik ve, kaj plača in kaj dobi za to. Namen prispevka je torej predvsem urediti podlage: opredeliti vzdrževanje programja, ustvariti izhodišče za boljše vzajemno razumevanje in pokazati, kaj je smotrno ali potrebno zaračunavati ali plačevati in kaj ne.

Pogosto se izkaže, da skušajo ponudniki vsaj del cene razvoja programja, bodisi da je izdelano po naročilu ali za neznanega kupca, kompenzirati z zaraču-

[1] Konkurenčni dialog med ministrstvom za zdravje in potencialnimi ponudniki informacijskega sistema nove pediatrične klinike v Ljubljani.

[2] V teku izbire ponudnika je nastal interni zapis [13], da bi pripomogel k ureditvi pojmov glede vzdrževanja in spremljajočih storitev.

navanjem storitev, ki jih opredelijo kot vzdrževanje, in se tako skušajo prikazati kot ugodnejši ponudnik, seveda na račun neke vrste odloženega plačila, ki bi ga pridobili čez čas prek tako imenovanega vzdrževanja. To nasploh in praviloma ponujajo kot celoto, torej v paketu, ki obsega garancijo, nameščanje programja, dopolnitve, pomoč uporabnikom, podporo uporabnikom in še kaj. Posledica je nepregledna in za uporabnike po njihovem mnenju praviloma preplačana storitev. Nasploh se vzdrževanje programja uporablja – da ne rečemo zlorablja – kot podlaga za pogosto nesorazmerno visoko zaračunavanje storitev uporabnikom, ki pogosto po odločitvi za določenega ponudnika programja nimajo več preproste poti za spremembo svoje odločitve in so ponudniku prepuščeni na milost in nemilost. Temu poglavju je treba zato posvetiti posebno pozornost ob sklepanju poslovnega odnosa, saj je vse drugo verjetno dražje in gotovo napornejše.

Za ilustracijo, kako vzdrževanje programja ponujajo in opisujejo ponudniki takih storitev, sta navedena primera, prevzeta z javno dostopnih domačih strani gospodarskih družb, katerih dejavnost je računalništvo ali informatika, namenoma pa vir ni naveden, da ne bi po nepotrebnem izpostavljali ponudnikov ali trpeli očitka, da oglašujemo njihove storitve:

- vzdrževanje standardne programske opreme delovnih postaj (MS Windows in MS Office) na lokaciji naročnika,
- vzdrževanje sistemske programske opreme MS Windows 2003 strežnik (aktivni imenik, datotečni strežnik).

Zaman bi iskali vsaj opis, kaj je vzdrževanje in kaj obsega. Pristop je približno takle: vsakdo ve, kaj je vzdrževanje; če to potrebujete, pač naročite. Pogledano s stališča naročnika je zadeva bistveno drugačna. Predvsem naročnik praviloma ni informatik, saj če bi bil, te storitve verjetno ne bi naročal; ima določeno težavo, ki ga ovira pri delu ali poslovanju in bi jo rad odpravil; potrebuje pomoč, mogoče bi storitev naročil, ne ve pa, kaj bo obsegala in koliko ga bo stala. Seveda gornjega primera ne moremo posplošiti, vendar je po ogledu množice tovrstnih ponudb mogoče dokaj meritorno skleniti, da je primer tipičen za odnos naročnik – ponudnik.[1] Presenetljivo pa je, da se opredelitve vzdrževanja programja vzdržijo tudi učbeniki. Primeri ([1], [2], [4]) kažejo, da avtorji kljub obetajočemu naslovu ne omenijo vzdrževanja programja, ne potrudijo se niti z definicijo vzdrževanja [sistemske programske opreme], kaj šele, da bi jo vsaj poskusili sistematično obravnavati. Ponovno naj pripomnimo, da nekaj primerov ne moremo posplošiti, vendar tudi dlje časa trajajoče iskanje ni dalo pričakovanega rezultata, torej vsaj definicije vzdrževanja programja. Če je to spregledano že v institucionalnem izobraževalnem sistemu, se ne moremo čuditi, da je v poslovnem svetu to področje še bolj zamegljeno in neurejeno.

2 VZDRŽEVANJE PROGRAMJA

2.1 Nekaj zgodovine in teorije

Da so računalniški programi obremenjeni z vsemi atributi izdelkov človeških rok in glav, je gotovo. Posledica je, da so zato tudi pri programju potrebni raznovrstni posegi za odpravo pomanjkljivosti ali za boljše delovanje. Ne da bi o tem kaj dosti razmišljali vsaj v začetnem obdobju široke rabe računalnikov, to je nekako v šestdesetih letih prejšnjega stoletja, se je za vse vrste posegov v programje začel uporabljati izraz *vzdrževanje* (v angl. *maintenance*). Izraz je bil domač, splošno razumljiv in pri roki, zato se je prijel ne glede na to, da v zvezi s programjem ni ustrezen, kar je pojasnjeno v nadaljevanju tega prispevka. Programerji so v programe posegali iz povsem praktičnih razlogov in konkretnih potreb; konec koncev je vsak program namenjen izvajanju povsem določene funkcije ali postopka in če se pri njegovem delovanju izkažejo pomanjkljivosti iz bodisi internih ali eksternih razlogov, jih je treba odpraviti z ustreznimi posegi. Z razvojem informatike se je s tem področjem začela ukvarjati tudi akademska sfera in ga opremila z nekoliko teorije ter z malo več klasifikacije, sistematike in metode. Vsaj od osemdesetih let prejšnjega stoletja naprej je začela nastajati o vzdrževanju programja strokovna in znanstvena literatura, ki je to dejavnost uredila do te mere, da je postala pregledna, obvladljiva in – enako kot programiranje sámo – predmet jasno opredeljenih in strokovno utemeljenih aktivnosti in postopkov. Znotraj informatike in računalništva se je razvila v še eno tehnično disciplino.

Precej gradiva o vzdrževanju programja lahko najdemo že kar na Wikipedii [15], med drugim tudi definicijo vzdrževanja programja: v informatiki je

[1] Avtor je pridobil tudi nekaj pogodb med ponudniki in naročniki, vendar jih v virih in referencah ne navaja, ker bi vsaj nekatere navedbe pomenile kršitev poslovne skrivnosti.

vzdrževanje programja po dobavi prilagoditev programskega produkta za odpravo napak, izboljšanje delovanja ali drugih lastnosti. S to problematiko so se sistematično ukvarjali poleg praktikov že vsaj od osemdesetih let prejšnjega stoletja tudi raziskovalci (npr. [16], [17], [18], [19]). Njihovi in drugi prispevki so rezultirali v preglednem, napovedljivem in obvladljivem o procesu vseh vrst posegov v programje, vidni pa so tudi v nacionalnih in mednarodnih standardih, kot so npr. ISO/IEC/IEEE 14764, ISO 9120, ISO 20000, ISO/IEC 14764:2006, ISO 27000. Rezultati teh prizadevanj so pozitivni, saj so utemeljili sistematičen pristop k opravljanju tovrstnih aktivnosti in jih povezali tudi s kakovostjo izdelkov, storitev in njihovega razvijanja.

2.2 Definicije

Namen tega prispevka ni enciklopedična obravnava vseh razpoložljivih definicij vzdrževanja programja, zato se omejuje le na zgoraj podane kot tipične in take, ki izkazujejo, da glede vzdrževanja programja še ni bila izrečena zadnja beseda. Posledica je, da sive ali celo bele lise na tem terenu izrabljajo tisti, ki to morejo in znajo, vse prepogosto v škodo tistih, ki to plačujejo. Seveda ambicija tega prispevka ne more biti, da bi enkrat za vselej razrešil vse dileme in nejasnosti, upati pa je, da bo vsaj skromen prispevek k boljšemu razumevanju med naročniki in ponudniki tovrstnih storitev.

Če naj bo obravnavana tematike obdelana sistematično, je treba najprej definirati vzdrževanje in programje in potem po možnosti na osnovi teh dveh ponuditi še definicijo vzdrževanja programja. Za klasiko razvrščanja in definicije vzdrževanja programja bi lahko šteli delo Swansona in Lientza [19]. Prvi je opredelil tri tipe vzdrževalnih posegov: za odpravo napak (korektivno), za prilagoditve (adaptivno) in izboljševalno. Drugi je te tipe posegov definiral in dodal še četrti tip – preventivno (preprečevalno) vzdrževanje in sicer:

- korektivno vzdrževanje so prilagoditve programja, ki se izvajajo po dobavi zato, da se odpravijo zaznani problemi;
- adaptivno vzdrževanje so prilagoditve programja, ki se izvajajo po dobavi zato, da je uporabno v spremenjenem okolju;
- izboljševalno vzdrževanje so posegi v programje po dobavi za izboljšanje delovanja ali možnosti vzdrževanja;
- preventivno vzdrževanje so prilagoditve programja po dobavi z namenom odkriti in odpraviti skrite napake v njem, preden začnejo učinkovati.

V vseh štirih definicijah se pojavi omejitev (*po dobavi*), ki pa je nepotrebna, saj o vzdrževanju ne more biti govor prej, preden kateri koli produkt, torej tudi programje, ne začne opravljati funkcije, za katero je bil razvit in izdelan. Definicije tudi kažejo na to, da so izhodišče zanje še vedno izdelki materialne narave in ne upoštevajo nematerialne narave programja. Zlasti za četrto se zdi, da povsem spregleda delo Jamesa Martina iz osemdesetih let prejšnjega stoletja o dokazljivo pravilnem načrtovanju programov in posledično tudi o dokazljivo pravilnem programiranju. Skrite napake so lahko le pomanjkljivost razvijalcev in nimajo nič opraviti z vzdrževanjem.

M. Krisper kot eden redkih domačih avtorjev, ki je vzdrževanje programja tudi opredelil, ga definira takole: *Vzdrževanje [programja] je faza, ki se po vpeljavi programske opreme v uporabo stalno odvija in je sestavni del življenjskega cikla programske opreme*.[2] Navaja tudi vrsto standardov in metodologij, ki obravnavajo to materijo in se dotakne odnosa med kakovostjo programja in vzdrževanjem le-tega ter namena in vrst posegov v različne tipe programja. Tej definiciji bi mogli očitati, da vzdrževanja programja ne opredeljuje kot aktivnost, temveč ohlapneje kot fazo v razvojnem ciklu programja.

Običajno pojmovanje vzdrževanja podaja Slovar slovenskega knjižnega jezika [7]: *Vzdrževati pomeni delati, da se kaj ohranja v dobrem stanju*. Za namen tega prispevka je treba biti natančnejši: *Vzdrževanje je celota postopkov in posegov, ki so potrebni, da ostane funkcionalnost izdelka, naprave ali tehničnega sistema taka, kakršna je bila v trenutku, ko je bil izdelan ali prevzet v uporabo*.[3] Ta definicija je splošna ne glede na vrsto, funkcijo ali namen *izdelka*, naprave ali *tehničnega sistema* in ne glede na to, kako, iz česa in kdo je to izdelal.

Programje je v Islovarju [6] opredeljeno kot *sistemski in uporabniški programi, ki so del računalniškega sistema – aplikacijski, servisni, sistemski, razvijalski.* Naš prispevek se nanaša na vse vrste programja, saj obravnava programje kot enovito kategorijo ne glede na namen in način uporabe. Gre namreč za entiteto z bistvenimi skupnimi značilnostmi.

---

[2] Krisper, Marjan: Vzdrževanje programske opreme (predstavitev), posvet Vzdrževanje programske opreme, Ljubljana, 15. 3. 2013.

[3] Schlamberger, Niko: Vzdrževanje programja – mit ali resničnost? (predstavitev), posvet Vzdrževanje programske opreme, Ljubljana, 15. 3. 2013.

Splošno sprejete definicije vzdrževanja programja pri nas nismo zasledili. Vse v uvodu navedeno pa dokazuje potrebo po sprejemljivem in konsistentnem pristopu. Treba je pripomniti, da obstoj definicij za *vzdrževanje* in *programje* sam po sebi še ne pomeni, da je kombinacija obeh izrazov, torej *vzdrževanje programja*, tudi smiselna ali da kot kategorija sploh obstaja.

2.3 Računalniški program kot tehnična naprava

Računalniške programe in različne njihove pojavne oblike (sistemsko programje, razvojna orodja, servisne programe in uporabniške programske rešitve) lahko obravnavamo kot tehnične naprave, saj opravljajo določeno nalogo ali funkcijo, so sestavljeni iz komponent in izdelani po pravilih stroke, v našem primeru informatike. Čeprav je programje tehnična naprava, je vendar treba upoštevati določene posebnosti. Predvsem računalniški program ni materialne narave. Pri uporabi (delovanju) programov za razliko od materialnih naprav zato ne prihaja do obrabe, staranja materiala, loma in podobnega, zato tudi ni potrebno vzdrževanje v klasičnem pomenu besede in v smislu na začetku podane definicije, to je táko, ki bi zahtevalo zamenjavo obrabljenih, pokvarjenih ali polomljenih delov in komponent naprave. Upoštevaje dejstvo, da programi podpirajo poslovanje, ki je zaradi sprememb v tehnologiji, okolju ali organizaciji dela podvrženo spremembam, mora temu slediti tudi ustrezno spreminjanje programskih rešitev, kar se v zelo širokem smislu pogosto razume in opredeljuje kot vzdrževanje, vendar je primerneje posege v zvezi s programskimi rešitvami poimenovati drugače, in sicer tako da upoštevamo njihove inherentne posebnosti.

3 VRSTE POSEGOV V PROGRAMJE

Potem ko naročnik pridobi program in ga začne uporabljati, se pokaže, ali je izdelan kakovostno. Kakovosten je programski produkt, ki a) ustreza specifikacijam, b) deluje zanesljivo in c) optimalno izkorišča vire [9]. V tem primeru nas predvsem zanima druga lastnost – ali deluje zanesljivo. Programje, v katerem so napake, praviloma ne more delovati zanesljivo, zato je kot pri vsaki napravi tudi pri programju ponudnik za zagotavljanje kakovosti zavezan z garancijo. V nadaljevanju so opisani bistveni koraki v procesu uvajanja programja, ki imajo vplive na raznovrstne posege v programje.

3.1 Namestitev

Potem ko je izbrano ali izdelano in preverjeno programje ustrezno, to je delujoče na predviden način, ga je treba namestiti na ustrezno računalniško napravo v okolju, kjer bo pripravljeno in razpoložljivo za uporabo. Namestitev lahko opravi ponudnik, naročnik pa mora zagotoviti ustrezno okolje (naprave, sistemske programe, komunikacije idr.). Specifikacijo okolja navadno poda ponudnik, lahko pa tudi naročnik, če je to posebna zahteva, ki vpliva na razvoj in uporabo programja.

3.2 Garancija

Garancija (jamstvo) obsega zavezo ponudnika, da bo izdelek določen čas ob predvidenem načinu in namenu uporabe (navadno eno leto) deloval brezhibno in skladno s specifikacijami.[4] Garancija je vključena v ceno izdelka in se zato ne plačuje posebej. Vse stroške, ki jih ima proizvajalec v zvezi z izpolnjevanjem obveznosti iz naslova garancije, pokriva kupnina ali (tudi v primeru programja) cena razvoja proizvoda. Pravica naročnika je, da v garancijski dobi lahko vrne proizvod in dobi povrnjeno kupnino, da zahteva, naj proizvajalec defektni proizvod brezplačno zamenja ali pa da proizvajalec odpravi napako na svoje stroške. Garancija lahko, ne pa tudi nujno, vključuje odpravo skritih napak, to je takih, ki jih ni bilo mogoče ugotoviti ob nakupu ali prevzemu, v času garancije ali tudi po izteku garancijske dobe.

3.2.1 Odpravljanje programskih napak

Kot programske napake razumemo delovanje programja, ki ni skladno s specifikacijami, in druge napake, ki ne vplivajo na funkcionalnost proizvoda. V času garancijske dobe jih odpravi proizvajalec na svoje stroške, zamenja proizvod ali vrne kupnino. Pri tem je smotrno določiti odzivne čase za izvajanje intervencije, ki pa ne morejo vključevati časa odprave napak. Pri odpravi programskih napak je težko opredeliti vnaprej, v kakšnem času bo napaka odpravljena, razmeroma lahko pa je določiti odzivni čas, saj je odvisen predvsem od poslovne in strokovne sposobnosti ponudnika ali razvijalca, ne nazadnje pa tudi od njegovega odnosa do določene stranke ali do poslovnih partnerjev nasploh.

[4] V primeru programja je časovna omejitev zaradi nematerialne narave izdelka neprimerna, se pa verjetno uporablja iz tradicionalnih, deloma pa tudi iz komercialnih razlogov.

3.2.2 Skrite napake

Skrite napake so napake, ki jih na način, kakor je bil proizvod preizkušen v času izdelave ali razvoja ni bilo mogoče ugotoviti, se pa pokažejo kasneje ob njegovi uporabi kot nepredviden način delovanja, ki ima za posledico omejeno funkcionalnost ali celo neuporabnost izdelka.[5] Skrite napake v programju so praviloma znak pomanjkljivosti procesa razvoja ali nestrokovnih razvijalcev ali obojega in ne bi smele bremeniti uporabnika. Lahko se pokažejo v času garancijske dobe ali kasneje, strošek njihove odprave po izteku garancijske dobe pa je stvar dogovora med proizvajalcem in kupcem. To je tudi razumljivo, saj skrite napaka kažejo na nestrokovno ali slabo izdelavo, pa naj gre za fizične naprave ali programje. Praksa gre vse bolj v smer zaščite naročnika, to je v smer odprave skritih napak na stroške izvajalca.[6] Pri programskih rešitvah je smiselno zahtevati odpravo skritih napak na stroške razvijalca, saj mora imeti urejen in kontroliran proces razvoja programskih rešitev in s tem minimalno tveganje za nastanek skritih napak.[7] Pri tem je podobno kot pri odpravi napak razumno določiti odzivne čase za izvajanje intervencije.

3.3 Pomoč uporabnikom[8]

Uporabniki programja imajo lahko pri uporabi le-tega težave, ki izvirajo iz subjektivnih ali objektivnih vzrokov, odražajo pa se kot nezmožnost uporabe programske rešitve. Ob predpostavki, da je izdelek brezhiben, je v interesu kupca in ponudnika (ali razvijalca), da pomaga uporabnikom pri delu s programom. Običajen način je večnivojsko organizirana storitev, ki triažira in svetuje, kako ravnati, da bo uporabnik lahko nadaljeval delo. Prvo raven pomoči je smiselno organizirati v okolju uporabnika (to je v njegovi delovni organizaciji), ker praksa kaže, da se večina potrebnih intervencij zreducira na nasvet uporabniku, manj pa je takih, da narekujejo bolj zapletene in strokovno zahtevnejše prijeme. Pomoč lahko zagotovijo ustrezne službe naročnika v kombinaciji s ponudnikom programske rešitve, lahko pa jo izvaja tretja oseba samostojno. V prvem primeru, to je pomoč v domačem okolju, je strošek zagotavljanja pomoči impliciten, sicer pa ne. Pomembna kakovost pomoči so kratki odzivni časi.

3.4 Podpora uporabnikom[9]

Podpora se nanaša na pomoč za odpravo težav v okolju programske rešitve (ob pogoju brezhibne programske rešitve) in ne na težave pri uporabi programa. Težava pri uporabi programja je lahko napaka programa ali stanje v okolju, v katerem je nameščen program (naprave, omrežje, komunikacije ipd.) ali v katerem deluje programska rešitev. Uporabniki programske rešitve ob nastopu težav pri njeni uporabi praviloma ne vedo, kaj je njihov vzrok. To ugotavlja ustrezna raven pomoči, ki tudi začne postopek za ustrezno intervencijo. Podporo lahko zagotovijo ustrezne službe naročnika v kombinaciji s ponudnikom ali z razvijalcem programske rešitve ali tretje osebe, podobno kot v primeru pomoči uporabnikom.

3.5 Dopolnjevanje programskih rešitev

Vsaka programska rešitev je informacijska preslikava določene funkcije, dela poslovanja ali postopka. S časom se okoliščine delovanja spreminjajo in v najbolj radikalnem primeru se je treba zateči k reinženiringu. Razlogi za reinženiring so znani [10]: a) bistvena sprememba funkcije (poslanstva), b) bistvena sprememba okolja ali c) bistvena sprememba obnašanja. V primeru manjših sprememb je mogoče in primerno reagirati s prilagoditvami (dopolnitvami in spremembami) programja. Alternativna možnost je prilagoditev funkcije, okolja ali obnašanja, kar niti ni vselej izvedljivo, skoraj vedno pa je dražje kakor prilagoditve programja.

Funkcionalnosti programja so povezane z organizacijo dela, predpisi, doktrino, pravili stroke, zahtevami okolja ipd. Iz različnih razlogov se lahko zato pojavijo potrebe za dopolnjevanje ali spremembe funkcionalnosti, katerih razlog niso napake v programih, temveč so posledica prej naštetih dejavnikov. Dopolnjevanja programov ne moremo razvrstiti niti kot vzdrževanje, niti kot garancijsko obveznost razvijalca, niti kot pomoč ali podporo, temveč kot povsem novo zahtevo za razvoj nove ali dopolnitev funkcio-

---

[5] Neuporabnost je bistveno merilo, saj npr. nepredvidenih dodatnih funkcionalnosti, ki jih specifikacija izvirno ne predvideva in ne povzročijo, da izdelek ne bi bil uporaben, naročnik praviloma ne bo obravnaval kot napake.

[6] Paradigma je avtomobilska industrija, v kateri lahko kupce določenega modela tovarna pozove, naj pripeljejo vozilo (ne glede na iztek garancijske dobe) na brezplačno zamenjavo določenega dela, ki je bil spoznan kot neprimeren za varno uporabo vozila. Podobno je tudi pri vestnih ponudnikih programov, ki pošiljajo nove verzije ali popravke obstoječih strankam brezplačno.

[7] Naj pri tem ponovno opozorimo, da je J. Martin že pred več kot dvema desetletjema razvil doktrino dokazljivo pravilnega programiranja, ki pa se le redko omenja in, kot kaže, žal še redkeje prakticira.

[8] Izraza pomoč in podpora sta sicer do neke mere vsebinsko sorodna, sta pa uporabljena v različnem pomenu prav zaradi tega, da je mogoče jasno opredeliti različne storitve, ki jih obsegata.

[9] Gl. opombo 8.

nalnosti obstoječe programske rešitve. Tako zahtevo je treba obravnavati v smislu novega odnosa med naročnikom in razvijalcem. Tudi cena realizacije take zahteve mora biti predmet posebnega dogovora.

3.8 Usposabljanje[10]

Uporaba programskih rešitev praviloma zahteva od uporabnikov določene veščine in znanja, ki so poleg poznavanja vsebine dela in delovnih postopkov potrebna za to, da jih lahko uporabljajo pri svojem delu na predvideni način. To so npr. osnovno razumevanje okolja programske rešitve (naprav, komunikacij ipd.), obvladovanje pristopa do programske rešitve, poznavanje njenih funkcionalnosti, njeni vplivi na delo uporabnika, pa tudi splošne veščine dela z računalnikom (vklapljanje naprav, aktiviranje programov, prenosi podatkov, tiskanje ipd.). Praviloma uporabnike usposablja za uporabo določene programske rešitve razvijalec ali ponudnik te programske rešitve, priporočljivo pa je sodelovanje naročnika.

4 POVZETEK, DISKUSIJA IN PREDLOGI

Po Islovarju [6] povzeta definicija opredeljuje le funkcijo informacijskega sistema, ne pa tudi njegove sestave in zgradbe. Zato jo je treba za namen in potrebo tega prispevka razširiti vsaj toliko, da pojasnimo njegovo sestavo. Sestavljajo ga naprave, programje, pravila in postopki, prispevek pa obravnava le v smislu vzdrževanja po avtorjevem prepričanju sporni del, to je programje. Glede naprav – računalnikov, perifernih naprav, pomnilnikov, napeljav – ni nobene dileme: kot vse druge naprave so tudi te podvržene staranju, obrabi, strojelomu, okvaram in jih je treba vzdrževati v klasičnem pomenu tega izraza.

Različni pogledi na vzdrževanja programja in finančne posledice le-tega so pogost vzrok nesporazumov med naročniki in ponudniki programja. Glavni argument ponudnikov, da vzdrževanje zaračunavajo, je, da so za vse vrste posegov v programje potrebni informatiki, njihova plača pa je za delodajalca strošek, ki ga je treba nekako pokriti. Če gre za programje po naročilu, ki je praviloma bistveno dražje od komercialno razpoložljivega programja za neznanega naročnika, je iz konkurenčnih razlogov skoraj neizvedljivo, da bi ponudnik v ceno razvoja vključil še stroške intervencij za določeno obdobje uporabe programja v prihodnosti. Programje za neznanega naročnika je cenejše, vendar iz različnih razlogov tudi potrebuje dopolnitve ali prilagoditve, ki jih morajo opraviti informatiki, in smo spet pri razmisleku z začetka tega poglavja. Na kratko bi stališče ponudnikov lahko povzeli takole: svojih informatikov ne moremo imeti v zamrzovalniku in jih odtajati takrat, ko jih bo kdo potreboval.

Z vidika naročnikov je stanje drugačno: programje, ki je dokazano brez napak in ki deluje tako, kakor določajo specifikacije, se pri uporabi ne obrablja in ne kvari in je zato strošek za vzdrževanje nesprejemljiv. Dodatno je moteče še to, da se za vzdrževanje praviloma zaračuna pavšal ne glede na to, ali je bilo v obračunskem obdobju iz tega naslova kaj dela, torej tudi stroškov, ali ne. Na kratko bi stališče naročnikov lahko povzeli takole: imamo občutek, da plačujemo nekaj, česar ne dobimo.

Kako ti dve diametralno nasprotni stališči vsaj zbližati, če že ne poenotiti? Podlaga za to je razumevanje med naročniki in ponudniki. Argumentom druge strani nobena stran ne oporeka, nesporazum je v formulaciji in obrazložitvi storitve. Prvi korak k večjemu razumevanju je to, da prenehamo govoriti o vzdrževanju. Nesporno je, da so posegi v programje potrebni tako rekoč ves čas njegove uporabe, nesporno pa je tudi, da ne gre za enako vzdrževanje kot pri drugih tehničnih napravah. Primernejši izraz je *upravljanje programja*. To obsega vse vrste, oblike in načine posegov v programje, ki so potrebni, da opravlja svojo funkcijo ves čas uporabe. Izraz upravljanje je na prvi pogled nenavaden, vendar ni nesprejemljiv, saj je bistveno širši in predvsem ni zavajajoč. Uporabljajo ga celo v bistveno bolj eksotičnih kontekstih, npr. v povezavi z rjavim medvedom [8]. Če lahko upravljamo z njim, ki je živo bitje, ali ne bi še laže upravljali s programjem? Verjetno bolj priporočljiv predlog pa je uporabljati izraz *obratovanje programja* (če se omejimo le nanj) ali pa *obratovanje informacijskih sistemov*, ki implicira tudi širše okolje, v katerem deluje programje. Na prvi pogled je predlog nenavaden, saj na ta izraz v povezavi s programjem ali operacijskimi sistemi, kolikor je avtorju poznano, ne naletimo. Če pa nekoliko pogledamo okoli sebe, vidimo, da je *obratovanje* splošno priznan in tudi uporabljen izraz. V Standardni klasifikaciji dejavnosti SKD 2008 [14] srečamo ta iz-

[10] Pogosto, če ne celo praviloma, se v tej zvezi uporablja izraz izobraževanje, ki pa je neprimeren. Namen usposabljanja je pridobivanje veščin, ki so potrebne za opravljanje določenega dela (tipičen tak primer je vozniško dovoljenje) in se izvaja zunaj institucionaliziranega sistema izobraževanja ali ob njem. Izobraževanje je institucionalizirano in se izvaja v nacionalnem izobraževalnem sistemu, ki ga sestavljajo javne in zasebne izobraževalne ustanove različnih ravni, stopenj in programov. Prim. tudi slovensko Standardno klasifikacijo dejavnosti (SKD 2008), ki je nacionalni standard in ima področje izobraževanja in usposabljanja zelo dobro urejeno.

raz v zvezi z najrazličnejšimi sistemi in napravami – rudniki, železnicami, žičnicami, postajami, taksiji, cestno infrastrukturo, pa tudi s takimi, ki so v tehničnem smislu bliže računalništvu, npr. *obratovanje in vzdrževanje preklopnih in prenosnih naprav za zagotavljanje točkovnih zvez*. Nanaša se tudi konkretno na informatiko: *obratovanje računalniških sistemov na lokaciji uporabnika ter druge strokovne in tehnične dejavnosti, povezane z računalniki* (SKD 2008, oddelek 62). Izraz *obratovanje* se pojavi v tej klasifikaciji več kot stokrat in tako rekoč v vseh njenih oddelkih. Za ilustracijo in argumentiranje predloga je primerov dovolj. Gre namreč za preprosto logiko: potem ko je tehnična naprava izdelana ali sistem vzpostavljen, mora obratovati, da služi svojemu namenu. Za nemoteno obratovanje je potrebnih brez števila najrazličnejših dejavnosti in posegov. Vse našteto velja tudi za informacijski sistem, ki omogoča zbiranje, obdelavo, shranjevanje, distribucijo ter uporabo podatkov in informacij in katerega element je tudi programje.

Druga ali dodatna možnost je drugačno lastništvo programja. Pri operacijskih sistemih smo se že zdavnaj navadili, da jih sicer plačamo, jih imamo v uporabi, nismo pa njihov lastnik. V tem primeru redno plačevanje za uporabo programja ne pomeni plačila vzdrževanja, temveč neke vrste licenčnino. To razume vsakdo in s tem ni večjih težav, pa tudi računi so čisti: ponudniku ni treba imeti razvijalcev v zamrzovalniku, naročniki pa niso zavedeni, da plačujejo nekaj, česar ne dobijo. S plačilom licenčnine pridobi naročnik pravico do uporabe programja ves čas njegovega delovanja in tudi vse dopolnitve programja, ki jih ponudnik opravi iz kakršnih koli razlogov, vzdrževanje programja pa v tem primeru ni več tema, ki bi vstopala v odnos med naročnikom in ponudnikom.

V prispevku je glede vzdrževanja programja odnosu naročnik – ponudnik namenjena posebna pozornost. Razlog za to je, da imajo lahko nesporazumi, če do njih pride, neugodne operativne in finančne posledice za eno ali drugo stran ali kar za obe. Poseben primer tega odnosa je, če naročnik zaposluje tudi informatike, ki so usposobljeni za izvajanje vsaj nekaterih intervencij v programju, vendar je takih primerov glede na celotno populacijo poslovnih subjektov v državi razmeroma malo. Večina se jih namreč razvršča med male poslovne subjekte, ti pa seveda nimajo svojih organizacijskih delov za informatiko in posledično zaposlenih informatikov. Če imajo zaposlene informatike veliki in srednje veliki poslovni subjekti, imajo vsaj del takih posegov pokritih z lastnimi zaposlenimi. Upoštevaje vse programje, ki ga imajo v uporabi, pa so verjetno glede nekaterih storitev odvisni od zunanjih ponudnikov. Če pride do nesporazumov v primeru, da storitve izvajajo njihovi zaposleni, to za poslovni subjekt navadno nima neposrednih ali vidnih finančnih posledic, k boljšemu delovanju ali višji produktivnosti pa gotovo ne pripomore. Odnos naročnik – ponudnik se tako prevede na odnos interni uporabnik – interni izvajalec, vsi drugi razmisleki pa ostanejo veljavni. Tudi v tem primeru je uporabnik bistveno odvisen od izvajalca.

Poseben primer predvidljivih, vendar nezaželenih posegov v informacijski sistem je vdor zlonamernega programja, kot so virusi, črvi, trojanski konji, vohunsko programje, stranska vrata. Čeprav je v večini primerov cilj vdora nepooblaščeno pridobivanje podatkov, s katerimi je mogoče zaslužiti na različne načine, so včasih cilj napada tudi naprave ali programje. V tem primeru je ta element informacijskega sistema poškodovan do take mere, da ne more več služiti izvirnemu namenu, ali celo uničen. Če pride do tega, rešitev ni katera od intervencij, ki so navedene v razdelku 3, temveč popolna rekonstrukcija poškodovanega ali uničenega programja. Tudi tu ne moremo govoriti o vzdrževanju programja, temveč prej o popolni zamenjavi prizadetega dela informacijskega sistema z njegovo brezhibno izvirno verzijo. Namesto izraza *vzdrževanje* bi bil v tem primeru bolj ustrezen izraz *zamenjava staro za novo*, vendar zgolj zaradi nedvoumnosti, pomensko pa je ta izraz neustrezen približno enako kot prej omenjeni.

Naj navržemo še nekaj misli za konec. Avtor ni naiven in ne misli, da bi se lahko s člankom na konferenci [20] ali po objavi v strokovni reviji razumevanje pri naročnikih in ponudnikih povečalo do take mere, da nesporazumov ne bi bilo več. Položaj ga spominja na podobno dilemo, ki se je sčasoma uredila in danes ni več spora okoli nje. Šlo je za izraz *informacijsko inženirstvo*. Ob neki priliki je avtor dokazoval nekemu profesorju neprimernost te kombinacije izrazov, ki je grob dobesedni prevod angleškega izraza *information engineering*, argumentiral z izrazi drugih tehničnih področij (gradbeništvo, elektrotehnika, strojništvo). Vsi ti imajo v angleščini tudi besedo *engineering*. Trud je bil zaman. Kapitalni argument sogovornika je bil – mi smo se pač odločili za ta izraz. *Ta izraz* je zdaj zgodovina, informatika je zmagala. Ali ne bi mogli narediti nekaj podobnega še pri vzdrževanju programja?

5 ZAHVALE

Članek je izvirno nastal kot predstavitev za posvet Slovenskega društva INFORMATIKA o vzdrževanju programja. Razprava po predstavitvi me je utrdila v prepričanju, da je predmet potreben širše javne obravnave, verjetno pa tudi obravnave v strokovnih in akademskih krogih. Posledica je bil članek za konferenco *Dneve slovenske informatike 2013*, na kateri je bil tudi javno predstavljen. Prva moja zahvala gre vsem, ki so se po predstavitvi udeležili diskusije in posredovali komentarje, mnenja in poglede, čeprav so večinoma zagovarjali stališča ponudnikov programja. Naslednja moja zahvala gre mag. Ranku Smokvini iz *Hrvatskega informatičkega zbora*, ki si je vzel čas, da je prispevek pazljivo prebral in mi posredoval več izredno koristnih komentarjev. Posebej se pa moram zahvaliti še zaslužnemu profesorju dr. Vladislavu Rajkoviču, ki je predlagal, naj prispevek z *DSI 2013* dopolnim za objavo v *Uporabni informatiki*. S svojo prijazno spodbudo in komentarjem o mojem pisanju je dosegel, da mi je bilo delo na tem prispevku v veselje.

6 VIRI IN REFERENCE

[1] Harej, Janko; Čuk, Roman: E-učbenik Vzdrževanje sistemske programske opreme http://www.tsc.si/vss/eucbenik/uvod_v_spo/uvod_v_spo.html (pridobljeno 10. 3. 2013).

[2] Vzdrževanje sistemske programske opreme, Andrej Štrancar, ftp://ftp.scv.si/vss/andrej_strancar/Predavanja/vso_predavanja.pdf (pridobljeno 10. 3. 2013).

[3] Domača stran Kompa, d. o. o.: http://www.kompa.si/podjetja/programska-oprema/ (pridobljeno 10. 3. 2013).

[4] Štrancar, Andrej: VSO Vzdrževanje sistemske programske opreme (ppt) andrej.strancar@fri.uni-lj.si (pridobljeno 10. 3. 2013).

[5] http://en.wikipedia.org/wiki/Software_maintenance (pridobljeno 12. 7. 2013).

[6] http://www.islovar.org (pridobljeno 10. 3. 2013).

[7] http://bos.zrc-sazu.si/cgi/a03.exe?name=sskj_testa&expression=vzdr%C5%BEevati&hs=1(pridobljeno 10. 3. 2013).

[8] Strategija upravljanja z rjavim medvedom (Ursus arctos) v Sloveniji (http://zakonodaja.gov.si/rpsi/r03/predpis_DRUG1103.html) (pridobljeno 10. 3. 2013).

[9] Schlamberger, Niko: Informacijski sistemi in kvaliteta, Informacijske storitve za lokalno samoupravo, zbornik, Ljubljana 1993.

[10] Schlamberger, Niko: Reinženiring – kaj pa negospodarstvo? Konferenca o prenovitvi informacijskih sistemov (3; 1994) Re '94: Zbornik referatov Ljubljana: SRC 1994 (str. 1–10), ISBN 961-6113-00-3.

[11] Ustna komunikacija, več naročnikov.

[12] Pogodbe o vzdrževanju programja, več primerkov.

[13] Schlamberger, Niko: Vzdrževanje programskih rešitev, Ljubljana 2009, (interni zapis, Ministrstvo za zdravje).

[14] Pojasnila k standardni klasifikaciji dejavnosti – SKD 2008: http://www.uradni-list.si/files/RS_-2008-017-00559-OB~P002-0000.PDF (pridobljeno 10. 3. 2013).

[15] wikipedia.org/wiki/Software_maintenance (pridobljeno 10. 3. 2013).

[16] Rajlich, V. T., Bennett, K. H., A Staged Model for the Software Life Cycle, Computer, July 2000, str. 66–71.

[17] Fred Brooks, *The Mythical Man-Month*. Addison-Wesley, 1975 & 1995. ISBN 0-201-00650-2 & ISBN 0-201-83595-9.

[18] K. H. Bennett, V. T. Rajlich, R. Mohamad Mazrul, *Legacy System: Coping with success. IEEE Software,* 1995, str. 19–23.

[19] Lientz, B. P. in Swanson, E. B., *Software Maintenance Management, A Study Of The Maintenance Of Computer Application Software In 487 Data Processing Organizations*. Addison-Wesley, Reading MA, 1980. ISBN 0-201-04205-3.

[20] Schlamberger, Niko: Vzdrževanje programja – mit ali resničnost? Zbornik prispevkov, Dnevi slovenske informatike 2013, Ljubljana 2013, ISBN 978-961-6165-38-9.

▪

Niko Schlamberger se je po diplomi na Fakulteti za strojništvo Univerze v Ljubljani zaposlil v razvojnem oddelku tovarne optičnih in finomehaničnih izdelkov, že med študijem pa ga je zanimalo računalništvo. Na področju informatike in računalništva deluje večino svoje poklicne poti. Njegove delovne izkušnje obsegajo programiranje, sistemsko analizo in razvijanje računalniških rešitev, predavanja, izvajanje usposabljanja na področju informatike, svetovanje, vodenje projektov, vodenje in upravljanje ter mednarodno sodelovanje.

Leta 1997 je bil izvoljen za predsednika Slovenskega društva INFORMATIKA, v katerem je bil izvoljen v drugi mandat decembra 2002 in v tretji mandat marca 2007. Leta 2001 je bil izvoljen za člana izvršnega odbora mednarodnega združenja za obdelavo podatkov (International Federation for Information Processing, IFIP), v katerem je dva triletna mandata (2003 in 2006) opravljal funkcijo podpredsednika. Leta 2006 je bil izvoljen (dveletni mandat je nastopil jeseni 2007) za predsednika evropskega združenja informatikov Council of European Professional Informatics Societies (CEPIS), katerega član je SDI od leta 1998. Sodeloval je pri ustanovitvi Evropskega foruma za poslovne registre (EBRF), ki deluje od leta 1999, in leta 2001 pri ustanovitvi regionalnega telesa za informatiko IT STAR (Information Technology Sanding Regional Committee), ki združuje štirinajst društev informatikov srednje- in vzhodnoevropske regije.