

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

DIPLOMSKO DELO
POSLOVNI MODELI ODPRTEGA IN PROSTEGA PROGRAMJA

Ljubljana, september 2001

ALJAŽ MAVRIČ

IZJAVA

Študent **Aljaž Mavrič** izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom **prof. dr. Borke Jerman Blažič** in dovolim objavo diplomskega dela na fakultetni domači strani.

V Ljubljani, dne 11.09.2001

Podpis:

Kazalo

1. Uvod.....	6
2. Odprto in prosto programje.....	6
2.1 prvo obdobje: začetek 1960-tih do začetka 1980-tih let.....	7
2.2 drugo obdobje: začetek 1980-tih do začetka 1990-tih let.....	9
2.3 tretje obdobje: začetek 1990-tih do danes.....	9
3. Razvojni modeli.....	10
3.1 Katedrala in tržnica (The Cathedral and the Bazaar).....	10
4. Novi poslovni modeli ekonomije odprte in proste programske kode.....	12
4.1 Model "Zunanje financirani projekti".....	12
4.1.1 Javno financiranje.....	12
4.1.2 Financiranje zaradi popravkov.....	13
4.1.3 Posredno financiranje.....	13
4.2 Model "Interno financiranje".....	14
4.3 Model "Nefinanciran razvoj".....	17
4.4 Model "Interna uporaba".....	17
5. Licence odprtega in prostega programja.....	18
5.1 BSD (Berkeley Software Distribution).....	18
5.2 GPL (GNU General Public License).....	18
5.3 MPL (Mozilla Public License).....	19
6. Ekonomska teorija odprtega in prostega programja.....	19
6.1 Motivacija razvijalcev.....	19
6.2 Primerjava nagibov programerjev.....	20
6.3 Vpliv odprtega in prostega programja na TCO.....	22
7. Vplivi odprte in proste programske opreme.....	28
7.1 Odprta programska oprema in nova konkurenca na trgih programske opreme.....	29
7.2 Intelektualna lastnina.....	30
7.2.1 Odprto, prosto programje in avtorsko pravo.....	30
7.2.2 Odprta, prosta programska oprema in programski patenti.....	31
7.3 Vplivi in možnosti uporabe v izobraževalnem sistemu.....	32
8. Nekaj specifičnih vplivov odprte in proste tehnologije.....	34
8.1 Promocija de-facto in de-jure standardov.....	34
8.1 Razpršitev in razširjenost informacijske tehnologije.....	35
8.2.1 Dostopnost varnejših sistemov.....	35

8.3 Vpliv uporabe odprte kode v manj razvitih državah.....	36
8.4 Nekaj možnih scenarijev za prihodnost odprte in proste programske opreme.....	37
8.4.1 Scenarij brez pomoči države ali drugih institucij.....	37
8.4.2 Scenarij delne podpore.....	38
8.4.3 Agresivni scenarij.....	39
8.5 Prihodnost odprtih in prostih sistemov po predvidevanjih Gartner Group.....	39
9. Aktualna ekonomska vprašanja.....	41
10. Sklep.....	47
11. Literatura	48

1. Uvod

Prav tako kot osebni računalnik in Internet je zadnje čase pozornost sektorja informacijske tehnologije¹, gospodarstva in nenazadnje tudi medijev pritegnilo odprto in prosto programje². Vpliv odprte in proste programske kode, njegovi modeli uporabe in razvoja, prinašajo veliko sprememb, ne le v programsko industrijo in z njo posredno v celotni informacijski sektor, ampak tudi na družbo kot celoto. Razloge za tovrstno zanimanje lahko iščemo v treh glavnih dejavnikih: široki uporabi in hitrem širjenju odprte in proste programske opreme, velikih kapitalskih vložkih v projekte odprte in proste programske opreme in novi organizacijski strukturi sodelovanja pri razvoju odprtih in prostih projektov, ki jo poslovni svet in tehnična literatura ocenjujeta kot pomembno organizacijsko novost. V diplomski nalogi bom zato predstavil poslovne modele odprtega in prostega programja in nato ocenil nekatere vplive, ki jih tovrstna oprema ima in jih bo imela na razvoj elektronskega poslovanja v najširšem pomenu besede. Nekateri vplivi so le projekcija trenutnih trendov, drugi postajajo realnost, ali pa bodo še nastali. Poudarek bo predvsem na novih ekonomskih in razvojnih modelih odprte in proste programske tehnologije, razlogih za uspeh, neuspeh, načinih in smereh zagotavljanja stalnega razvoja in inovacij, predvsem pa dolgoročnega obstoja in povečevanja lastniške vrednosti kapitala podjetij, ki tovrstne poslovne modele uporabljajo.

2. Odprto in prosto programje

Odprto in prosto programje je več kot le tehnologija, je ideologija z bogato zgodovino, ki vključuje celotno zgodovino operacijskega sistema Unix³, Interneta in hackerske⁴ kulture. Za razumevanje prihodnosti je potrebno poznati tudi njegovo kulturo in zgodovino.

1 IT - informacijska tehnologija (ang.: information technology)

2 odprto in prosto programje (ang.: open source software, free software, libre software)

Glede na slovensko strokovno literaturo se izraz 'open source / free software' prevaja kot odprta in prosta programska oprema, pri čemer je potrebno opozoriti na razlike med odprto programsko opremo in prosto programsko opremo po definiciji Free Software Foundation [URL: <http://www.fsf.org>], 11.09.2001.

Beseda 'free' v 'free software' se uporablja v pomenu 'free speech', 'freedom' in ne v pomenu 'free beer'. Najboljši približek bi v slovenskem prevodu pomenil 'brezplačen', pri čemer je potrebno opozoriti, da besede 'brezplačen' ne gre zamenjevati za besedo 'zastonj'.

3 Unix - operacijski sistem, ustvarjen v AT&T Bell Laboratories, katerega začetki segajo v leto 1969.

4 hacker, cracker

Pomembno je razumevanje razlike med hackerji in crackerji. Hackerji so tisti, ki dobro poznajo tehnologijo in tudi sami razvijajo in prispevajo inovacije. Crackerji so tisti, ki svoje znanje ali znanje drugih oseb izrabljajo za vdiranje v tuje računalnike in sisteme.

Čeprav so vse zgodbe povezane s programsko opremo na splošno zelo kratke, je zgodovina odprte programske kode ena izmed najdaljših. Lahko rečemo, da je bila na začetku samo prosta programska oprema brez lastnika. V toku časa se je rodila lastniška (v smislu last podjetja, ki jo je razvilo in jo trži) programska oprema, ki je hitro prevladala in postala za večino edini možni model razvoja in prodaje. Šele pred kratkim, je začela programska industrija razmišljati drugače in je med opcije vključila tudi odprto in prosto programje. V svetu, kjer monopol podjetja Microsoft⁵ grozi popolnemu prevzemu in kontroli računalništva, Interneta in standardov, njegovi najmočnejši konkurenti niso več tradicionalni tekmeči, ampak odprto in prosto programje. Medtem ko ta fenomen preseneča nekatere ekonomske analitike, ne bi smel presenečati tistih, ki poznajo nekaj osnovne zgodovine gibanja odprtega in prostega programja.

Odprto programje, ki ga je financirala ameriška zvezna vlada je pomenilo izvir in začetek računalniške industrije in je še danes srce obstoja in razvoja Interneta. Desetletja je ameriška zvezna vlada podpirala razvoj odprtega programja, tako da je podpirala razvoj in sprejem odprtih standardov⁶. Z opustitvijo vladnega financiranja in podpore odprtih standardov zaradi kombinacije ideoloških nasprotovanj, zahtev industrije in pomanjkanja stabilne vladne birokracije v začetku devetdesetih let ter komercializacijo programske opreme in Interneta se je odprla podjetjem kot je Microsoft možnost dominacije računalniškega sveta z lastnimi zaprtimi standardi in nekompatibilno programsko opremo.

Razvoj odprtega in prostega programja lahko razdelimo v tri obdobja:

2.1 prvo obobje: začetek 1960-tih do začetka 1980-tih let

Ko so v 1960-tih IBM in ostali prodali svoj prvi veliki komercialni računalnik, je bila zraven priložena brezplačna programska oprema, da so jo lahko uporabniki prosto uporabljali in delili, popravljali in razširjali. V tem obdobju so to opremo lahko uporabljali na specifični strojni opremi IBM, ker oprema ni bila prenosljiva. V poznih 1960-tih se je stanje spremenilo. V sredini 1970-tih je bilo popolnoma normalno, da je programska oprema zaprta in komercialna.

5 Microsoft [URL: <http://www.microsoft.com>], 11.09.2001.

6 Protokoli Interneta postanejo standard Interneta, če jih sprejme in potrdi združenje IETF (Internet Engineering Task Force) [URL: <http://www.ietf.org>], 11.09.2001 in če je neodvisna koda izdelana v dveh neodvisnih okoljih sposobna nemoteno komunicirati.

Seveda se je potrebno zavedati dejstva, da se je večino ključnih elementov in gradnikov računalniških operacijskih sistemov in Interneta razvilo v akademskem okolju kot sta Berkley⁷ in MIT⁸ med leti 1960 in 1970 pa tudi v avtonomnih razvojnih centrih večjih korporacij kot sta Bell Labs⁹ in Xerox's Palo Alto Research Center¹⁰ ob podpori ministrstva za obrambo. V teh letih je bila skupna uporaba izvirne kode¹¹ tudi med različnimi organizacijami popolnoma naraven pojav. Veliko skupinskega napora in sodelovanja je bilo v 1970-tih vloženo v razvoj operacijskega sistema, ki bi ga lahko uporabljali na različnih računalniških platformah (rezultata tega sta, na primer programski jezik C in operacijski sistem Unix). Večina organizacij, ki je na ta način razvito programsko opremo uporabljala, je prispevala svoje inovacije in popravke in jih tudi delila z ostalimi. Proces skupne uporabe in deljenja programske kode je izredno pospešil razvoj Usenet-a, računalniške mreže, ki se je začela razvijati leta 1979, da bi med seboj povezala uporabnike in razvijalce opreme za operacijski sistem Unix. S strmo rastjo števila udeležencev v omrežju (trije leta 1979, štiristo leta 1982) se je tudi možnost skupne uporabe močno povečala. Ta skupen programski razvoj se je vršil na zelo neformalni bazi, ki ni vlagala nobenega napora v kakršnokoli zaščito avtorskih pravic ali omejitve uporabe. Ta neformalnost se je izkazala za velik problem v začetku 1980-tih, ko je AT&T začel uveljavljati intelektualne pravice nad operacijskim sistemom Unix.

7 University of Berkley [URL: <http://www.uofb.com>], 11.09.2001.

8 MIT - Massachusetts Institute of Technology [URL: web.mit.edu], 11.09.2001.

9 Bell Labs [URL:<http://www.research.att.com>], 11.09.2001.

10 Xerox Palo Alto Research Center [URL: <http://www.parc.xerox.com/>], 11.09.2001.

11 izvorna koda (ang.: source code)

Programerji pišejo izvorno kodo v računalniških jezikih kot so C, C++, Java, ... Pred uporabo je potrebno ta človeku razumljiv zapis spremeniti v ničle in enice, ki jih bo lahko razumel računalnik.

2.2 drugo obdobje: začetek 1980-tih do začetka 1990-tih let

Kot odgovor tem grožnjam in pravnim problemom so se pojavili prvi napori formalizacije pravil skupinskega razvoja programskih projektov. Najpomembnejša ustanova tega obdobja je bila Free Software Foundation¹², ki jo je ustanovil Richard Stallman, raziskovalec pri MIT Artificial Intelligence Laboratory leta 1983. Ena izmed najpomembnejših inovacij predstavljena s strani Free Software Foundation, je bila licenca, ki je preprečevala komercializacijo in zapiranje skupinsko razvitih programskih tehnologij. GNU¹³ programska oprema tako uporabniku daje pravico do proste uporabe, razvoja in distribucije programske opreme.

2.3 tretje obdobje: začetek 1990-tih do danes

Hiter razvoj Interneta in števila njegovih uporabnikov je povzročil izredno hiter razvoj odprte in proste programske opreme. Število razvijalcev in raznolikost je ustvarila množico projektov, med njimi, lahko rečemo, tudi najpomembnejšega Linux (operacijski sistem, katerega začetki segajo v leto 1991, ko ga je iz nezadovoljstva nad takratnim stanjem na trgu operacijskih sistemov ob pomoči takrat že razvitih orodij GNU projekta, začel finski študent Linus Torvalds. Linux se je s pomočjo tisočih programerjev s celega sveta razvil v moderen, dozorel in stabilen operacijski sistem, ki v določenih pogledih presega svoje komercialne tekmece in je vsakomur na voljo pod pravili FSF General Public Licence).

Značilen za to obdobje je tudi vstop največjih podjetij IT sveta (Oracle¹⁴, Sun¹⁵, IBM¹⁶, SAP¹⁷, HP¹⁸, Sgi¹⁹, ...) v svet Linuxa in odprte programske opreme. Razloge lahko iščemo v marketinškem in tehnološkem priznanju kvalitet odprte in proste programske opreme, pa tudi kot reakcijo na Microsoftovo strategijo prevzemov.²⁰ Tekmeci, ki jim je njihove komercialne alternative potopil ali prevzel Microsoft, so v odprti programski opremi začeli opažati možnost

12 Free Software Foundation [URL: <http://www.fsf.org>], 11.09.2001

13 GNU - akronim za GNU is not Unix

GNU programska oprema pomeni programsko opremo razvito pod licenco GPL (General Public Licence), poznana tudi kot 'copyleft'.

Potrebno je opozoriti tudi na projekte kot so Berkley Software distribution (BSD), kjer so uporabili alternativni pristop. BSD licenca je veliko manj restriktivna, kot GPL in omogoča modifikacijo in redistribucijo programa brez izvirne kode.

14 Oracle [URL: <http://www.oracle.com>], 11.09.2001.

15 Sun [URL: <http://www.sun.com>], 11.09.2001.

16 IBM [URL: <http://www.ibm.com>], 11.09.2001.

17 SAP [URL: <http://www.sap.com>], 11.09.2001.

18 HP [URL: <http://www.hp.com>], 11.09.2001.

19 Sgi [URL: <http://www.sgi.com>], 11.09.2001.

20 strategijo prevzemov in načina preprečevanja konkurenčnih projektov v strokovni angleški literaturi imenujemo 'embrace and extend'.

za ustavitev Microsoftove prevlade. Zaupanje v velikane informacijskega sveta bo gotovo pripomoglo pri uveljavljanju Linuxa in odprtih tehnologij, ustvarilo pa bo tudi oportunistične napetosti, ki jih bo potrebno uspešno razrešiti, če želimo, da odprta in prosta programska koda uspe. Druga značilnost tega obdobja je razvoj alternativnih pristopov k licenciranju kooperativno razvitih programskih paketov. V 1980-tih je bila GPL dominantna licenca. To se je močno spremenilo v 1990-tih, predvsem zaradi potrebe po večji fleksibilnosti pri povezavah z zaprtimi programskimi paketi in določenih preveč restriktivnih omejitvah GPL licence.

3. Razvojni modeli

Vpliv odprte in proste programske kode se ne kaže samo na trgu informacijske tehnologije, ampak tudi na področju modelov razvoja programske opreme. Glede na klasične koncepte programskega inženiringa lahko samo centraliziran management in močna kontrola pri dostopu do izvorne kode določata dober, visoko kakovosten programski izdelek. Ta koncept se je vsaj delno izkazal za nepravilnega zaradi uspeha večjega števila odprtih in prostih projektov, kjer je večjemu številu razpršenih razvijalcev uspelo zgraditi visoko kvalitetne, zanesljive in uspešne izdelke.

3.1 Katedrala in tržnica (The Cathedral and the Bazaar)²¹

Obstaja množica poizkusov razlage stanja, ki kaže na uspeh načinov razvoja odprtih programskih projektov. Kar nekaj je dejavnikov, ki se pojavljajo samo pri odprtih in prostih projektih in se ne skladajo s standardnimi zaključki. Enega izmed njih lahko povzamemo z besedami Linusa Torvaldsa: "z dovolj velikim številom oči tudi hrošči v programski opremi izginejo", ki kaže na fazo razhroščevanja²² in omogoča paralelno izvajanje le te s pomočjo koordinacije med razvijalci. Seveda je ta proces možen samo v primeru proste dosegljivosti izvorne kode. Ena izmed prvih študij metod in novih modelov, ter uspešnih načinov vodenja razvoja programske opreme je delo Erica Raymonda "The Cathedral and the Bazaar"²³, v katerem analizira dve različni kategoriji razvoja. Ena izmed njiju, ki ji pravi "cathedral-like development" (analogija z načinom izgradnje katedral v srednjem veku) in karakterizira relativno močno kontrolo pri načinu izvedbe in obliki. Drugi (imenuje ga "bazaar-like development"), je zasnovan na neformalni komunikaciji med razvijalci in večjim številom projektov, med katerimi se lahko razvijalci prosto odločajo.

21 Eric S. Raymond. The cathedral and the bazaar, 1998. [URL: <http://tuxedo.org/esr/writings/cathedral-bazaar/>], 26.09.2001.

22 razhroščevanje (ang.: debug) - pomeni iskanje in odpravljanje napak v programu s testiranjem programa ali pregledom izvorne kode.

23 Eric S. Raymond. The cathedral and the bazaar, 1998. [URL: <http://tuxedo.org/esr/writings/cathedral-bazaar/>], 26.09.2001.

Prvi je tradicionalni način programskega vodenja in je v uporabi tudi pri nekaterih odprtih in prostih projektih. V tem modelu je močna kontrola, kdo lahko prispeva popravke in kontrola nad načinom njihove integracije, jasen in natančno definiran plan novih različic in izdaj. V večini primerov je plan novih različic in izdaj veliko bolj določen zaradi trga, kot zaradi kvalitete, kar vodi do programske opreme slabše kvalitete. Drugi model, ki je zasnovan na neformalni komunikaciji, nima natančno določenih rokov in relativno neformalne kontrole nad prispevki in popravki ter razvojem nove funkcionalnosti. Ta model lahko karakteriziramo z izrazom "programiraj veliko, objavljaj nove verzije"²⁴, ki kaže na trenutno stanje v svetu odprte in proste programske kode, kjer je stabilna različica programske opreme na voljo za splošno uporabo, nova funkcionalnost in popravki, pa se delajo na razvojni različici. Medtem ko nove verzije stabilne različice izhajajo bolj poredko zaradi bolj natančnega testiranja in procedur za zagotavljanje kvalitete se razvojne verzije izdajajo veliko bolj pogosto in so namenjene testiranju, tako zaradi programerjev, kot tudi uporabnikov. Zaradi stopničastega razvoja obstaja določeno obdobje, ko v začetni fazi razvoja program še ni dovolj funkcionalen, da bi privabil pozornost ostalih programerjev in virov financiranja (v obliki denarja, časa in drugih virov) za dosego praga uporabnosti. Ko je ta prag dosežen, se lahko začne kazati moč razvojnega modela odprte in proste programske opreme.

Oba primera sta ekstrema med razvojnimi modeli programske opreme. Medtem ko lahko odprti programski projekti uporabljajo poljuben model, komercialna, lastniška, zaprta programska oprema le težko uporabi kaj drugega od svojih tradicionalnih modelov in zato izgublja vse prednosti in lastnosti razvojnih modelov bazarja.

Seveda pa je potrebno razumeti, da odprto in prosto programje lahko ustvarjamo z obema modeloma. Primeri kot so gcc prevajalnik²⁵ in XFree86²⁶ grafični sistem jasno kažejo, da je tudi "katedralni" model lahko uspešen. Bistvo je, da odprta in prosta programska oprema dopušča tudi uspešne alternativne razvojne modele in to v primerih, kjer "katedralni" način niti ni možen.

24 ang.: code often, release often

25 gcc [URL: <http://gcc.gnu.org>], 17.09.2001.

26 XFree86 [URL: <http://www.xfree86.org>], 17.09.2001.

4. Novi poslovni modeli ekonomije odprte in proste programske kode

Poglavlje novih ekonomskih modelov je pomembno, ker tradicionalni modeli programske industrije močno slonijo na zaprti, komercialni programski opremi, kjer je glavni prihodek direktno povezan s ceno licenčne kopije izdelka. Z nekaj izjemami ti modeli niso možni pri odprti in prosti programski opremi, predvsem zato, ker prihodki ne nastajajo s prodajo kopij programa (svoboda redistribucije in prostega kopiranja postavlja ceno, kjer so mejni stroški reprodukcije skoraj 0). Zato mora poslovni model odprte in proste programske opreme iskati vire svojih prihodkov drugje.

Klasifikacija predstavljenih novih ekonomskih modelov v tem poglavju ni le analiza trenutno obstoječih modelov. Prav nasprotno, predstavljena je kar najbolj popolna kategorizacija samoobstoječih ali vsaj dosegljivih modelov. Resnični primeri so dodani vsakemu modelu, seveda, če so poznani.

4.1 Model "Zunanje financirani projekti"

V ta model spadajo skupine ali podjetja, ki razvijajo odprto in prosto kodo zaradi pobude (v finančnem smislu) neke zunanje organizacije. Ponavadi te zunanje organizacije določajo način porabe denarja in smeri razvoja projekta. Razvijalci več ali manj le sledijo tem bolj ali manj natančno postavljenim smernicam. Lahko rečemo, da zunanja organizacija "podpira, sponzorira" razvoj nekega projekta. V tej kategoriji lahko določimo vsaj tri modele glede na to, kdo financira projekt in zakaj. Imenujmo jih javno financiranje, financiranje zaradi popravkov in posredno financiranje.

4.1.1 Javno financiranje

Delovne skupine ali posamezniki dobijo plačilo za razvoj dobrega programskega produkta, dokumentacije, izvedbo procesa testiranja ali karkoli tretjega. Ponavadi je edina omejitev, ki jo postavlja financer, da se morajo sredstva porabiti za uspešno zaključitev projekta. To je značilno za velike računalniške znanstvene projekte, ki jih ponavadi financira univerza ali proračun. Dejstvo je, da se večina velikih projektov, kot so radioastronomija, računalniška kemija in biologija financirajo na ta način. Dodatno imajo nekateri konzorciji za razvoj Internet orodij in tehnologije (ali so imeli), takšno strukturo financiranja. V večini tovrstnih primerov investitor ne pričakuje povrnitve investicije ali direktnih koristi od nje. Pričakujejo se socialne izboljšave in razvoj znanja.

4.1.2 Financiranje zaradi popravkov

Podjetje ali organizacija morda potrebuje novo ali izboljšano verzijo programskega paketa, produkta. Zato financira svetovalce oziroma programerje ali proizvajalca programske opreme. V kasnejši fazi razvoja se rezultat dela preda v roke skupnosti odprte in proste programske opreme zaradi vseh prednosti, ki jih prinaša velika skupina kvalitetnih razvijalcev in uporabnikov, ki izdelek še naprej razvijajo in popravljajo.

Primer tovrstnega financiranja je model dela Macadamian Software²⁷ Corel korporacije pri razvoju Wine Windows²⁸ emulatorja²⁹. To delo je omogočilo podjetju Corel³⁰, tako da je preneslo³¹ celoten spekter svoje programske opreme tudi na druge platforme, enostavno samo s ponovnim prevodom³² programske opreme. To je bilo veliko ceneje kot izvedba celotnega prenosa ali razvoja zaprtega tipa emulatorja.

Bolj marginalen primer tovrstnega razvoja v Sloveniji predstavlja projekt POP TV 24ur radijske novice, ki z uporabo odprte in proste programske opreme in orodij omogoča varno in ažurno redistribucijo večje količine mp3 zvočnih zapisov preko ISDN povezav na večje število lokacij. V tem primeru je slovensko podjetje Agenda Open Systems³³ pri procesu razvoja sistema razvilo kar nekaj uporabnih orodij montaže mp3, wav in ostalih formatov datotek, ki jih sedaj uporablja tudi nekaj tujih razvijalcev.

4.1.3 Posredno financiranje

Podjetje se lahko odloči financirati odprte in proste programske projekte, če ti projekti ustvarjajo znaten delež prihodkov povezanih izdelokov, ki niso direktno povezani z odprto kodo ali programsko opremo. Pri posrednem financiranju podjetje išče komplementarne vplive odprtega produkta.

27 Macadamian Software [URL: <http://www.macadamian.com>], 11.09.2001.

28 Wine [URL: <http://www.winehq.com>], 11.09.2001 - wine je implementacija Windows 3.x in WIN32 API sistema na osnovi X grafičnega sistema in Unixa in omogoča uporabo Windows aplikacij na Unix sistemih.

29 emulator - sistem, ki v določeni obliki simulira značilnosti in lastnosti drugega sistema.

30 Corel [URL: <http://www.corel.com>], 26.09.2001.

31 prenos programske opreme (ang.: porting) pomeni pripravo programskega produkta za uporabo na drugih platformah ali operacijskih sistemih.

32 prevod v tem pomenu pomeni spremembo izvorne kode v obliko razumljivo računalniku (ang.: compilation).

33 Agenda d.o.o. [URL: <http://www.agenda.si>], 17.09.2001.

Primer v tej skupini je lahko O'Reilly & Associates³⁴, založnik specializiran za tehnično literaturo. Na listi svojih zaposlenih imajo nekaj najpomembnejših razvijalcev programskega jezika Perl³⁵ kot način zagotavljanja razvoja programskega jezika in s tem povezanim povpraševanjem po Perl literaturi, katere založnik, vsaj najkvalitetnejši so oni.

Drug primer je podjetje VA Linux³⁶, do nedavnega prodajalec strežniških sistemov s prednameščeno Linux opremo, danes pa usmerjeni predvsem v storitve s področja odprte in proste programske opreme. Zaposlili so nekaj pomembnih razvijalcev jedra in s tem podprli konstanten razvoj Linuxa. Z boljšim Linux jedrom se razširijo področja uporabe Linuxa in s tem posredno tudi prodaja in uporaba Linux strojne, programske opreme. Kljub temu, da imajo tudi ostali ponudniki strojne opreme koristi od te rasti, ima VA Linux velik tržni delež, s katerim lahko opraviči investicije v splošne izboljšave Linux jedra. Npr. VA Linux porabi cca. 1 mio USD za izboljšave jedra, medtem ko je rast trga zaradi tovrstnih izboljšav 200 milijonov USD, od katerih ima VA Linux približno 20% (40 mio), kar pomeni dodaten dohodek cca. 4 mio USD. To očitno opraviči porabo enega milijona.

Na drugi strani imamo podjetje Red Hat³⁷, ki je odličen primer podjetja, ki posredno financira razvoj prostih in odprtih produktov. Ustanovili so Red Hat Advanced Development Labs in zaposlili vrhunske strokovnjake, na strateških področjih odprte in proste programske opreme. S tovrstnim financiranjem so ustvarili nekaj produktov, ki močno izboljšujejo prodajo njihovega glavnega produkta (Red Hat Linux distribucije).

4.2 Model "Interno financiranje"

V to kategorijo vključujemo modele, pri katerih organizacija razvija in vzdržuje odprte in proste projekte zaradi prodaje storitev ali produktov, ki so s projektom povezani ali izboljšujejo njegovo produktivnost.

34 O'Reilly [URL: <http://www.oreilly.com>], 17.09.2001.

35 Perl - eden izmed najpomembnejših, vsaj kar se tiče Interneta in Unixa skriptnih programskih jezikov (www.perl.org)

36 VA Linux [URL: <http://www.valinux.com>], 17.09.2001.

37 Red Hat [URL: <http://www.redhat.com>], 11.09.2001.

Najboljše znanje brez omejitev

V tem modelu podjetje deluje kot plačani svetovalec zaradi znanja, ki ga posedujejo zaposleni. Ta model lahko uporabi vsako podjetje, saj ne obstajajo nobene omejitve, ki bi preprečevale sposobnemu razvijalcu pridobitev znanja s področja določenega odprtega ali prostega projekta. Podjetje je v tem modelu izpostavljeno tveganju konkurence, ki ga prehititi v primeru, da investicije v znanje in strokovnost niso dovolj velike, saj imajo konkurenti popolnoma enake možnosti pridobivanja konkurenčnih prednosti. Primer tovrstnega podjetja je Linuxcare³⁸, ki je eno najuspešnejših svetovnih podjetij s področja prodaje storitev odprtih in prostih sistemov. V Sloveniji med tovrstna podjetja štejemo predvsem podjetje Agenda d.o.o. in pa nekaj začetkov manjših računalniških podjetij, ki pa zaenkrat nimajo dovolj strokovnjakov. Seveda je tu potrebno omeniti še večja podjetja, ki imajo strokovnjake s področja Unix sistemov, in ki lahko zelo hitro pridobijo tudi znanje odprtih in prostih sistemov.

Najboljše znanje z omejitvami

Za zaščito pred možnostjo prevzema in 'kraje' strank podjetje lahko določi arbitrarne omejitve na proces deljenja znanja s pomočjo patentov in poslovnih skrivnosti. Seveda je to popolnoma neskladno z načini promocije in razvoja odprte kode. Obstaja možnost objave le manjšega dela programa (seveda v večini primerov vitalnega) pod bolj restriktivno ali celo popolnoma komercialno "black box"³⁹ licenco. Tovrstna omejitve se lahko uveljavlja tudi zaradi dodatnih pravnih pravic ali zaradi izognitve potencialnim pravnim problemom. V preteklosti je skupnost odprtega in prostega programja tovrstne omejitve odpravila s ponovnim pisanjem zaprtih delov in ustvarjanjem odprtih alternativ.

38 LinuxCare [URL: <http://www.linuxcare.com>], 11.09.2001.

39 black box licenca - pomeni, da ima uporabnik dostop le do binarne različice programe, medtem, ko nima informacije o načinu delovanja programa in možnosti varnostnega pregleda programa. V tem primeru je popolnoma odvisen od svojega ponudnika in mu mora tudi 100% zaupati.

Najboljša programska koda brez omejitev

V tem primeru podjetje razvije določen odprt programski produkt in prodaja svetovanje in vzdrževanje. Model je podoben modelu "najboljše znanje" vendar z dodatnimi prednostmi glede časovnega roka, saj tekmelec potrebuje daljše obdobje za produkcijo podobnega produkta ali za razumevanje produkta nekoga drugega. To daje časovno prednost podjetju, ki je produkt ustvarilo. Primeri tovrstnega modela so Cygnus solutions⁴⁰ in njihova podpora pri gcc/egcs prevajalniku in glibc knjižnicah. Lep primer je tudi podjetje GreatBridge⁴¹, kjer so glavni razvijalci ena najuspešnejših odprtih in prostih podatkovnih baz podatkov PostgreSQL⁴² ustanovili podjetje, ki trži podporo in storitve in razvoj dodatnih modulov.

Najboljša programska koda z omejitvami

Prejšnji model lahko dopolnimo z nekaterimi dodatnimi omejitvami predvsem zaradi ustvarjanja večjega časovnega razdobja in prednosti pri prodaji produkta. Največkrat uporabljena omejitev je 'zadržana dobavljivost', ko avtor postavi umetne omejitve, ki zastarajo s časom. Primer je podjetje Alladin⁴³ s svojim GhostScript⁴⁴ orodjem.

Najboljša programska koda z dodatki

Razvoj kvalitetnega produkta lahko omogoči prodajo komercialnih dodatkov, ki prvotni produkt naredijo boljši, uporabniku prijaznejši ali mu dodajo funkcionalnost. Eden boljših tovrstnih primerov je podjetje Zend⁴⁵, ki stoji za razvojem trenutno najpopularnejšega odprtega in prostega spletnega skriptnega jezika PHP⁴⁶, in ki svoje prihodke ustvarja z dodatnimi komercialnimi aplikacijami kot so razvojna okolja, cache strežniki in podobno.

Posebne licence

Možno je prodajati tudi 'izjeme' odprih licenc za specifične komercialne aplikacije. Npr. SleppyCat⁴⁷ programska hiša distribuira svoje podatkovno orodje pod licenco GPL, vendar s posebno licenco omogoča integracijo svojega produkta v zaprte projekte, kar je nemogoče pri upoštevanju GPL licence.

40 Cygnus [URL: <http://www.cygnus.com>], 11.09.2001.

41 GreatBridge [URL: <http://www.greatbridge.com>], 10.08.2001.

42 PostgreSQL [URL: <http://www.postgresql.org>, <http://www.postgresql.com>], 11.09.2001.

43 Alladin GhostScript [URL: <http://www.artifex.com/>], 17.09.2001.

44 GhostScript [URL: <http://www.cs.wisc.edu/~ghost/>], 17.09.2001.

45 Zend [URL: <http://www.zend.com>], 11.09.2001.

46 PHP [URL: <http://www.php.net>], 11.09.2001.

47 SleppyCat [URL: <http://www.sleepycat.com>], 11.09.2001.

Prodajanje blagovne znamke

Prepoznavnost imena, znamke, pripomore k večji prodaji in lahko postane medij prodaje kar samo, prav tako kot klasično prodajanje blagovne znamke na drugih trgih. Dober primer je podjetje Red Hat⁴⁸, ki promovira svojo distribucijo tako uspešno, da veliko ljudi enači Linux s specifično Linux distribucijo. V tem lahko iščemo tudi razloge visokih vrednosti delnic in kapitalizacije pri vstopu podjetja na borzne trge.

4.3 Model "Nefinanciran razvoj"

Če obstaja dovolj "mrežnega učinka", potem skoraj ne obstaja potreba po financiranju, ampak le minimalen napor organiziranja razvijalcev in novih različic, popravkov. Primeri tovrstnega razvoja so: Linux jedro, Linux distribucije, kot so Debian⁴⁹, BSD operacijski sistem kot so FreeBSD⁵⁰, NetBSD⁵¹, OpenBSD⁵² in ostali. V večini primerov so se tovrstni primeri začeli kot projekti enega ali manjše skupine razvijalcev. Ti so z dobro organizacijo in prostovoljnimi delom ustvarili široko mrežo razvijalcev. Tudi brez večjih investicij je veliko teh projektov doseglo in preseglo svoje komercialne tekmece.

4.4 Model "Interna uporaba"

Nekateri programski projekti se začnejo kot cenejša alternativa zaprtim, komercialnim sistemom. V tem primeru razvijalci ne pričakujejo, vsaj ne na začetku, kakršnegakoli zunanjskega financiranja ali pomoči. Podjetje razvija nek produkt, ker ga ima za lastno uporabo. Kasneje odloči, da ga bo objavilo pod eno izmed prostih in odprtih licenc, predvsem zaradi prednosti, ki jih prinaša odprti razvojni model. Skoraj gotovo bodo pritegnili kar nekaj razvijalcev, ki bodo prispevali popravke, odpravke hroščev in poročila o napakah. V kasnejši fazi razvoja lahko tovrsten produkt celo doseže priznanje trga in zaradi tega ima podjetje ekonomske prednosti. Lep primer tovrstnega ravnanja je podjetje Cisco⁵³ s svojim Cisco printing sistemom⁵⁴.

48 Red Hat [URL: <http://www.redhat.com>], 10.08.2001.

49 Debian [URL: <http://www.debian.org>], 11.09.2001.

50 FreeBSD [URL: <http://www.freebsd.org>], 17.09.2001.

51 NetBSD [URL: <http://www.netbsd.org>], 17.09.2001.

52 OpenBSD [URL: <http://www.openbsd.org>], 10.09.2001.

53 Cisco Systems [URL: <http://www.cisco.com>], 10.09.2001.

54 CEPS [URL: <http://ceps.sourceforge.net>], 10.09.2001.

5. Licence odprtega in prostega programja

Licenca programske opreme natančno določa pravice proizvajalcev in uporabnikov. Večina zaprte, komercialne programske opreme odvzema uporabniku pravico razmnoževanja, popravljanja, posojanja, uporabe na večih računalniških sistemih in podobno. V večini primerov lastnik programa še vedno ostaja proizvajalec, ki prodaja le omejeno pravico do uporabe. Tudi v svetu odprte in proste programske opreme je licenca primarnega pomena, saj zagotavlja nekaj osnovnih pravic uporabnikov (uporaba, razmnoževanje, popravki, širjenje funkcionalnosti, ...) in seveda tudi avtorjev (navedba imena avtorja in ohranjanje licence v izvorniku in vseh izdelkih, ki iz njega nastanejo). Večina odprtih in prostih projektov uporablja eno izmed najbolj uporabljenih licenc (GPL, LGPL, Artistic, BSD, MPL, ...). Glavne razlike se kažejo v pomembnosti naslednjih dejavnikov:

- varovanju odprtosti (nekatero licence zahtevajo, da vse redistribucije ohranjajo prvotno licenco in z njo pravice uporabnikov)
- varovanju moralnih pravic (v mnogih državah pravo varuje pravico priznanja avtorju za njegovo delo. Nekatero licence izrecno navajajo tudi to pravico, predvsem zaradi zavarovanja pred razlikami v različnih pravnih sistemih)
- varovanje lastniških pravic
- skladje z zaprtimi, komercialnimi licencami
- skladje z drugimi odprtimi in prostimi licencami

Največkrat uporabljene odprte in proste licence so:

5.1 BSD (Berkeley Software Distribution)

BSD licenca je dober primer popolnoma proste in odprte licence, ki uporabniku ne postavlja skoraj nobenih pogojev. Na kratko to pomeni, da lahko redistributor počne s programsko opremo, kar želi, vključno z njeno uporabo v zaprtih produktih. Tovrstna licenca ne vključuje nobenih omejitev, ki bi zagotavljale, da bo tudi v prihodnje delo ohranilo svojo odprtost.

5.2 GPL (GNU General Public License)

GPL je licenca GNU⁵⁵ projekta. Obstaja večja količina programske opreme, ki ni povezana z GNU projektom, vendar kljub temu uporablja GPL licenco (eden najboljših primerov je Linux jedro). GPL licenca je bila zasnovana s ciljem zagotavljanja produkcije čim večje količine proste programske opreme. Zato jasno prepoveduje uporabo GPL izvorne kode v komercialnih zaprtih programih. Glavne značilnosti GPL licence so: dovoljuje distribucijo binarnih programov le pod pogojem, da je omogočen dostop do izvorne kode, dopušča

⁵⁵ GNU [URL: <http://www.gnu.org>], 17.09.2001, GNU - GNU is not Unix.

popravke, brez omejitev (seveda pod pogojem, da so tudi popravki pod licenco GPL), popolna integracija z drugo programsko opremo je možna samo, če je tudi ta programska oprema pod licenco GPL. To ne velja za licenco LGPL (GNU Lesser General Public License), ki je tudi uporabljena v GNU projektu, kar omogoča integracijo z vsemi licencami, vključno s popolnoma komercialnimi.

5.3 MPL (Mozilla Public License)

Ta licenca je bila narejena v podjetju Netscape⁵⁶ za distribucijo projekta Mozilla⁵⁷ (nova verzija Netscape Communicatorja). V večini pogledov je podobna licenci GPL, vendar bolj podjetniško orientirana.

6. Ekonomska teorija odprtega in prostega programja

6.1 Motivacija razvijalcev

Razvijalec sodeluje pri projektu pa naj bo to komercialen ali odprt projekt samo če iz sodelovanja zanj izhajajo določene neto koristi. Neto koristi so enake takojšnjemu plačilu: (trenutne koristi - trenutni stroški) + odloženo plačilo. Srečuje se z večjim številom takojšnjih stroškov in koristi. Kot prvo dobi določeno denarno plačilo, kot drugo lahko popravlja ali nadgrajuje program zaradi lastnih interesov (ali v primeru odprte in proste programske opreme v dobro drugih), kot tretje se srečuje z oportunitetnimi stroški svojega časa. Medtem ko dela na določenem projektu, ne more sodelovati pri drugih projektih. Dejanski strošek tega časa je odvisen od njegovega veselja pri delu na določenem projektu. Odloženo plačilo je sestavljeno iz dveh različnih, vendar težko ločljivih nagibov. Povečevanje svoje vrednosti zaradi dela na določenem projektu in posledično boljše prihodnje zaposlitve, deleža v kapitalu⁵⁸ podjetij ali dostopa do tveganih kapitalskih trgov. Pomemben nagib je tudi priznanje skupnosti, sodelavcev, kolegov, tekmecev. Verjetno imata oba močan vpliv na razvijalce. Med obema obstaja nekaj razlik. Razvijalec, kateremu največ pomeni priznanje skupnosti ("peer recognition"), se lahko odpove prihodnjim denarnim nagradam in si želi svoje priznanje doseči pri drugačni ciljni populaciji, kot tisti, ki ga motivira predvsem uspešna kariera. Z ekonomske perspektive sta oba nagiba zelo podobna, zato bomo karierni nagib in nagib priznanja združili pod skupno ime: signalni nagib.

56 Netscape [URL: <http://www.netscape.com>], 17.09.2001.

57 Mozilla [URL: <http://www.mozilla.org>], 17.09.2001.

58 Linus Torvalds in ostali so bili nagrajeni z deleži delnic v delniških družbah Linux podjetij. Verjetno so bile te nagrade nepričakovane in niso vplivale na motivacijo programerjev in razvijalcev odprtih sistemov. Če tovrstna praksa postane dejstvo, bodo tovrstne nagrade v prihodnosti pričakovane in bodo zato imele večji vpliv na motivacijo vodij razvoja odprte in proste programske kode.

Predvidimo, da je signalni nagib močnejši:

- a) bolj, ko so dosežki vidni pri ciljni populaciji (sodelavci, trgi dela, podjetniški kapital)
- b) bolj, ko delo razvijalca vpliva na dosežke
- c) bolj, ko dosežki kažejo na velik talent

Prvi pogoj razlaga strateško komplementarnost, kar pomeni, da bodo programerji zaradi občinstva želeli sodelovati pri projektih, ki bodo pritegnili večje število programerjev. To posledično pomeni, da bodo močno motivirani za produkcijo visoko kvalitetnih izdelkov, saj so lahko prepričani v to, da bodo dosegli široko občinstvo.

6.2 Primerjava nagibov programerjev pri odprtih, prostih in zaprtih, komercialnih projektih

Primerjava nagibov v projektih odprtega in prostega programja in komercialnih okoljih zahteva razlago glavnih zancilnosti teh dveh okolij. Najprej bo pozornost usmerjena na kratkoročne nagrade, nato pa na bolj dolgoročne. Komercialni projekti v večini primerov zaradi komercialne narave programske opreme in cene licenc ustvarjajo določen prihodek in s tem ponudijo svojim zaposlenim plačo⁵⁹. Ta zaključek je eden izmed najstarejših ekonomskih razlogov, ki pravi, da pričakovanje dobička, spodbuja investicije in je uporabljen tudi za utemeljevanje upravičenosti patentov kot spodbujevalcev inovacij.

Odpri projekti lahko znižujejo stroške programerja zaradi naslednjih dveh razlogov:

- **izobraževalni vpliv**

Ker je izvorna koda prosto dosegljiva, jo lahko uporabljajo tudi šole in univerze pri svojih učnih projektih. S tem se znižujejo stroški programiranja, saj so programerji z njim že dobro seznanjeni in lahko uporabljajo dele že razvitega znanja.

- **prednosti prilagoditev in popravkov**

Stroški prispevanja k odprtim in prostim projektom so nižji, če tovrstna aktivnost prinaša tudi privatne prednosti (prilagoditve, popravki) za programerja in podjetje, ki ga zaposluje. Seveda je potrebno opozoriti, da je faktor zniževanja stroškov močno povezan in možen samo pri odprtosti programske kode.

⁵⁹ Določenim komercialnim podjetjem kot so Netscape, Sun, O'Reilly, Transmeta, ... podpora projektov odprtega in prostega programja posredno prinaša finančne koristi, s čimer lahko plačujejo in podpirajo razvijalce.

Tudi na strani dolgoročnih nagrad prinašajo odprti in prosti sistemi nekaj prednosti v primerjavi z zaprtimi programskimi paketi. Kot smo že omenili, so signalni nagibi močnejši, bolj ko je uspeh posameznika viden in je ta uspeh skupine možno v nekem projektu pripisati posamezniku. Signalni nagibi so zato v odprtih in prostih modelih lahko močnejši zaradi naslednjih razlogov:

- **boljše merjenje rezultatov**

Zunanji opazovalec lahko le opazuje funkcionalnost in/ali kvaliteto posameznih elementov tipične komercialne programske opreme, saj nima dostopa do znanja, ki program sestavlja. Pri odprtih in prostih projektih lahko zunanji opazovalci spremljajo prispevke vsakega posameznega programerja in zahtevnost njegovega dela, predvsem pa način reševanja izziva in možnosti uporabe kode v drugih projektih.

- **samoiniciativa**

Programerji odprtih in prostih sistemov sami odločajo in prevzemajo odgovornost za uspeh ali neuspeh svojega dela projekta. V hierarhičnem zaprtem podjetju je uspeh programerja odvisen od vplivov njegovega nadrejenega. Ekonomska teorija bi zato predvidela, da se da zmogljivost in uspeh programerja bolj natančno določiti v prvem primeru.

- **boljša dinamika**

Predvidimo lahko, da sta dinamičnost in pretok delovne sile večja pri odprtih in prostih projektih zaradi odstotnosti omejitev, ki bi preprečevale prenos naporov iz enega projekta na drugega.

Ti teroretični argumenti kažejo tudi na to, kdo je tisti, ki bo bolj verjetno prispeval svoje znanje in čas, in kateri projekti so tisti, ki bodo najbolj uspeli v svetu odprtega in prostega programja. To so napredni uporabniki, ki takoj dobijo določene ugodnosti, ko popravijo ali prilagodijo določen odprti ali prosti projekt⁶⁰. Drugo kategorijo razvijalcev sestavljajo posamezniki z močnimi signalnimi nagibi, ki lahko odprte in proste projekte uporabijo, kot svoj vstop na različne trge⁶¹.

60 Standardni argument pri prednostih odprte in proste programske opreme in njenega razvoja je masivno paralelno razhroščevanje. Komercialno podjetje lahko svoje uporabnike le prosi, naj pokažejo na določene probleme in težave. Beta uporabniki ne popravljajo, ampak samo poročajo o napakah, medtem ko pri odprtih projektih z določeno mero nadzora lahko omogočimo paralelno popraviljenje in odpravljanje napak.

61 Pogosto omenjeni argument skupnosti odprte in proste programske kode je, da posamezniki sodelujejo pri odprtih in prostih projektih, ker je programiranje zabavno in ker želijo biti del skupine. Medtem, ko ta argument sicer delno drži, je zanimivo, zakaj potem programerji v komercialnih ekipah ne dosegajo enakih intelektualnih izzivov in iste pripadnosti skupini. To kaže na možnost, da programerji preko odprtih in prostih projektov premagujejo določene omejitve trga, ki oneomogočajo učinkovito posredovanje signalov.

Individualni nagibi

Koristi uporabnikov so ključ in glavno vodilo večjega števila odprtih in prostih programskih projektov. Začetke gibanja prostega programja lahko iščemo v Stallmanovi nezmožnosti popraviti program za tiskanje, ker podjetje Xerox⁶² ni želelo objaviti izvirne kode. V vseh največjih, lahko pa rečemo tudi najuspešnejših odprti in prosti projektih (Apache⁶³, Sendmail⁶⁴, Linux, ...), so bili ustanovitelji motivirani z izzivi informacijske tehnologije. V vseh primerih je prva različica omogočala uporabo in testiranje, predvsem pa je predstavljala potencial, ki je bil kljub svoji nedokončanosti rešitev problema. Priznanje avtorju za opravljeno delo je tudi eno izmed osnovnih gibal odprte in proste programske opreme. Ugled, pridobljen zaradi uspešnih prispevkov avtorjev k odprtim in prostim projektom, ima močan vpliv na razvijalce. Npr. Raymond [1999] omenja tri primarne ugodnosti, ki so rezultat uspešnih prispevkov k odprti in prosti programski opremi: dober ugled v skupnosti, pozornosti in sodelovanje zaradi drugih in boljši položaj v trenutni tržni ekonomiji. Projekt spletnega, aplikacijskega strežnika Apache dobro potrjuje tovrstna opazovanja. S priznanjem vseh sodelujočih pri projektu, celo tistih, ki so problem le določili, ne da bi predlagali rešitev in navedbo glavnih razvijalcev, ki imajo kontrolo nad smerjo in načinom evolucije projekta, priznava posamezniku zasluge za njegove prispevke. Večina uspešnih programerjev Apache projekta je tudi finančno nagrajena zaradi sodelovanja pri projektu.

6.3 Vpliv odprtega in prostega programja na celotne stroške lastništva TCO (Total cost of ownership)

Analiza poslovnih modelov nikakor ni dovolj, da bi opravičili zanimanje za odprto in prosto programsko opremo podjetij, ki niso povezana z razvojem in storitvami tega področja ali vzdrževanjem programske opreme. Prisotnost novih možnosti ne more biti zadosten razlog za razširitev tega modela na bolj tradicionalno usmerjena razvojna okolja in še posebej ne na podjetja, katerih glavna usmeritev ni povezana z informacijsko tehnologijo (čeprav lahko morda sloni na njej).

TCO je široko uporabljen in tudi izrabljen parameter določanja cene programske in strojne opreme v življenjskem obdobju. Poizkuša zajeti vse realne stroške razvoja, implementacije in vzdrževanje informacijskega sistema v vseh stopnjah njegovega življenjskega ciklusa. Ne poskuša pa meriti prednosti, ki jih ponuja določena informacijska tehnologija, za katero se ponavadi uporablja ROI (return on investment).

62 Xerox [URL: <http://www.xerox.com>], 10.09.2001.

63 Apache [URL: <http://www.apache.org>], 10.09.2001.

64 Sendmail [URL: <http://www.sendmail.org> <http://www.sendmail.com>], 10.09.2001.

V tem delu bom predstavil primerjavo med TCO zaprtih sistemov in sistemov, zgrajenih na odprti in prosti programski opremljeni. TCO je skupen strošek pridobitve in implementacije informacijskega sistema v njegovi življenjski dobi. To življenjsko dobo lahko razdelimo na več delov: pridobivanje informacij, implementacija, izobraževanje, uporaba in vzdrževanje, opustitev in prehod na nove tehnologije. Poglejmo si vsako posebej malo natančneje:

- **zbiranje informacij**

Vključuje zbiranje podatkov o možnih dosegljivih rešitvah. Trditev, da je zbiranje informacij o popolnosti in uporabnosti odprtih in prostih projektov težko, drago in zahtevno, ker ni centralne lokacije, oglaševanja ali arhiva dostopnih informacij, ne drži, saj je s pojavom sistemskih integratorjev in svetovalcev s področja odprte in proste programske opreme tudi ta vzrel zapolnjena. Seveda pa moramo tu upoštevati še prednosti realnosti zbranih informacij, možnost enostavnega testiranja in direkten kontakt z razvijalci in skupnostjo, ki omogoča delno izničenje pretiravanja komercialnega oglaševanja.

- **pridobitev**

Navadno programsko opremo enostavno kupimo, vsaj v primeru, da primeren produkt že obstaja. Je pa v večini primerov potrebna še integracija v obstoječe sisteme. Pridobitev odprtega in prostega programskega produkta je, lahko rečemo, brezplačna (zanemarimo medij, internet povezavo, ...) zaradi odstotnosti uporabniških licenc. Poleg tega lahko organizacija ali podjetje samo sodeluje pri popravkih in razvoju dodatne funkcionalnosti.

- **implementacija**

Sicer je ocenitev stroškov implementacije dokaj težavno dejanje, odvisno od pojmovanja širine izraza implementacija. Če predpostavimo, da pod implementacijo upoštevamo korake, potrebne za vzpostavitev informacijskega sistema v svojo polno funkcionalnost, oba modela v tem pogledu dosejata zelo podobno oceno. Razlike nastopijo le v pogledu dosegljivosti, kvalitete in cene implementacije (pokritost trga z določenim znanjem).

- **izobraževanje**

Je v osnovi neodvisno od vrste sistema (lastniški zaprti, odprti). Delna prednost odprte programske opreme je možnost pregleda izvorne kode v primerih, kjer dokumentacije ni, ali je napisana zelo površno. Tovrstna funkcionalnost je sicer zanimiva za zelo majhen del trga, lahko pa ta del trga predstavlja pomembno nišo. Seveda moramo upoštevati tudi dejstvo, da velike komercialne programske hiše ustvarjajo široke izobraževalne procese, dokumentacijo in izobraževalni material, ki je v osnovi lahko dosegljiv. Kljub temu, da odprti projekti izdajajo veliko kvalitetne literature, v določenih pogledih ne dosegajo kvalitete komercialnih ponudnikov (kar gotovo ne drži v primerih kot je O'Reilly literatura in podobni). Vendar se tudi to spreminja s pristopom vedno več velikih podjetij k revoluciji

odprte kode. Na dolgi rok je izredno težko predvideti kakršnokoli razliko v dokumentaciji in izobraževanju, ne glede na vrsto programske opreme. Poseben primer je razvoj programskih produktov v lastni režiji, kjer lahko z eno izmed odprtih ali prostih licenc privabijo k razvoju širšo skupnost razvijalcev, ki bo razvila dobro in kvalitetno dokumentacijo.

- **uporaba in vzdrževanje**

Vsebujeta v večini primerov največji delež stroškov v življenjski dobi projekta in sta tudi del, kjer se pokažejo največje prednosti odprte in proste programske opreme. Dosegljivost izvorne kode omogoča takojšnje popravke napak in odpravo varnostnih lukenj⁶⁵ in hroščov, za razliko od zaprtih sistemov, kjer je proces daljši in veliko bolj počasen ter v nekaterih primerih skoraj nemogoč (v primeru, da se programski paket ne razvija več). Dodatno možnost izdelavi popravkov in dodatkov zagotavlja daljšo življenjsko dobo projektov in inkrementalno reuporabo in izboljšave.

- **opustitev projekta in prehod na novo tehnologijo**

Je pogled, ki je velikokrat zanemarjen pri izračunu TCO, saj opustitev v resnici ni prava uporaba tehnologije. Seveda so informacijski sistemi v večini primerov narejeni za zamenjavo obstoječih sistemov, tako da mora biti ta proces enostaven in natančno določen. Odprta in prosta programska oprema v tej fazi omogoča izdelavo začasnih in postopnih sistemov, kar omogoča enostavnejši prehod. Pomaga tudi pri prenosu podatkov. Podatki pa imajo v večini primerov edino pravo vrednost v tem procesu. Podatkovna transakcija je v večini primerov težaven proces in včasih celo nemogoč, če podatki niso v pravi obliki. Odprta programska oprema se lahko prenaša in prilagaja novim arhitekturam, podatki pa se lahko prenašajo na enostaven in hiter način. V zaprtih sistemih ste zaprti v določen format in prepuščeni na milost in nemilost svojemu ponudniku.

65 Večina nasprotnikov GNU in open source gibanja bo varnost izrabila za enega prvih očitkov proti Linuxu: "Če je programska koda sistema in servisov odprta, to pomeni, da jo lahko vsakdo pregleda in v njej najde varnostne luknje". Seveda pa vam bo večina pravih varnostnih strokovnjakov zagotovila, da "security through obscurity", torej zagotavljanje varnosti s tem, da stvar ostaja skrita in upanje, da je ne bo nihče preizkusil ali po nesreči naletel nanjo, nikakor ni prava oblika varnosti. Varnost je zagotovljena s hitrimi popravki varnostnih lukenj, ki jih zagotavlja stalno testiranje, pregledovanje in izboljšave izvorne kode. To pomeni, da čakanja na varnostne popravke ni več. Revija Wired je takole predstavila to veliko prednost odprtih sistemov: Ko je "Ping of Death" napad zrušil veliko število operacijskih sistemov po celem svetu, je hiter popravek Linuxa zatrl napad na Linux strežnike v nekaj urah. "Nekdo je objavil podrobnosti napada" se spominja Alan Cox, avtor popravka, "tako, da sem napisal popravek in ga v trenutku objavil". Uporabniki ostalih operacijskih sistemov so morali živeti z grožnjo še kar nekaj časa.

V določenih fazah razlike niso očitne. V fazah, kjer odprta in prosta programska oprema zaostaja, se razmere razvijajo tako hitro v pozitivno smer, da bodo kmalu zanemarljive. V fazah, kjer se kažejo očitne prednosti, se kaže, da so odprti sistemi nepremagljivi, kar se tiče majhnosti TCOja. Za zaključek poglavja TCO si lahko stvar pogledamo na praktičnih primerih, ki sicer v določenih pogledih ne sledijo natančno vsem postavkam definiranega TCO modela, vendar so dovolj natančen odraz dejanskega stanja.

Primer 1: Metalka trgovina d.d.

V podjetju Metalka trgovina d.d. so se pred kratkim srečevali z izzivom postopnega uvajanja sistema elektronske pošte in učinkovitega dostopa do Interneta v kombinaciji z povezovanjem oddaljenih lokacij. V prvi fazi so predvidevali povezovanje dveh glavnih lokacij, v nadaljnih fazah projekta pa še povezovanje vseh oddaljenih lokacij. Možnosti za implementacijo so bile naslednje:

1. možnost: Microsoft Small Business Server (4.5 ali 2000)

Izbrana platforma bi omogočala delo do 50 sočasnih uporabnikov v okolju Windows 2000/NT4. Na eni izmed primarnih lokacij bi bil nameščen centralni strežnik z ustrežno infrastrukturo (analogni ali ISDN modemi v sistemu RAS) ali namenski RAS strežnik. Vsaka izmed drugih lokacij bi zahtevala router, svoj ISP dostop in verjetno tudi lokalni NT strežnik.

Tabela 1: Ocena stroškov projekta

<i>Postavka</i>	<i>Cena (DEM)</i>
Strežnik (P III 800, 512 MB RAM, RAID - 3x18GB,NIC)	15,000.00
SBS 4.5 (MS NT srv, EXCHANGE, SQL, RAS,FAX)	2,500.00
licence za uporabo SBS 4.5 - 30 CAL (max. 50)	3,000.00
router INTEL 8110 (2 kom)	3,000.00
modem externi (2 kom)	500.00
vzdrževanje in podpora / letno	3,600.00
predvideni ISP (Internet + impulzi) stroški (2 lokaciji - 1000 DEM mesečno) / letno	24,000.00
Skupaj:	51,600.00

vir: Metalka trgovina d.d. - sektor za informatiko.

Prednosti: enostavna namestitvev, upravljanje in vzdrževanje okolja zaradi znanja systemskega osebja v podjetju, skupni koledar, upravljanje s stiki, sestanki. V grobem je to strežnik za sodelovanje in sporočanje, ki ponuja tudi elektronsko pošto.

Slabosti: cena, omejen le na 50 istočasno priklopljenih uporabnikov (zahteva do 300, kar močno poveča licenčne stroške, pogoste nadgradnje sistema (vsaki 2 leti, ponovna investicija v licence, varnostni problemi, občasno nestabilno delovanje, namestitvev vezana le na en strežnik, kar posledično pomeni v fazi ISDN WAN povezav možno zelo počasno delovanje.

2. Možnost: Microsoft Backoffice server 2000

Velja enako kot za prejšnjo postavko, razen sprememb funkcionalnosti in licenčnih pogojev ter zahtev po strojni opremi. Prav tako, kar se tiče prednosti in slabosti.

Tabela 2: Ocena stroškov projekta

Postavka	Cena (DEM)
Strežnik (P III 800, 1024 MB RAM, RAID - 5x18GB,NIC)	25,000.00
BackOFFICE 2000	6,000.00
licence za uporabo BackOFFICE - 60 CAL	30,000.00
router INTEL 8110 (2 kom)	3,000.00
modem externi (2 kom)	500.00
vzdrževanje in podpora / letno	7,200.00
predvideni ISP (Internet + impulzi) stroški (2 lokaciji - 1000 DEM mesečno) / letno	24,000.00
Skupaj:	95,700.00

vir: Metalka trgovina d.d. - sektor za informatiko.

3. Možnost: Linux in odprti sistemi

Na trgu so ponudniki rešitev z uporabo LINUX -a, ki ponujajo integracijo strojne in programske opreme s svojim znanjem. Eden takih ponudnikov je podjetje Agenda Open Systems s produktom Internet/Gateway strežnik/usmerjevalnik. Sistem vsebuje vse potrebno za povezavo večjih enot in omogoča izmenjavo elektronske pošte znotraj posamezne lokacije (kjer je nameščen Internet GateWay, kar zmanjšuje stroške Interneta in pospešuje delo) in med večjima enotama. Satelitske (manjše enote) se povezujejo klicno (na zahtevo). Omogočen je enoten dostop preko ene klicne povezave, kar na vsaki posamezni lokaciji pomeni samo stroške impluzov. V vsak priklop je vključena tudi osnovna vzdrževalna pogodba, kar zagotavlja varnosti in dovolj učinkovite ter hitre odzivne čase.

Tabela 3: Ocena stroškov projekta

Postavka	Cena (DEM)
Internet GateWay R120 (PIII, software RAID, 2x ISDN, WAN, routing)	4,000.00
Internet GateWay R110 (PIII, software RAID, 1x ISDN, WAN, routing)	3,000.00
modem externi (2 kom)	500.00
vzdrževanje in podpora v ceni priklopa	
predvideni ISP (Internet + impulzi) stroški (2 lokaciji - 1000 DEM mesečno) / letno	15,000.00
Skupaj:	22,500.00

vir: Metalka trgovina d.d. - sektor za informatiko.

Prednosti: cena, neomejeno število uporabnikov, kvalitetna preizkušena rešitev, vzdrževalna pogodba v ceni priklopa na Internet, enostavno širjenje funkcionalnosti, router funkcija implementirana, velike možnosti nadzora in nadgradenj. Lokalni poštni strežniki na vsaki posamezni lokaciji, integriran RAS sistem.

Slabosti: izobraževanje kadra za uporabo novega operacijskega sistema (Unix), še ni skupnega sistema upravljanja z naslovi. Skupen sistem upravljanja z naslovi se implementira s pomočjo OpenLDAP sistema, ki ga podpirajo vsi standardni odjemalci elektronske pošte.

Že v postopku osnovne investicije odprti in prosti sistemi v kombinaciji z znanjem systemskega integratorja omogočajo do 50% nižje stroške. Razlogi so licenčni stroški pri konkurenčnih platformah, zahteve po večji zmogljivosti strojne opreme in večjih stroških vzdrževanja, predvsem pa v stroških nadgradenj.

Kljub zgoraj navedenim razlogom pa se še vedno večina podjetij odloča za lastniške zaprte produkte. Razloge lahko iščemo predvsem v monopolnem položaju določenih ponudnikov in njihovi dolgi in številčni prisotnosti tudi na slovenskem trgu. Pomembno dejstvo predstavljajo tudi dosedanje investicije podjetij v znanje zaposlenih, ki so se v večini primerov šolali po programih velikih programskih hiš. Vsak prehod in pridobitev novega znanja predstavlja velik finančni zalogaj za podjetja, pa tudi velike zahteve po pridobivanju popolnoma novega znanja za zaposlene v informacijskem sektorju.

Trendi⁶⁶ kažejo, da se v večini primerov za odprte in proste rešitve odločajo večja podjetja (SiMobil⁶⁷, Mobitel⁶⁸, Mercator⁶⁹, Metalka⁷⁰, Inles⁷¹, Vodovod Kanalizacija⁷², Pošta Slovenije, Cimos⁷³, ...). Razloge za tovrstne odločitve lahko v določeni meri iščemo v ceni odprtih in prostih operacijskih sistemov, vendar se v večji meri kaže uporaba tovrstne tehnologije v kompleksnosti rešitev, ki jih zahtevajo in v dobri informiranosti vodstvenega kadra (vodje informatike), o prednostih in lastnostih odprtih in prostih sistemov. Manjša podjetja odprte in proste operacijske sisteme, ki jih uporabljajo v procesu svojega poslovanja, v večini primerov zanj niti ne vedo, saj so v uporabo prišli preko rešitve na ključ (usmerjevalniki, datotečni strežniki, požarni zidovi, ...). V osnovi jim je pomembna rešitev in ne platforma, na kateri rešitev teče. Seveda pa tudi v Sloveniji v zadnjih dveh letih lahko spremljamo očitno spremembno mišljenje vodstvenega kadra informacijskih sektorjev podjetij, ki danes v večini primerov upošteva Linux kot eno izmed alternativ svojih izzivov, medtem ko je prodajni pristop še pred letom in pol zahteval osnovno predstavjanje prednosti odprtih in prostih operacijskih sistemov in v primerjavi s prodajo zaprtih lastniških rešitev veliko več truda pri prepričevanju in utemeljevanju predstavljene rešitve.

7. Vplivi odprte in proste programske opreme

Obstoj proste in odprte programske opreme ima kompleksen in pomemben vpliv na ekonomijo kot celoto. Dejstvo je, da informacijska tehnologija postaja dosegljivejša in da je gradnik novih trgov. Obstoj populacije naprednih uporabnikov (uporabnikov, ki so tudi razvijalci programske opreme in informacij) igra odločilno vlogo pri razvoju novih projektov, naprav in storitev. V posebnih sektorjih kot je izobraževanje, zmanjševanje TCO-ja, ki ga omogoča uporaba odprtih produktov in uporabo kreativnosti skupnosti in posameznikov, dajejo in omogočajo večjo produktivnost javnih servisov. Dinamična stabilnost odprtih projektov, torej velika verjetnost dolgoročnih inovacij, omogoča možnost učinkovitih investicijskih strategij v uporabniški industriji.

66 Vir: Agenda PE Ljubljana, sektor prodaje in svetovanja.

67 SiMobil [URL: <http://www.simobil.si>], 10.09.2001.

68 Mobitel [URL: <http://www.mobitel.si>], 10.09.2001.

69 Mercator [URL: <http://www.mercator.si>], 10.09.2001.

70 Metalka [URL: <http://www.metalka.si>], 10.09.2001.

71 Inles [URL: <http://www.inles.si>], 10.09.2001.

72 JP Vodovod Kanalizacija Ljubljana [URL: <http://www.vo-ka.si>], 10.09.2001.

73 Cimos [URL: <http://www.cimos.si>], 10.09.2001.

Medtem ko je realnost teh makro ekonomskih vplivov dobro dokumentirana, njihova realna številčna ocenitev (z upoštevanjem potencialnih negativnih makro-ekonomskih vplivov) danes še ni možna zaradi pomanjkanja dejanskih statističnih podatkov glede uporabe odprte in proste programske opreme in njenih povezav z ekonomskimi in neekonomskimi aktivnostmi. Razvoj pravih makro ekonomskih indikatorjev je potreben, če želimo, da bo vpliv odprte programske opreme na ekonomijo in svet kot celoto lahko stal na dejstvih. Čas bo pokazal svoje, verjamem, da v zelo pozitivnem smislu, bolj kot pa je bil prikazan tu.

7.1 Odprta programska oprema in nova konkurenca na trgih programske opreme

Standardno vprašanje, ki se postavlja pri razvoju odprte programske opreme pa tudi v sami skupnosti je, ali so odprti in prosti razvojni modeli bolj ali manj učinkoviti od zaprtih modelov. V večini primerov je lahko zaprta programska oprema dobičkonosna le, če ne obstaja noben prost produkt s podobnimi karakteristikami. Model proste in odprte programske opreme zagotavlja, da lahko ponudi produkte boljše in enake kvalitete kot katerikoli komercialni tekmeč, ko enkrat vstopi na določen nišni trg z dovolj veliko energijo. Primeri Apache-ja⁷⁴, Gnat-a⁷⁵ in mnogih drugih kažejo na dejansko stanje.

Če vse to drži, pomeni, da bi se odprti in prosti razvojni modeli vzdrževali sami. To pomeni, da bi se s časom nabralo dovolj virov za ohranjanje stabilne ravni razvoja. Seveda pa virov tu ne smemo upoštevati le kot denar (čas, znanje, ...). Če trenutno že obstaja vodilen izdelek na trgu, je edina prava odločitev za podjetje odprt in prost model. V drugačnem primeru verjetno ne bo doseglo nobenih omembe vrednih rezultatov, ker je tekmovanje na področju komercialnih pravil izredno težko, predvsem na že uveljavljenih trgih. Z odprto kodo pa se celotno okolje in pravila spremenijo. Vodilni proizvalalec nima več vseh prednosti. Veliko strank bo dovolj motiviranih (zaradi nižjih stroškov, boljšega produkta, večjega nadzora nad smerjo razvoja ali kaj tretjega), da preizkusi odprt produkt. To sili vodilnega proizvajalca k spremembi njegove strategije, k izboljšanjem produkta in storitev, k prestopom na odprte in proste modele ali z dodatno vrednostjo. To se dogaja tudi danes in dosega največje informacijske igralce kot je večina Unix ponudnikov, ki tekmujejo z Linux operacijskim sistemom ali katerikoli spletni strežnik, ki poizkuša tekmovati s spletnim strežnikom Apache.

74 <http://www.apache.org>

75 <http://www.gnat.com>

7.2 Intelektualna lastnina

Kot je ponavadi primer v katerikoli informacijski tehnologiji so stvari povezane z intelektualno lastnino, primarnega pomena tudi za prosto in odprto programje. Med štirimi mehanizmi, ki jih daje mednarodno pravo pri zaščiti intelektualne lastnine, so samo trije (copyright, patenti in blagovne znamke), ki so lahko primerni za programsko opremo. Četrta, poslovna skrivnost, ni primerna za odprto in prosto programsko opremo, ker zahteva prikrivanje (preprečuje dostop do izvirne kode) ali prepoveduje popravke, preprodajo in distribucijo del, ki izhajajo iz izvirnika.

V tem delu bomo pregledali nekaj vplivov patentnega in avtorskega prava v uporabi pri odprtih in prostih produktih. Blagovne znamke imajo v svetu odprte in proste programske opreme popolnoma enako vlogo kot v kateremkoli drugem poslu in jih bomo zato v tem poglavju izpustili. Seveda pa je potrebno omeniti, da so blagovne znamke jedro mnogih podjetij, ki se ukvarjo z odprtim in prostim programjem in z blagovno prepoznavnostjo svojih storitev.

Žal pa v veliko državah (predvsem pa v ZDA) avtorsko pravo ni več edini način zaščite programske opreme. Vse pogostejše v te namene uporabljajo patentno pravo.

7.2.1 Odprto, prosto programje in avtorsko pravo

Avtorsko pravo je največkrat uporabljena metoda zaščite programskih produktov. Odprte in proste licence je potrebno pravno upoštevati, ker na tak ali drugačen način uporabljajo enega izmed pravil avtorskega prava. Razlog je enostaven: avtorsko pravo praviloma ne dopušča redistribucije (ali celo uporabe) programa. Edini način, da dosežemo tudi redistribucijo, je z dodatkom posebnih pravil v licenci. Ta redistribucija mora seveda upoštevati različne pogoje od priznanja in omembe avtorja (kot BSD licenca) do obveze, da se vsa dela, ki izhajajo iz izvirnika, izdajo pod enakimi pogoji kot originalna licenca (GPL licenca).

Večina odprtih in prostih licenc je bila zasnovana glede na pravo Združenih držav Amerike. Šele pred kratkim so bile narejene raziskave o uporabi in pravnosti le teh tudi v drugih državah. Te študije so izrednega pomena za gibanje odprte in proste programske opreme, ker so vsi odprti in prosti modeli vsaj v večjem delu, odvisni od pravilnosti licence in možnosti njihove izrabe in izpodbijanja na sodiščih. Prve ugotovitve kažejo, da GPL (in večina podobnih licenc), veljajo v večini evropskih držav. V tej smeri raziskav je potrebno še veliko dela, tako da se bo gibanju odprte in proste programske opreme lahko zagotovila močna pravna osnova, posebej v deželah izven Amerike.

Posebno poglavlje obsegajo področja, kjer se odprta in prosta programska oprema srečujeta in integrirata s komercialnimi programi. V nekaterih primerih so bila podjetja prisiljena omogočiti dostop do natančnih informacij delovanja programa ali operacijskega sistema, da so omogočila razvijalcem nadaljen razvoj in integracijo programskih komponent v njihove sisteme in programe. Tovrstne informacije so bile navadno avtorsko zaščitene in prodane le registriranemu razvijalcem z natančno kontrolo nad informacijami, kar pa ni skladno z večino odprtih in prostih licenc.

7.2.2 Odprta, prosta programska oprema in programski patenti

Programske patente, posebej tiste, ki so dani trivialnim algoritmom in ki jih lahko enostavno ponovno izumimo, predstavljajo hudo grožnjo posameznim odprtim in prostim projektom in manjšim organizacijam, ki si ne morejo privoščiti stroškov zagovarjanja ali izpodbijanja patentnega prava. Ironično je situacija še bolj pomembna za svet odprte in proste programske opreme kot za zaprt model komercialnih ponudnikov, saj je programska koda dosegljiva lastnikom patentov.

V večini primerov si podjetja in posamezniki prizadevajo pridobiti ekskluzivno pravico uporabe določenih tehnologij preko patentov. V zadnjem času lahko spremljamo vedno več patentov na popolnoma osnovne mehanizme in procedure. Verjamem, da je to potencialno izredno nevarna praksa in to ne le za odprto in prosto programsko opremo, ampak za celotno programsko in informacijsko industijo. Relativno dolga življenjska doba patentov in obsežnost nekaterih izmed njih je posebej zaskrbljujoča. Dodatno ni dovolj nadzora nad obstojem del, ki že obstajajo. Ti patenti so lahko uporabljeni kot orožje proti konkurenci, posebej manjšim, ki si ne morejo privoščiti pravnih stroškov za izpodbijanje tega.

Do sedaj je bila predstavljena množica neumnih in nepotrebnih patentov. Npr. eden izmed novejših primerov je patent metode za odpravljanje Y2K problemov, ki uporablja očitne in široko znane pristope. Dodaten primer je patent na izboljšave preglednosti fontov na tekočih kristalih. Ta tehnika ni bila le dobro poznana in uporabljena na starih Apple II računalnikih, ampak predstavljena v mnogih tekstih.

Svet odprtega in prostega programja je še posebej občutljiv na patentne napade, ker ima le malo število organizacij odprtega in prostega programja dovolj finančne moči, da se bori proti patentnim tožbam. Če je patent zastavljen preveč široko, onemogoča tudi razvoj proste alternative.

7.3 Vplivi in možnosti uporabe odprte in proste programske opreme v izobraževalnem sistemu

Prvi člen gradnikov informacijske tehnologije je gotovo izobraževalni sistem in učni program določene države, saj znanje pridobljeno v procesu šolanja, določa zahteve in želje tudi v nadaljnjem življenju, predvsem pa delovnem okolju (npr. uporaba določenega programskega paketa). Popolnoma razumljivo je, da mora sodobni človek obladovati vsaj osnovno znanje iz področja osnov informacijske tehnologije, uporabe operacijskih sistemov, uporabe pisarniških programskih paketov in Interneta. Seveda, ker nismo minimalisti, pričakujemo tudi sposobnost logičnega razmišljanja, logičnega pristopa k problemom (računalništvo, matematika, programiranje so v svojem bistvu logični problemi, torej si želimo osebe, ki bodo na inovativen, logičen način znali reševati probleme). Seveda je eden izmed ciljev izobraževalnega sistema tudi ustvarjati razvijalce, ne le uporabnike (torej tisti, ki bodo gibalno prihodnjega razvoja informatike, ne pa le uporabniki in sestavljalci kock, ki jim jih bo na mizo ponudil nekdo drug in seveda s tem krojil način in smeri njihove uporabe).

Vse naštetu zahteva določeno investicijo v dovolj veliko število računalnikov v računalniški učilnici (cena strojne in programske opreme), prost dostop do priročnikov, literature, znanja, programske opreme, orodij (želimo si vsem ponuditi enake možnosti, ker prosta dosegljivost orodij omogoča izdelavo in razvoj lastnih testnih sistemov in učenje na praktičnih primerih). Izkušnje kažejo, da večini, ki jih stvar seveda zanima, ni dovolj le nekaj ur tedensko, ki jim jih ponuja šolski sistem, ampak bi radi raziskovali sami. Predpostavimo, da danes na tak ali drugačen način skoraj vsaka družina lahko zbere dovolj sredstev za en računalnik. Investicija pa se v večini primerov ustavi pri programski opremi. Z uporabo določene drage programske opreme v šolskem sistemu, pa naj bodo to pisarniške aplikacije, risarski programi, razvojna orodja ali kaj tretjega, spodbujamo udeležence izobraževalnega procesa k kriminalnim dejanjem, saj je skoraj edini način dostopa do tovrstne programske opreme možen le preko kršenja avtorskih pravic izdelovalcev. V nobenem primeru ne smemo zanemariti tudi investicijo v izobraževanje izobraževalnega kadra. Zaključek je očiten. Manj denarja za licence, pomeni več denarja za investicije v strojno opremo in izobraževanje predavateljev.

Odprta in prosta programska oprema vsebuje ključ do ohranjanja in razvijanja raziskovalne, šolske in izobraževalne odličnosti, ki se vedno bolj zliva z informacijsko tehnologijo. Pomemben je torej dostop do znanja in informacijske tehnologije, ker se le na ta način zagotavlja, da ji bomo sledili in jo tudi razvijali, kar posledično pomeni, da je le malo področij, ki bi lahko z uporabo odprte in proste programske opreme pridobili več kot izobraževanje.

Če si pogledamo računalniško pismenost z vidika udeleženca v izobraževalnem procesu, je računalniško opismevanje vedno bolj videti kot tečaj Microsoft⁷⁶ programskih produktov. Nekateri bodo trdili, če imamo Microsoft radi ali ne, še vedno moramo učiti dijake in študente uporabo programske opreme, s katero se bodo srečevali in to žal pomeni Microsoft. To je tudi v interesu podjetij, saj jim tako ni potrebno namenjati denarja za dodatno izobraževanje. S tem se ne morem strinjati in trdim, da je pristop napačen na srednješolskem pa tudi univerzitetnem področju. Dodatno se ne strinjam s tezo, da pripomore k večjemu uspehu gospodarskega sektorja, ki mu ni potrebno plačevati dodatnega izobraževanja.

Ozka usmeritev na določene programske produkte ni pravilna zaradi naslednjih razlogov:

- če predpostavimo, da demokracija sloni na informiranih državljanih, potem bi srednje šole in univerze morale imeti obvezo, da svoje učence seznanijo s konceptualnim razumevanjem informacijske tehnologije, torej tiste, ki gre preko meja tega, kateri gumb je potrebno pritisniti v določenem programu. Potrebujemo torej učence, ki bodo vedeli dovolj o informacijski tehnologiji in problemih, s katerimi se ukvarja, vključno z vedno večjo nevarnostjo zmanjševanja pravice po privatnosti in tveganja, ki jih prinaša programska industrija z zahtevami po novih intelektualnih pravicah in pravilih.
- podjetja bi res nekaj pridobila z že izobraženo delovno silo, vendar se ves pozitiven učinek zabriše s stalno potrebo po dragih nadgradnjah in popolnoma specifičnim znanjem, v primeru, da se nekje zamenja en ukaz.
- hitrost razvoja informacijske tehnologije je danes taka, da je šolanje, ki uči, "kateri gumb je potrebno pritisniti", zastarel prej kot v štirih letih, torej v času, ko poteka proces šolanja.

V svetu, kjer vedno bolj velja načelo: svoboda tiska pripada tistim, ki so njegovi lastniki, imamo možnost, da z odprto in prosto programsko opremo naredimo svet boljši, omogočimo dostop do znanja in orodij vsem in vsakomur ter omogočimo razvoj raznovrstnih znanj. V svetu, ki je vedno bolj voden in nadzorovan s strani velikih podjetij, imamo možnost ustvariti individuum, ki bo imel pomembno mesto v družbi in bo razmišljal s svojo glavo in bo del pomembne skupnosti, skupnosti človeštva in odprte in proste programske opreme.

76 [URL: <http://www.microsoft.com>], 17.09.2001

8. Nekaj specifičnih vplivov odprte in proste tehnologije na razvoj in način uporabe informacijske tehnologije

Obstaja nekaj specifičnih vplivov odprte in proste tehnologije na način razvoja in uporabe informacijske tehnologije, ki zaradi svoje pomembnosti zahtevajo bolj natančen opis.

8.1 Promocija de-facto in de-jure standardov

Pomembna lekcija, ki jo uči zgodovina, vsaj na področju informacijske tehnologije, je, da imajo vsiljeni standardi veliko težav pri vzpostavljanju prave baze uporabnikov. V tem pogledu lahko dosegljivost odprte in proste referenčne implementacije postane način za razširjanje uporabe protokola. Odlični primeri tega dejstva so mnogi protokoli in večina infrastrukture. Ta je bila odločilnega pomena pri razvoju Interneta. Skladovnica protokolov TCP/IP, ki je temelj povezovanja v Internetu, je svoj uspeh dosegel zaradi proste dosegljivosti in odprtosti. Osnove protokola je razvila v Berkleyu skupina CSRG⁷⁷. Poštni strežnik Sendmail⁷⁸ ki še vedno ostaja, kot referenčna implementacija za SMTP⁷⁹, ESMTP⁸⁰ in napredne poštno funkcionalnosti, je le še eden med primeri. BIND, tudi razvit v Berkeley je še vedno najbolj uporabljena programska implementacija DNS (imenski prostor Interneta) protokola. Dosegljivost odprte in proste različice XWindows⁸¹ sistema je bila odločilnega pomena pri vzpostavitvi X11 kot standardnega protokola za grafična Unix okolja.

Pomembnost referenčne odprte in proste implementacije se kaže v naslednjih vidikih:

- referenčna koda omogoča hitro razširjanje in uporabo protokola komercialnih in nekomercialnih delov
- omogoča primerjavo lastnih različic, testiranje skladnosti in odpravo nekompatibilnosti v primeru, ko protokol ni popolnoma natančno določen.

Referenčna implementacija lahko pomaga tudi v primerih "agresivne odcepitve" protokola. Npr. če se določeno podjetje odloči, da bo naredilo delno nekompatibilno različico istega protokola zaradi prednosti, ki jih bo pridobil proti konkurenci, in razlogov zapiranja uporabnikov v lastne zaprte protokole, je obstoj referenčne implementacije vedno tisti, ki lahko pomaga pri preprečevanju tovrstnega deljenja, seveda le pod pogojem, da vsi ostali ponudniki upoštevajo dogovorjena pravila, kar pa ponavadi v večini primerov velja.

77 CSRG [URL: <http://www.cs.toronto.edu/csri/>], 18.09.2001.

78 Sendmail [URL: <http://www.sendmail.org>], 18.09.2001.

79 SMTP (Simple Mail Transfer Protocol) definiran v RFC (Request for comment) 821.

80 ESMTP (Extended Simple Mail Transfer Protocol).

81 XFree86 [URL: <http://www.xfree86.org>], 18.09.2001.

Kot je bil v preteklosti pomemben nadzor nad operacijskim sistemom in standardi formatov izmenjave podatkov, bo v prihodnosti izrednega pomena dosegljivost odprtih podatkovnih vmesnikov in omrežij za elektronsko poslovanje in izmenjavo vsebine, med različnimi servisi Interneta. Ti aplikacijski protokoli bi morali ostati odprti in imeti odprto in prosto referenčno implementacijo, ki jo lahko uporabi vsako podjetje ali posameznik glede na svoje želje in zahteve. Drugače bo vsakdo, ki bo želel uporabljati eno izmed teh storitev, prisiljen v nakup zaprte komercialne opreme podjetja, ki je standard postavilo. Ko te storitve vedno bolj postajajo del osnovne infrastrukture v vsakodnevni uporabi posameznika in podjetij, postaja vedno bolj kritično dejstvo, da ne smejo biti pod nadzorom kateregakoli podjetja.

8.1 Razpršitev in razširjenost informacijske tehnologije

Odprta in prosta programska oprema obsega tako količino kode in programskih sistemov, da je možno sestaviti popolnoma proste sisteme le z uporabo odprtih in prostih komponent brez komercialne programske opreme. To lahko pomeni veliko prednost, če lahko zmanjša razmak med tistimi, ki imajo in tistimi, ki nimajo. Dodatno pomeni dostopnost programske kode in možnost svobode popravkov, specifično internacionalizacijo in lokalizacijo projektov. Možno je, naprimer prirediti program tujim znakom in alternativnim vnosnim metodam. Ta fleksibilnost je pomembna tudi pri prireditvi programov za zelo specifične potrebe.

8.2.1 Dostopnost varnejših sistemov

Vzpon Interneta in ostalih cenovno ugodnih komunikacijskih tehnologij prinaša neverjetne možnosti za ustvarjanje prave 'informacijske skupnosti'. Seveda prinaša tudi mnogo dvomov, predvsem kar se tiče varnosti večine informacijskih sistemov. Dobro znano je, da je večina strežniških sistemov (kaj šele namiznih sistemov) zelo ranljivih (vdori in ostali problemi povezani z varnostjo), kar lahko pomeni veliko tveganje, če ni zagotovljeno dovolj dobro varovanje.

Zelo hitro lahko opazimo, da ima odprta in prosta programska oprema očitno prednost pred zaprtimi sistemi, ker je možna hitra identifikacija in odprava potencialnih varnostnih problemov. Obstajajo poštni sezname in nadzorne (ang. auditing) skupine, ki aktivno pregledujejo in sledijo varnostni problemom.

Na praktičnih primerih je bilo dokazano, da "security through obscurity", kar pomeni, skrivanje določenih delov sistema, ne prinaša večje stopnje varnosti kot odprti sistemi. Celo danes, ko Internet obsega le del sveta, pomenijo crackerji, virusi in Internet črvi pomembno privatno in denarno grožnjo.

Dodatni vidik je ustvarjanje odprtih in prostih testnih paketov, ki ugotavljajo skladnost implementacije z določenimi standardi. To pomaga pri testiranju programskih paketov, ki se hitro spreminjajo in zagotovi, da se programski paket obnaša pravilno in glede na specifikacije.

8.3 Vpliv uporabe odprte kode v manj razvitih državah

Vpliv odprte in proste programske opreme v manj razvitih deželah je lahko veliko večji kot v državah, kjer je informacijski sektor dobro razvit. Med številnimi razlogi lahko izpostavimo naslednje:

- enostaven dostop do programskih produktov
Odprta in prosta programska oprema je lahko dosegljiva posameznikom in podjetjem. Enostavno se lahko začne distribucija in uporaba programske opreme, saj v ceno ni vključenih per-copy stroškov. Programska oprema tako postane dosegljiva povsod.
- cenovno učinkoviti prenos programske tehnologije
Razvite države lahko cenovno učinkovito prispevajo manj razvitim državam s pomočjo pri uporabi odprtih in prostih tehnologij.
- direkten dostop do informacijske tehnologije
Dostopnost vrhunske, najkvalitetnejše informacijske tehnologije brez plačila dragih licenčnin. Ponavadi države nimajo enostavnega dostopa do teh tehnologij, vsaj ne v začetni fazi razvoja. Dobijo le uporabniške programe po visokih cenah in brez dostopa do tehnologije.
- možnosti pri sodelovanju razvoja naprednih tehnologij
V svetu odprte kode lahko vsak razvijalec (posameznik ali podjetja) dostopa do kateregakoli odprtega, prostega projekta (ob upoštevanu minimalne infrastrukture kot je Internet dostop). Zanimivo je, kako so pomembni odprti in prosti projekti vodeni iz držav z malo tradicije ali brez nje pri naprednem razvoju programske opreme. V prihodnosti lahko pričakujemo še več primerov razvoja odprtih projektov zaradi manj razvitih držav. Ker so osnove na voljo in dostopne komurkoli, in ker so distribucijski kanali dostopni vsem projektom, imajo ti projekti veliko možnosti za uspeh.

Za zaključek lahko rečemo, da prosta in odprta programska oprema spreminja podobo sveta in omogoča vsakomur ne le prednosti, ki jih prinaša prosto programje, ampak tudi sodelovanje pri razvoju in popravkih. Seveda je potrebno opozoriti na dejstvo, da boljša infrastruktura v razvitih državah (predvsem kar se tiče komunikacijske infrastrukture, izobraževalnega sistema, razpoložljivosti in dosegljivosti tehničnega osebja, etc..) ponuja številne prednosti pri razvoju in uporabi prostih programskih produktov.

8.4 Nekaj možnih scenarijev za prihodnost odprte in proste programske opreme

Napovedovanje možnih smeri razvoja programske industrije je skoraj nemogoče kljub temu pa bomo poizkusili predvideti nekaj scenarijev, ki bodo pokazali, kako bi lahko pomoč države ali kakšne druge organizacije vplivala na prihodnost odprtega in prostega gibanja. V večini primerov je gibanje odprte in proste programske opreme že dokazalo, da se lahko razvija samo. Dejstvo je, da se razvija zelo dobro in to brez pomoči vlade in velikih korporacij.

Predvidimo naslednje scenarije:

- nobene pomoči države ali drugih institucij
- omejena podpora
- agresivna podpora

8.4.1 Scenarij brez pomoči države ali drugih institucij

Do danes ni bilo skoraj nobene podpore države pri razvoju odprte in proste programske opreme. Ta scenarij predvideva, da se bodo stvari razvijale na zelo podoben način.

Pomembno dejstvo je, da odprti in prosti sistemi sestavljajo velik del našega računalniškega okolja predvsem na področju strežnikov in razvojnih orodij, kjer igrajo pomembno vlogo v infrastrukturi informacijske tehnologije. Kot ugotavljajo podjetja, je možno tudi živeti na račun odprte in proste programske opreme in storitev povezanih z odprtimi in prostimi sistemi.

V primeru, da država ne pristopi k podpori odprtih in prostih sistemov, lahko pričakujemo, da bodo le privatna podjetja in posamezniki vodili razvoj v tej smeri. Ker odprto in prosto gibanje dokazuje, da je sposobno samovzdrževanja in rasti v ekonomskem in v tehnološkem smislu in ker tekmuje z vodilnimi na številnih visoko tehnoloških trgih, ni dvoma v njegov uspešen razvoj tudi v prihodnosti.

Po vsej verjetnosti bo odprta in prosta programska oprema uporabljena kot standarden del infrastrukture večine držav brez potrebe po specifičnih obvezah. Namesto nakupa zaprtega sistema se bodo nekatere vladne službe odločile za nakup odprtih sistemov in strežnikov. Ta trend se aktivno dogaja v kar nekaj državah (vsaj v Evropi). Odprta in prosta programska oprema bo v večini primerov izbrana predvsem zaradi svojih ekonomskih in tehničnih vplivov kot boljša tehnološka ali cenovna alternativa zaprtim sistemom. V tem primeru ne bo nobenih specifičnih prednosti za države ali skupnosti, razen direktnih prihrankov pri stroških in licencah.

V tem scenariju (in delno tudi v drugih) je nekaj groženj, s katerimi se bo moralo odprto in prosto gibanje spoprijeti. Med njimi so:

- FUD⁸² (fear, uncertainty, doubt) tehnike, uporabljene v podjetjih in organizacijah, ki zagovarjajo zaprt model programske opreme in se seveda bojijo izgube svojega tržnega deleža in vodilnega položaja. Do danes ni bilo resnih problemov, razen manjših poizkusov. Seveda pa se lahko situacija spremeni v prihodnosti, ko se več in več virov posveča temu namenu.
- izguba globalne vizije ljudi, ki "vodijo" odprto in prosto gibanje. Ko več in več ljudi uporablja odprto in prosto programsko opremo, obstaja tveganje, da jih večina v osnovi ne razume načina delovanja in razvoja odprtih sistemov in vseh lekcij, ki se jih je skupnost naučila v zadnjih nekaj desetletjih.
- pravne omejitve, ki jih lahko postavi pravni sistem. Poglavlje programskih patentov je ena izmed pomembnejših pravnih ovir, s katero se vedno pogosteje srečuje odprta in prosta programska oprema.

Nekaj izmed teh problemov se lahko odpravi s pomočjo državnih institucij, kar bomo predstavili kasneje.

8.4.2 Scenarij delne podpore

V tem scenariju je pravilno, da se vprašamo, kaj je tisto najpomembnejše, kar lahko pridobi skupnost, družba, država na splošno od odprtega in prostega programja in kako ga lahko uporabijo za svoje prednosti, kot pa obrnjeno (kaj lahko odprto in prosto gibanje pridobi od podpore države)

V tem primeru bi investirali nekaj časa in virov v ocenitev zmožnosti odprte in proste programske opreme in področij zanimanja ter določitve ovir, ki se lahko postavljajo pri uvajanju odprte in proste tehnologije. S časom bi določili nekaj strategij odprtih in prostih projektov, kjer bi si želeli prispevati in bi bolj jasno razumeli prednosti v smislu fleksibilnosti, uporabnosti v celotnem življenjskem ciklu in prilagodljivosti. Izkaže se lahko, da bi interes pri uporabi odprte in proste programske opreme sestavljal kar nekaj komponent infrastrukture. Na dolgi rok bi to produciralo rezultate, predvsem v smislu večjega sprejemanja in naklonjenosti odprte in proste programske opreme v družbi na splošno, zaradi vpliva, ki ga ima uporaba v državnih ustanovah. Če končno spoznajo prednosti odprtih in prostih

82 FUD - Izraz je določil Gene Amdahl, potem, ko je zapustil podjetje IBM in začel z lastnim podjetjem. FUD pomeni strah (fear), negotovost (uncertainty) in dvom (doubt), ki so ga je marketinška ekipa IBM-a vnašala v misli potencialnih kupcev konkurenčnih [Amdahl] produktov. Bistvo ideje je v tem, da je varneje kupovati IBM-ovo opremo kot katero drugo. Po letu 1990 se izraz FUD vedno bolj povezuje s podjetjem Microsoft, ki je na novo definiralo načine njegove uporabe kot konkurenčnega orožja.

modelov, bodo po vsej verjetnosti premagali tudi prihodne probleme (prevsem tiste povezane s pravnim sistemom). Slovenijo lahko uvrščamo v zelo začetno fazo scenarija delne podpore odprte in proste programske opreme. Z javnim razpisom⁸³ za sofinanciranje projektov razvoja, uporabe, lokalizacije in razširjanja programske opreme temelječe na odprti kodi, je tudi Slovenija naredila prvi korak pri podpori odprte in proste programske opreme.

8.4.3 Agresivni scenarij

Popolnoma nemogoče je določiti in predvidevati, kaj 'agresivno' pomeni, zaradi določenih pravnih pravil, ki lahko tisto, kar je danes agresivno, postavijo za popolnoma normalen in vsakodneven pojav. Kot primer lahko navedemo pravni okvir, ki bi določal prioriteto uporabo odprtih in prostih rešitev v državnih in javnih ustanovah, ko bi to bilo mogoče. Podoben primer bi bila aktivna podpora in promocija razvoja odprtih in prostih alternativ zaprtim sistemom na področjih, ki bi jih določili kot strateška zaradi socialnih ali ekonomskih razlogov.

To bi ustvarilo ogromen trg za svetovanje na področju odprtih in prostih sistemov in rešitev. Tovrstna podpora bi verjetno imela velik vpliv tudi na ravnotežje pri uvozu/izvozu informacijskih produktov, ki v večji meri izhajajo iz ZDA.

Če bo odprta in prosta programska oprema v resnici spremenila celotno področje informacijske industrije, bi se podpora države (Evrope, ali kakšne druge države) lahko spremenila samo v več koristi. Če odprta in prosta programska oprema ni le nova moda in muha enodnevnica, bi bil njen vpliv zelo podoben vplivu Internet tehnologij v zadnjem desetletju.

8.5 Prihodnost odprtih in prostih sistemov po predvidevanjih Gartner

Group

Eno izmed najpogosteje zastavljenih vprašanj poslovnih uporabnikov Linuxa in prostih, odprtih sistemov je: "Kdaj bo Linux in posredno z njim tudi odprta in prosta programska oprema pripravljena za uporabo v najzahtevnejših produkcijskih okoljih?" Širina medijskega poročanja o Linuxu je velika in v večini primerov pozitivna. Podpora in trženje Linuxa večjih proizvajalcev tudi ne zaostaja veliko. Razprava o prednostih in slabostih odprte in programske opreme dosega svoj vrhunec z Microsoftovim zagovarjanjem komercialnega programskega modela in očitanjem uničevanja intelektualne lastnine, ki je ogrožen zaradi uporabe odprtih in prostih sistemov. Zagovorniki odprtih in prostih sistemov na drugi strani očitajo Microsoftu škodljivo delovanje in zaviranje razvoja na informacijskem področju, za

83 Ministrstvo za informacijsko družbo [URL: <http://www2.gov.si/mid/mid.nsf>], 09.09.2001.

katerega ni več prostora v odprti, Internet infrastrukturi. Uporabniki, razvijalci in ponudniki programske opreme so zaradi vseh teh debat zmedeni, kar je po eni strani tudi Microsoftov cilj (prej omenjeni FUD).

Gartner⁸⁴ verjame, da je Linux evolucija in ne revolucija pri promociji novih tehnologij, s čimer so lepo določili trende razvoja Linuxa in odprte, proste programske opreme. Njihov model določa naslednja obdobja rasti in razvoja Linuxa:

- prva faza (1998-2000)
Prvi uporabniki optimistično napovedujejo svetlečo prihodnost Linuxa kot izredno široko uporabljene razvojne platforme.
- druga faza (2000-2005)
Linux se iz preveč optimističnih predvidevanj uveljavi na določenih IT področjih kot uspešen tekmelec Unix in NT/Windows 2000 strežnikom kot omrežni strežnik ali spletne strežniške farme.
- tretja faza (2003-2005)
Neodvisni ponudniki programske opreme (Independent Software Vendors) razširijo prisotnost Linuxa v elektronskem poslovanju, medtem ko uporaba Linuxa v visoko zmogljivih podatkovnih okoljih ostaja omejena, razen v zelo raznovrstnih okoljih.

S predvidevanjem Gartner Group se lahko strinjamo le delno. Začetki prvi faze (če predvidevamo, da so upoštevali predvsem poslovne uporabnike) segajo vsaj v leto 1997, če ne celo v leto 1996. Velika pričakovanja, kot jih opisuje Gartner, so povzeta predvsem po predvidevanju nerealnih ocen o rasti in hitrosti razvoja Linuxa, v vsakem primeru pa padec (razen morda vrednosti delnic določenih podjetij, ki se ukvarjajo z odprto in prosto programsko opremo) ni tako opazen. Predvsem pa predvidevam, da bo rast veliko bolj strma in konstantna. Prva napaka, ki jo v večini primerov povzemajo vse večje medijske hiše in hiše, kot je Gartner Group je, da je Linux najbolj primeren ali bolje rečeno, primeren samo za spletne strežnike. Kvaliteta ostalih omrežnih servisov je enaka, če ne še večja, le medijsko je manj opevana in definirana. Ena izmed prelomnih točk, ki bo pospešila razvoj in uveljavljanje Linuxa, je tudi razvoj zanesljivega uporabniku prijaznega grafičnega okolja. Če previdevamo, da je kvaliteta strežniških servisov potrebovala cca. 4 leta, da je dosegla tehnološko priznanje tudi poslovnih uporabnikov in enako formulo predvidimo tudi za grafična okolja lahko v letu 2003-2004 pričakujemo ponoven velik skok uporabe Linux sistemov predvsem zaradi razlogov iskanja novih tržnih niš razvijalcev komercialnih programov (zniževanje stroškov lastne rešitve z nižanje stroškov zahtevane platforme). Poleg tega prav področje podatkovnih strežniških storitev, ki je tudi področje, ki je za

84 Gartner - The future of Linux and Open Source 20.Junij.2001.

poslovne uporabnike izredno zanimivo, veliko obeta. Podatkovni bazi PostgreSQL⁸⁵ in MySQL⁸⁶ že danes ponujata vrhunsko kvaliteto in zmogljivost. Z vstopom podjetij kot je Red Hat tudi na trg trženja odprtih in prostih podatkovnih baz, predvsem pa z razvojem podpornih orodij, ne verjamem, da se bodo uresničila predvidevanja Gartner Group.

9. Aktualna ekonomska vprašanja

"Padanje vrednosti delnic podjetij, ki se ukvarjajo z odprto in prosto programsko opremo (predvsem Linuxom), kaže na neuspeh odprte in proste programske opreme (Linuxa)."

Kakšna je povezava med vrednostjo delnic podjetij, ki se ukvarjajo z Linuxom, in možnostmi za uspeh Linuxa na splošno? Ko so delnice podjetij kot sta Red Hat in VA Linux dosegale astronomske vrednosti, je bil to indikator za prihod in uspeh Linuxa. Seveda zagovorniki odprtih sistemov temu niso ugovarjali in so to celo izrabljali za dokazovanje resnosti Linuxa. Sedaj, ko delnice teh podjetij prodajajo močno pod prvotno ceno to dokazuje, da sijaj Linuxa počasi ugaša. Poglejmo le število Linux podjetij, ki so se zaprla ali pa odpuščajo svoje delavce. Ali to kaže na konec Linuxa in velikih idej in moramo sedaj nazaj k komercialni programski opremi in podjetjem?

Prvo, Linux je popolnoma neodvisen od Linux podjetij in to na način, katerega trg do sedaj ni poznal. Windows pomeni Microsoft, Netware pomeni Novell, OS/390 pomeni IBM. Če zapremo podjetje Novell, to gotovo pomeni tudi konec Netware operacijskega sistema, razen v primeru, ko določeno podjetje kupi produkt in ga vzdržuje pri življenju še naprej (seveda pa se po drugi strani lahko Microsoft enostavno odloči in kupi Novell ter ga uniči). Take in podobne stvari se Linuxu ne morejo zgoditi. Odprti sistemi, kot je Linux so neodvisni od komercialnih neuspehov in podjetij, ki jim ne uspe zgraditi uspešnega poslovnega modela z odprtimi in prostimi sistemi. Dejstvo je, da se podjetja s svojimi izjavami, da podpirajo Linux močno motijo, saj je ravno obratno, Linux podpira njih.

"Odprta in prosta programska oprema ne more biti ekonomsko uspešna."

Odprti in prosti sistemi lahko obstajajo še naprej. Če podjetja z njimi ne morejo služiti denarja (kot naj bi to kazala nizka vrednost delnic Linux podjetij), potem je to samo še en velik tehnični uspeh, ki je postal komercialni neuspeh. Zgodovina je pozna mnogo podobnih primerov. Linux in z njim odprti in prosti sistemi ne morejo uspeti, če ljudje z njimi ne morejo služiti in to je argument, s katerim imajo probleme celo zagovorniki odprtih in prostih sistemov - zaključka, da morajo odprti in prosti sistemi producirati denar, če želijo uspeti.

⁸⁵ PostgreSQL [URL: <http://www.postgresql.org>], 18.09.2001.

⁸⁶ MySQL [URL: <http://www.mysql.org>], 18.09.2001.

Tovrsten argument ni popoln, saj se usmerja le na ponudbeno stran, zanemarija pa povpraševanje. Na strani povpraševanja uporabniki prihranijo ogromne količine denarja z uporabo odprtih in prostih sistemov. Ker prihranjen tolar, pomeni tudi zaslužen tolar, dobimo močno ekonomsko podlago za uspeh odprtih in prostih sistemov. Ko nekdo zaradi nečesa privarčuje, smo lahko prepričani, da se bo močno potrudil, da bo tovrstne prihranke ohranil tudi v prihodnosti.

"Neplačevanje programske opreme bo uničilo programsko industrijo"

Obstajajo strahovi, da se bodo tovrstna predvidevanja res uresničila. Ti strahovi so utemeljeni s krakoročnimi prihranki uporabnikov, ki pa dejansko na račun prihrankov uničujejo dolgoročni uspeh, saj potrebujejo finančno močne ponudnike za kvalitetno podporo in konstanten razvoj.

V realnosti so tovrstne trditve zelo dvomljive, predvsem iz besed oseb, ki same sebe opisujejo kot tržne kapitaliste. Ko kupec izvede nakup, ali razmišlja o svojih prihrankih ali o uspehu ponudnika in ekonomije? Ali je razumno od njih pričakovati, da bodo kupovali dražje produkte, ker naj bi enkrat v prihodnosti od tega imeli nekakšne koristi? Ko je ogrožena generacija dobaviteljev (konjske vprege -> vlaki -> letala -> ...) poskušajo prepričati ostale, da je ogrožena tudi skupnost kot celota. Če vzamemo zgodovino, kot določen vodnik, se bodo uporabniki sami odločili, kakšni so njihovi interesi. Ponudniki se bodo morali tem interesom pač prilagoditi. Odločitev uporabnikov jih bo morda uničila, ampak civilizacija bo gotovo preživela kot vedno do sedaj.

"Zakaj bi programerji prispevali svoje znanje in čas, če iz tega ni denarja?"

Pri izbiri med prostim in komercialnim produktom s ceno je razumljivo, da se bo uporabnik odločil za prosti produkt. Vendar, zakaj bi kdorkoli ta produkt napisal? Odgovor smo sicer dokaj podrobno razložili v enem izmed prejšnjih poglavij, zahteva pa še nekaj kratkih povzetkov in dodatkov.

Predvidevanje, da obstajajo samo tri vrste transakcij med različnimi stranmi:: uspeh-uspeh, uspeh-neuspeh, poraz-poraz (neuspeh-neuspeh transakcija se pod pogojem racionalnega odločanja skoraj ne more zgoditi). Uspeh-neuspeh se zgodi, ko je stran, ki zmaga, močnejša od druge in lahko transakcijo izsili. Vse ostale transakcije nastopijo s privoljenjem obeh strani in pomenijo uspeh-uspeh. V primeru odprtih sistemov so prejemniki programske opreme gotovo zmagovalci, na drugi strani pa izgleda, kot da tisti, ki programsko opremo pišejo, ne dobijo ničesar, saj jim uporabniki nič ne plačujejo. Naš zaključek bi bil lahko, da potem to ni uspeh-uspeh transakcija in v tem ni ekonomskega motiva za programerje.

Predvidimo najprej, da je edina motivacija pisanja programske opreme ekonomska, kar gotovo ne drži, saj obstajajo tudi druge vrste transakcij, ki jih nismo upoštevali: uspeh - nevtralnno, neuspeh - nevtralnno, nevtralnno - nevtralnno. Pod predpostavko racionalnih odločitev, neuspeh - nevtralnno in nevtralnno - nevtralnno možnosti skoraj nista mogoči, medtem ko se zmaga - nevtralnno transakcija lahko in to zelo pogosto.

Večina programerjev odprte in proste programske opreme verjetno ne bi razvijala programskih paketov, če bi jih to kaj stalo. Seveda vložene napore in čas pri pisanju ne dojemajo kot stroške. Večina jih piše programske pakete, zato da rešijo specifičen problem, s katerim se srečujejo. Proces razvoja je v tem primeru zelo pozitivno vpliva na programerja. Ko je program enkrat napisan, ga lahko uporabljajo tudi drugi, saj s tem ne omejuje svobode avtorja pri njegovi uporabi, niti mu zaradi tega ni traba česarkoli plačevati. Imamo torej transakcijo uspeh - nevtralnno in s tem nobenega ekonomskega razloga, ki bi jo preprečeval.

Pravi ekonomisti bi sicer omenili, da stroški pri dajanju programske opreme brezplačno, dejansko obstajajo - oportunitetni stroški neprodajanja. Seveda pa za večino razvijalcev problematika prodajanja prinaša več stroškov kot koristi, tako da so realni oportunitetni stroški dejansko nič.

Če to ni zadosten razlog, programerji odprih in prostih sistemov težijo k delu s skupnostjo, ki z njimi deli znanje in novo kodo. Metaforično bi lahko rekli, da vsak programer prispeva eno opeko, v zameno pa dobi celotno hišo. Pri programski opremi pridobitev ene osebe ne izhaja iz stroškov druge, saj se kopija popolnoma v ničemer ne razlikuje od originala. Za zaključek lahko zato potrdimo, da denarne nagrade ne vplivajo na obseg in nagibe pri pisanju odprih in prostih programov.

"Razvoj odprte in proste programske opreme zahteva določen vložek znanja in napora, tako da mora za ta napor obstajati plačilo."

Programerji odprtega in prostega programja ne izgubijo ničesar, če dajejo na razpolago programsko opremo, ki so jo že napisali. Seveda pa ne morejo živeti le od programske opreme in zadovoljstva. Za doseganje dolgoročne uspešnosti potrebujejo mehanizem, ki bo omogočal tudi finančno podporo razvijalcev. Obstaja kar nekaj poslovnih modelov, ki poizkušajo doseči finančno kompenzacijo za razvijalce. SourceForge⁸⁷ in Collab.Net⁸⁸ sicer vključujeta možnost prispevkov uporabnikov za plačilo razvijalcem, vendar uspeh tovrstnih mehanizmov še ni dokazan. Razvijalci bi sicer lahko služili s podporo svojih produktov, prodajo kopij ali svetovanjem in dodatnimi storitvami, vendar dokazano učinkovit tovrstni

87 SourceForge [URL: <http://www.sourceforge.net>], 21.09.2001.

88 Collab.net [URL: <http://www.collab.net>], 21.09.2001.

poslovni model še ne obstaja. Tudi ob upoštevanju najslabšega scenarija, ne pomeni, da se bo razvoj odprte in proste programske opreme končal. Dokler imajo programerji odprtega in prostega programja alternativne vire zaslužka (dnevne službe), ne izgubljajo ničesar s svojim delom za odprte in proste projekte v svojem prostem času. Z vedno večjo uporabo odprtih produktov se povečuje tudi število potencialnih razvijalcev in s tem garancija razvoja v prihodnosti.

"Ali programerji odprte in proste programske opreme uničujejo lastne službe?"

Koliko časa lahko programerji delajo za komercialna programska podjetja in v prostem času pišejo programske opreme, ki lahko potencialno ta podjetja tudi uniči? To vprašanje lahko vzamemo kot končni argument proti uspešnosti ekonomije odprte in proste programske opreme. Proti tovrstnim argumentom lahko uporabimo statistične podatke, ki jih navaja Eric Raymond⁸⁹: le 5% programerjev dejansko piše komercialno programske opreme. Ostalih 95% piše in vzdržuje programske opreme, narejene po meri za lastno uporabo podjetij. Odprta in prosta programska oprema tekmuje le s programske opreme, ki se prodaja kot produkt. V najslabšem primeru bi tako programerji odprte in proste programske opreme povzročili izgubo dela 5% programerjev, to pa je glede na skupne koristi sprejemljiva stopnja skupne škode.

"Kdo bo investiral v razvoj programske opreme, če ta ne bo prinašala donosa?"

Kako je lahko strošek programske opreme nič? Ali razvoj ne zahteva znatnega vložka napora? Celo odprta in prosta programska oprema se ne razvijata sama od sebe. Programerji pri razvoju porabijo več mesecev svojega časa, truda in znanja kako je lahko potem strošek programske opreme nič?

Prodajanje pod ceno imenujemo "dumping". Je odprta in prosta programska oprema potem takem kriva dumpinga? Če tovrstnega dejanja ne preprečimo, lahko to uniči klasično lastniško programske industrije. Kdo bo investiral v razvoj programske opreme, če ta ne bo prinašala donosa?

Odgovor je lahko presenetljiv, saj zanika mnoge izmed naših osnovnih zaključkov o načinu delovanja sveta.

Če predvidimo, da bi Linux razvilo eno izmed komercialnih podjetij ta ne bi bil nikoli brezplačen. Lastniške organizacije, financirane z dolgom ali kapitalom morajo prikazati

⁸⁹ Eric Raymons [URL: <http://www.tuxedo.org/~esr/writings/magic-cauldron/magic-cauldron-3.html>], 03.09.2001.

določen donos na svoje investicije. Čeprav so mejni stroški ustvarjanja novih kopij programske opreme nič, je potrebno začetni vložek v razvoj razporediti med te kopije. Zaradi tega je potrebno programski paket oblikovati kot produkt in prepovedati vsakršno reprodukcijo. Enostavnost reprodukcije je potrebno omejiti s pravnim sistemom. Le tako lahko tovrsten model deluje. Lastniški model programske opreme deluje na osnovi postavljanja umetnih omejitev dosegljivosti. Fizično in ekonomsko je sicer mogoče ustvariti neomejeno število kopij, vendar je pravno kaznivo. Z vidika družbe je to strošek za izbiro komercialnega modela razvoja programske opreme.

Poglejmo si še alternativo investicijskega modela. Če vložek v razvoj programske opreme razumemo kot stroške in ga enostavno odpišemo, programski opremi ni potrebno izkazovati določenega donosa na investicijo. Brez obstoja umetnih omejitev na reprodukcijo ima lahko svet poljubno število kopij in nobene potrebe po pravnih omejitvah. Z vidika družbe je to gotovo bolj učinkovito. Seveda pa velikih stroškov ne moremo enostavno odpisati. Potrebno jih je amortizirati. Tu se za nepogrešljivo kaže še ena lastnost programske opreme. Programsko opremo lahko skupaj razvija tisoče programerjev. Med vsemi izdelki človeštva tako fizičnimi in intelektualnimi le programska oprema vključuje lastnosti neskončne replikacije in amortizacija vložene truda. V tem pogledu je odprta in prosta programska oprema bolj naraven in učinkovit način razvoja programske opreme. S pridobitvijo večjega števila razvijalcev za delo na določenem projektu vsakdo izmed njih dela manj kot nekaj ur dnevno in ta vložek enostavno odpišejo brez pričakovanja finančne nagrade. Zato so odprti in prosti operacijski sistemi in spremljajoče aplikacije prosto dosopne vsakomur.

"Odprta in prosta programska oprema je sicer našla svojo tržno nišo, vendar pa bodo zaprti, lastniški produkti še vedno zavzemali večinski tržni delež".

Morda smo do sedaj dokazali, da bo odprta in prosta programska oprema tudi dolgoročno preživela in se razvijala. Ali morda močno pretiravamo, če predvidimo konec zaprte, lastniške programske opreme? Ali ni večina zaprtih, lastniških programskih produktov veliko pred svojimi tekmeci odprte in proste programske opreme? Kako lahko odprta in prosta programska oprema premaga lastniško programsko opremo v naboru funkcionalnosti? Če vzamemo za primer odprto in prosto podatkovno bazo PostgreSQL⁹⁰ in jo poskusimo primerjati z Oraclom⁹¹. Uporabniki ne bodo uporabljali manj zmogljivih proizvodov le zato, ker so brezplačni. Pripravljeni so plačati za boljše produkte. Primer je podoben anekdoti dveh mož, ki sta v džungli srečala tigra. Eden izmed njiju si začne obuvati svoje tekaške superge, medtem ko mu drugi zašepeta: "Si nor! Saj nisi hitrejši od tigra!". Prvi odgovori: "Saj mi ni treba. Hitrejši moram biti le od tebe!" Odprti in prosti programski opremi ni treba preseči svojih

90 PostgreSQL [URL: <http://www.postgresql.org>], 21.09.2001.

91 Oracle [URL: <http://www.oracle.com>], 21.09.2001.

lastniških konkurentov, ampak mora biti le dovolj dobra, da bo ustrezala uporabniškim zahtevam. Zakaj bi uporabniki plačevali za funkcionalnost, ki jo ne potrebujejo?

Seveda je to le ekonomski argument in ne tehnološki, ki bi se skladal s trditvijo zagovornikov odprte in proste programske opreme v smislu "odprta in prosta programska oprema se razvija hitreje, hitrejši so tudi popravki napak, tako da bo zelo kmalu dohitela in prehitela svoje komercialne tekmece." V prihodnosti bodo morda komercialna podjetja prisiljena prodajati 90% funkcionalnost, ki jih bo razlikovala od ostalih in ki je trg v resnici niti ne potrebuje, medtem, ko bo standardna funkcionalnost dosegljiva vsem brezplačno. Ta standardna funkcionalnost bo upoštevala standarde, medtem ko bo dodatna funkcionalnost ostala lastniška in zaprta v lastne standarde proizvajalca. To bi pomenilo, da je odprta in prosta programska koda postala glavni igralec na programskem trgu.

"Uporabniki ne bodo nikoli zaupali stvari, ki je brezplačna".

Teorija je sicer zelo prepričljiva, vendar se realnost kaže drugače. Odprta in prosta programska oprema se ne bi nikoli razvijala s tako hitrostjo, če bi to držalo. Uporaba odprte in proste programske opreme v poslovnem svetu je počasnejša. Je to morda zato, ker ljudje ne spoštujejo in ne zaupajo brezplačnim stvarim?

Imamo konkurenčne izdelke in substitute. Veliko težavnejši je premik na substitut kot na konkurenčni izdelek. Odprta in prosta programska oprema je substitut in ne konkurenčni izdelek celotnemu spektru zaprte lastniške programske opreme. Seveda pa je pri uporabnikih potreben dodaten preskok v mišljenju, ki zahteva določen čas, preden bodo popolnoma sprejeli odprto in prosto programsko opremo. Potrebujejo torej pozitivne izkušnje in čas. Trenutno tržno stanje odseva natančno to. Možni prihranki, varnost in nadzor nad sistemom so dovolj močni argumenti za preskok.

"Odprta in prosta programska oprema sicer dodaja vrednost, vendar je ne ustvarja".

Raziskave finančnih analitikov so pokazale, da večina podjetij, ki prikazuje večje dobičke iz leta v leto to dosega z zniževanjem stroškov, ne pa s povečevanjem prihodkov. Z zmanjševanjem nepotrebnih stroškov in povečevanjem produktivnosti podjetja dodajajo vrednost, ki je bila ujeta v neučinkovitih procesih, vendar nove dodane vrednosti ne ustvarjajo, ker ne inovirajo in ne producirajo novih idej, iz katerih izhaja pravo bogastvo in dodana vrednost. Obstaja torej meja, do katere bodo ta podjetja slej ko prej tudi prišla.

Ali ni odprta in prosta programska oprema nekaj podobnega? Pomaga nam sicer zniževati stroške, ampak ali nam pomaga ustvarjati nove stvari? Če je le cenejši nadomestek obstoječe programske opreme, bo njen dolgoročni vpliv marginalen in ne revolucionaren.

Obstaja torej rezlika med trgom in skupnostjo. Za udeleževanje na trgu potrebujete denar. S tem trg avtomatsko izključuje vse osebkke, ki denarja nimajo, kar potrjuje dokaz, da izmed 6 milijard ljudi polovica globalne trge le opazuje. Povezava na odprto in prosto programsko kodo je očitna.

10. Sklep

Odperta in prosta programska oprema je že začela spreminjati pravila na trgih informacijske tehnologije, kar kaže na to, da bodo tista podjetja, ki bodo prva pri uporabi odprtih in prostih tehnologij, imela določene konkurenčne prednosti. Z vidika družbe so koristi pri uporabi odprtih in prostih tehnologij še večje. Splošno slabo stanje podjetij informacijske tehnologije se kaže tudi na uspehu podjetij, ki svoje poslovne modele gradijo na osnovi odprte in proste programske opreme. Težko je določiti ali je njihov poslovni neuspeh rezultat splošnega gospodarskega stanja ali rezultat zasledovanja družbenih idealov, ki v ekonomskem pogledu ne morejo uspeli. Zavedati se je potrebno dejstva, da je veliko lažje prodajati licence, kot pa storitve. V vsakem primeru pa je bil vpliv odprte in proste programske opreme že do danes tako velik, da tudi zaprta, lastniška programska oprema in načini njene prodaje ne bodo nikoli več taki, kot so bili do sedaj. Z vedno večjo restriktivnostjo in omejevanjem, predvsem pa uvajanjem novih licenčnih modelov bo vedno več uporabnikov rešitev iskalo v uporabi odprtih in prostih sistemov. Če odprta in prosta programska oprema ne bo uspela v ekonomskem pogledu, smo lahko prepričani, da bo zmagala v družbenem.

11. Literatura

1. Browne, Christopher B.: Linux and Decentralized Development, 1999.
[URL: http://www.firstmonday.dk/issues/issue3_3/browne/index.html], 11.09.2001.
2. DiBona Chris, Sam Ockman, Mark Stone: Open Sources: Voices from the Open Source Revolution. O' Reilly, 1999. 280 str.
3. Duane Stoddard, Gunnison Carbone: Open Source Enterprise Solutions. Canada, John Wiley & Sons, Inc., 2001. 330 str.
4. Dempsey, Bert J., Debra Weiss: A Quantitative Profile of a Community of Open Source Linux Developers, 1999, Unpublished working paper, School of Information and Library Science, University of North Carolina at Chapel Hill.
[URL: <http://www.ibiblio.org/osrt/develop.html>], 11.09.2001.
5. Eric S. Raymond The Cathedral and the Bazaar.
[URL: <http://tuxedo.org/~esr/writings/cathedral-bazaar/>], 11.09.2001.
6. Eric S. Raymond The Magic Cauldron.
[URL: <http://tuxedo.org/~esr/writings/magic-cauldron/>], 11.09.2001.
7. Eric S. Raymond Halloween-I. open source software, a (new?) development methodology, 1998, comments.
[URL: <http://www.opensource.org/halloween1.html>], 11.09.2001.
8. Free Software / Open Source: Information Society Opportunities for Europe?, April 2000, različica 1.2, [URL: <http://eu.conecta.it/>], 11.09.2001.
9. IDA Study, Study into the use of Open Source Software in the Public Sector June 2001, OSS Fact Sheet, Use of Open Source in Europe, The Open Source Market Structure.
[URL: <http://europa.eu.int/ispo/ida>], 17.09.2001.
10. Josh Lerner, Jean Tirole: The Simple Economics of Open Source, Working Paper 7600
[URL: <http://www.nber.org/papers/w7600>], 11.09.2001.
11. Linux Magazine. Inside the Red Hat IPO, the story of the first open source public stock offering. Linux Magazine, 1999.
[URL: http://www.linux-mag.com/1999-11/redhatipo_01.html], 11.09.2001.
12. Open Source Initiative, Open Source Definition, 1999.
[URL: <http://www.opensource.org/osd.html>] 11.09.2001.
13. Richard Stallman: The bsd license.
[URL: <http://www.gnu.org/philosophy/bsd.html>] 11.09.2001.
14. Richard Stallman: Copyleft: Pragmatic idealism, 1998.
[URL: <http://www.gnu.org/philosophy/pragmatic.html>], 11.09.2001.
15. Richard Stallman: Why free software is better than open source , 1998.
[URL: <http://www.gnu.org/philosophy/free-software-for-freedom.html>], 11.09.2001.

16. Richard Stallman: Why software should not have owners, 1998.

[URL: <http://www.gnu.org/philosophy/why-free.html>], 11.09.2001.

17. Thomas Roessler, Kristian Köhnert: How free software development can be supported,

1999. [URL: http://www.koehnert.de/kris/artikel/oss_funding/], 10.09.2001.