

Elektronski fakultet, Univerzitet u Nišu
Katedra za računarstvo

Diplomski rad

Poslovni modeli u softverskoj industriji

Kandidat: Milena Nikolić, 11244

Komisija:

Datum prijave: _____

1. _____

Datum predaje: _____

2. _____

Datum odbrane: _____

3. _____

Jul 2008.

Sadržaj:

1. Uvod.....	3
2. Istorija i okviri poslovanja	5
2.1. Istorija softverske industrije.....	5
2.1.1. Programerske usluge.....	5
2.1.2. Enterprise Softverski Proizvodi.....	6
2.1.3. Paketski softver.....	8
2.1.4. Internet era.....	9
2.2. Istorija softvera otvorenog koda.....	10
2.2.1. Uspon svojinskog softvera i slobodnog softvera.....	10
2.2.2. Svestan otpor.....	11
2.2.3. Slučajan otpor.....	12
2.2.4. “Slobodan softver” nasuprot “softveru otvorenog koda”.....	13
2.3. Pravna regulativa u softverskoj industriji	15
1.3.1. Licence.....	15
1.3.2. Softverski patenti.....	17
2. Ekonomske paradigme razvoja softvera.....	21
2.1. Ekonomska funkcija softvera u poslovanju.....	21
2.2. Ekonomske paradigme razvoja softvera.....	22
2.2.1. Paradigma kupovine gotovog softverskog proizvoda.....	22
2.2.2. Razvoj unutar kompanije ili ugovoreni razvoj.....	24
2.2.3. Kolaboracija bez licenci otvorenog koda.....	24
2.2.4. Paradigma razvoja otvorenog koda.....	25
2.3. Zaključak o ekonomskim paradigmama razvoja softvera.....	27
3. Primeri poslovnih modela u softverskoj industriji.....	28
3.1. Komercijalni softver zatvorenog koda.....	32
3.1.1. Primer poslovanja po ovom modelu - Microsoft.....	32
3.2. Softverske usluge.....	36
3.2.1. Primer poslovanja po ovom modelu – Red Hat.....	36
3.3. Pospešivanje prodaje hardvera softverom.....	39
3.4. Dualna licenca.....	42
3.4.1. Primer poslovanja po ovom modelu – MySQL.....	43
3.5. Prodaja oglasnog prostora na web portalima.....	44
3.6. Održivi nekomercijalni softverski proizvod.....	48
4. Zaključak.....	50
5. Literatura.....	51

1. Uvod

Softverska industrija je mlada. Za nešto više od 60 godina koliko postoji, postala je jedna od najvećih industrija na svetu, kao i jedna od najdinamičnijih. Prihodi koji se globalno ostvaruju u ovoj industriji su veći od \$800 milijardi godišnje.

Poslovni model predstavlja okvir ostvarivanja vrednosti. Ovaj termin se koristi za široki opseg neformalnih i formalnih opisa koje koriste kompanije za prikaz različitih aspekata svog poslovanja, uključujući ciljeve, ponudu, strategije, infrastrukturu, organizacione strukture, prakse trgovanja i operativne procese i politike.

Informacione tehnologije su promenile način poslovanja svih industrija. Šta više, i dalje ga menjaju, iz dana u dan. Poslovni procesi postaju efikasniji i sadržajni, kao i komunikacija, deljenje informacija i zabava. Dinamična oblast IT-a se menja i razvija iz dana u dan, a svojim napretkom pokreće i sve ostale sektore ljudskog života i rada. Posledično, informacione tehnologije menjaju i same sebe, unapređuju i svoju industriju iz dana u dan. Novi poslovni modeli se bez sumnje primećuju pri svakoj značajnoj promeni u svetu računarstva - kao primere možemo navesti pojavu personalnih računara, interneta ili razvoja otvorenog koda.

Poslovni modeli u softverskoj industriji su još specifičniji zbog same prirode softvera, koji mnogi nazivaju „tek za nijansu različitim od čiste ideje“. Načini na koje se danas prihodi ostvaruju od softvera kreću se u širokom rasponu od modela baziranih na prodaji proizvoda, preko onih koji na tržište plasiraju softverske usluge, do originalnih kombinacija sa drugim industrijama kakve su advertajzing ili proizvodnja hardvera. Sve poslovne modele i tokove ostvarivanja prihoda u ovoj oblasti je teško izolovati i klasifikovati, kao što je teško odabrati kompanije koje predstavljaju najboljeg predstavnika konkretnog modela.

Da bi kompanija u ovako dinamičnom sistemu opstala, neophodno je odlično poznavanje trenutnog stanja nauke (tzv. state-of-the-art) u oblasti informacionih tehnologija. Ipak, ovo često nije dovoljno, već je potrebno poznavati istoriju i trenutno stanje same industrije. Mnoge kompanije vode računa o tome kada je trenutak da promene poslovni model, da krenu u osvajanje novog segmenta tržišta ili odustanu od prevaziđenog. U ovom radu je dat jedan pogled na istoriju i stanje industrije kakva je sad.

U radu će biti opisani trenutno aktuelni poslovni modeli u softverskoj industriji. Najpre će biti postavljen okvir za priču o poslovnim modelima opisom okruženja i faktora koji utiču na razvoj poslovanja u ovoj oblasti.

Ukratko će biti izložena istorija ove mlade industrije, kroz opis njenih ključnih segmenata od početaka u 1950-im godinama pa do danas. Zatim će biti opisana istorija pokreta za razvoj softvera otvorenog koda. Ovi pokreti su bitni jer su pod njihovim uticajem formirani mnogi novi poslovni modeli razvoja softvera, i bitno je poznavanje ove ideje da bi se shvatio razvoj industrije kakva je danas. Takođe će biti opisana pravna regulativa u ovoj oblasti, jer ona postavlja okvir funkcionisanja. Konkretno, biće objašnjene softverske licence kojima se štiti intelektualna svojina, kao i kontroverzno pitanje softverskih patenata oko koga još uvek nema saglasnosti u zajednici.

U drugom delu rada su izložene ekonomske paradigme razvoja softvera. Ovde se razvoj posmatra sa strane korisnika softvera, kako profesionalnog (odnosno korporativnog) tako i kućnog, i razmatraju se svi aspekti razvoja iz ove perspektive – troškovi i rizik razvoja, efikasnost finansiranja, mogućnost diferencijacije u odnosu na konkurentne proizvode i stepen u kom drugi mogu biti ograničeni u korišćenju softvera. Detaljno se opisuju četiri osnovne ekonomske paradigme razvoja softvera.

U trećem delu rada su nabrojani i opisani poslovni modeli u softverskoj industriji, sa strane proizvođača softvera. Svaki model je opisan na primeru neke poznate kompanije koja proizvodi softver da bi ostvarivala prihod. Vredi naglasiti da kao primer nisu korišćene samo kompanije kojima je osnovna delatnost razvoj softvera, već mnoge od njih pružaju usluge (između ostalog usluge konsaltinga, treninga, sertifikacije, podrške) ili koriste softver za unapređenje ili osnovu prodaje (računarske opreme, oglasnog prostora itd.). Danas je tehnologija u tolikoj meri povezana sa poslovanjem, da je sve teže izolovati poslovne modele u kojima je softver jasni izvor prihoda, a u kojima je trošak. Postoji veliki broj modela, a ovde će biti nabrojano desetak, i detaljno opisano nekoliko njih.

Za svaku kompaniju su predstavljeni objektivni podaci u vidu vrednosti prihoda, kao i tabela i grafikona sa rastom i udelom u tržištu. Pomenute su i kritike ili opasnosti po održivost poslovnog modela kompanije koje se najčešće pominju u javnosti.

2. Istorija i okviri poslovanja

U ovom delu će biti dati okviri i kontekst poslovanja u softverskoj industriji. Najpre je data kratka istorija softverske industrije, zatim razvoj pokreta otvorenog koda, i najzad trenutna pravna regulativa u ovoj oblasti

2.1. Istorija softverske industrije

Softverska industrija predstavlja onaj deo aktivnosti programiranja kojim se trguje između organizacija za proizvodnju softvera i korporativnih ili individualnih kupaca softverskih proizvoda i usluga. Trgovina softverom predstavlja samo deo softverske aktivnosti koji ne može biti tačno procenjen, zato što se veliki deo programiranja odvija unutar firmi i ta vrednost nije obuhvaćena analizama industrijskih analitičara. Po procenama, veličina tržišta trgovine softvera u svetu u 2007. godini je bila oko \$800 milijarde.

Tradicionalna Industrija trgovine softverom se može podeliti u 3 glavna sektora: programerske usluge, enterprise softverski proizvodi i paketski softverski proizvodi. Ova 3 sektora su se izdvojila sredinom 1950-ih, sredinom 1960-ih i krajem 1970-ih, respektivno, a u skladu sa tehnološkim mogućnostima i poslovnim okruženjem tog vremena. Najuspešnije firme su razvile specijalizovane sposobnosti koje su im omogućile da prosperiraju unutar svog sektora; ipak, ova specijalizacija je otežavala osvajanje drugih sektora, i veoma mali broj firmi je bio uspešan u više od jednom od njih. Svakako treba napomenuti da se softverska industrija ne sastoji od nezavisnih snabdevača softvera, već uključuje i proizvođače računara kao što su IBM, Unisys i NCR koji pružaju programerske usluge i softverske proizvode pored njihove ponude hardvera, i oni spadaju među najveće snabdevače softvera.

Tabela 1. Tržište softvera (prihodi u milionima \$), 1970–2005

Godina	1970	1975	1980	1985	1990	1995	2000	2005
Usluge programiranja	744	2,704	5,980	12,466	31,400	45,957	66,800	200,800
Softverski proizvodi	250	1,620	5,730	26,576	79,060	116,622	209,378	409,378
UKUPNO	994	4,324	11,710	39,042	110,460	162,579	266,178	610,178

SAD su kroz istoriju bile lider u softverskoj industriji, i danas čine polovinu svetskih prihoda i tri četvrtine tržišta softverskih proizvoda.

2.1.1. Programerske usluge

Prvi komercijalni elektronski računari – obično poznati kao mejnfrejm računari – su počeli da se prodaju u ranim 1950-im. Oni su bili veoma skupi, obično su se iznajmljivali za \$100,000 godišnje. Većina korporacija koja je imala računare je sama razvijala svoje programe, za šta je upošljavala oko 30 do 40 radnika. Ovo nije bio disproporcionalan trošak u odnosu na ukupnu cenu izvršavanja računara.

Do sredine 1950-ih je ipak cena mejnfrejm računara značajno pala, i korišćenje računara se brzo proširilo – pri čemu je broj mejnfrejm računara porastao sa 240 u 1955. na preko 1000 do 1960. Vlasnici ovih povoljnijih računara često nisu bili voljni da regrutuju stalno zaposlene programere, umesto toga preferirajući naručivanje programa od softverskih kontraktora. Da bi bila zadovoljena tražnja, otvarane su prve firme za programerske usluge. Prvu takvu firmu, Computer Usage Corporation (CUC) su otvorila u Njujorku 1955. godine 2 bivša programera IBM-a, i firma se inicijalno specijalizovala za razvoj tehničkih aplikacija za naftnu i inženjersku industriju. Glavne barijere za sklapanje podugovora za proizvodnju softvera su bile (i ostale) veoma male, i često se kaže da je sve što je potrebno „sveska i olovka za kodiranje“, odnosno računar [4]. Najvažnije sposobnosti su bile tehničko znanje direktora, koje je sticano kroz rad sa korisnicima ili proizvođačima računara. Nekoliko desetina firmi je ušlo na tržište programerskih usluga u drugoj

polovini 1950-ih. U većini slučajeva, firme su se specijalizovale za konkretne tehničke primene ili u vertikalnim tržištima kao što su finansijske usluge, prodaja ili proizvodnja.

Veoma drugačiji tip ulaska na tržište programerskih usluga je počeo da se javlja sredinom 1950-ih, specijalizovanjem za kreiranje veoma velikih programa koji su bili iznad tehničkih mogućnosti čak i najvećih i najsofisticiranijih korisnika. Prva firma ove vrste je bila Systems Development Corporation (SDC). SDC je 1956. unajmljen da razvije veliki SAGE sistem vazdušne odbrane. Ovaj sistem je bio nevidjen do tad, i sastojao se od više od milion računarskih instrukcija. SDC je zapošljavao nekoliko stotina programera, što je u to vreme predstavljalo skoro polovinu programera u svetu. Ipak, došlo je do velikih promena u strukturi zaposlenih, jer su iskusni programeri odlazili u privatni sektor zbog većih primanja. U to vreme je SDC bio smatran univerzitetom programera i tokom 60-ih su se bivši zaposleni ove kompanije mogli naći u svim bitnijim programerskim firmama u svetu.

SAGE je bio sistem za rad u realnom vremenu gde je računar bio u srcu informacionog sistema koji je simultano odgovarao na spoljne događaje. Kako je vlada SAD upošljavala više i više sistema za odbranu u realnom vremenu u kasnim 50-im i 60-im, integratori sistema kao što su TRW, MITRE, General Electric, Westing House, Hughes Dynamics i Lockheed su počeli da razvijaju ekspertizu u konstrukciji softvera. Ove tehnologije za rad u realnom vremenu su bile izuzetno skupe za razvijanje i inovaciju, ali kada su jednom bile napravljene, bilo je lako proširiti ih na civilni sektor u aplikacije kao što su avionske rezervacije i on-line bankarstvo. Kada je Evropa počela da dostiže SAD u 1960-im godinama, američke nezavisne softverske firme i operateri programerskih usluga su već bili u značajnoj prednosti.

Do kasnih 60-ih, najuspešniji softverski kontraktori su postali značajne firme. Npr, do 1967. CUC je imao 700 zaposlenih, kancelarije u 12 gradova u SAD i godišnju prodaju od \$13 miliona. Ove firme su sad nudile širok raspon računarskih usluga, koje su išle puno iznad pisanja programa. Druga firma, Computer Sciences Corporation (CSC) je osnovana 1959. od strane 5 programera u cilju pisanja softvera za proizvođače računara. Ova firma je postala jedna od najvećih firmi za pružanje računarskih usluga na svetu, i danas je to, sa prihodima of \$18.1 milijarde u 2007. i 98,000 zaposlenih u celom svetu.

Ipak, velike firme su izuzetak, i industriji programerskih usluga značajno nedostaje koncentracija. Do kasnih 60-ih bilo je nekoliko stotina firmi za programerske usluge u SAD-u, ali je manje od 50 njih imalo više od 100 zaposlenih. Danas postoji nekoliko hiljada organizacija za softverski kontraktoring, ali je njihov srednji broj zaposlenih manji od 20. Samo mali procenat od ovih postaju globalni igrači.

2.1.2. Enterprise Softverski Proizvodi

1960-ih je došlo do eksplozije u korišćenju računara u celom svetu. Računarska tehnologija je dramatično evoluirala – tranzistori su zamenili vakuumske cevi, a onda je i mikroelektronika zamenila diskretne tranzistore – sa konsekvantnim poboljšanjima u brzini, kapacitetu, pouzdanosti i ceni. U svetu je broj računara rastao sa 4,400 u 1960-oj na 63,000 do kraja dekade. Koncept paketskog softvera se javio kao tehnološko rešenje – skoro kao istorijska neophodnost – na problem snabdevanja softvera uz šireću populaciju računara. Ista je do sredine 60-ih počela da prevazilazi programersku radnu snagu potrebnu za pisanje softvera za svaku od ovih instalacija. Softverski paket je bio program napisan za konkretnu industriju, ili za čestu primenu kao što je procesiranje platnog prometa ili upravljanje inventarom. Na početku su softverski paketi bili isporučivani besplatno od strane proizvođača računara, kao deo skupa usluga neophodnih za korišćenje računara, a koji je takodje uključivao i obuku korisnika, inženjering na licu mesta i sl.

U srednjim i kasnim 1960-im, solidan broj firmi za programerske usluge je počeo da prodaje softverske pakete. Paketi su obično bili izvedeni iz softvera razvijenog za nekog od prethodnih

klijenata, i nazivani su proizvodima da bi se razlikovali od softverskih paketa proizvođača računara i da bi se istakao stepen korisničke podrške koji proizvođači računara obično nisu pružali uz svoj besplatan softver. Dva rana primera softverskih proizvoda su bili Autoflow i Mark IV. Autoflow je bio program koji je pomagao korisnicima računara sa softverskom dokumentacijom i na tržište ga je izbacila kompanija ADR 1965 godine. Mark IV je bio rani oblik baze podataka koji se pojavio 1967. Iako su ovo bila 2 najpoznatija proizvoda 60-ih, svaki od njih je dostigao prodaju od svega nekoliko stotina primeraka do kraja dekade.

Činjenica da su proizvođači distribuirali besplatne softverske pakete bila je glavno ograničenje rasta industrije softverskih proizvoda, zato što je bilo izuzetno teško takmičiti se protiv besplatnih paketa pružanjem dodatnih funkcionalnosti koje bi trebalo da opravdaju kupovnu cenu od \$5,000-\$50,000. Kao rezultat antimonopolskog pritiska, i privatne tužbe kompanije ADR, 1970-e je IBM razdvojio svoj softver i usluge od hardvera, time konvertujući mnoge od svojih softverskih paketa u softverske proizvode koje je trebalo kupiti. Ovo odvajanje je pokrenulo velike promene u industriji; ADR i Informatics su, na primer, utrostručavale svoju prodaju više godina za redom posle toga. Tokom 1970-ih, nekoliko stotina postojećih firmi za programerske usluge i mnogo novih je ušlo na tržište softverskih proizvoda. Neke od novih firmi koje su ušle bile su Computer Associates i Oracle, koje su kasnije postale glavni igrači u Enterprise softverskoj industriji. Ipak, 70-e nisu bile tako svetla dekada za softversku industriju zbog nedostatka kapitala koji je usledio nakon pada berze u kasnim 60-im, i posle kog je usledila kompjuterska recesija 1970-71. Zbog toga je rast u toku 70-ih bio skroman, i ukupna prodaja u industriji nije prekazila \$1 milijardu sve do 1978. (godina u kojoj su, samo poredjenja radi, prihodi IBM-a bili \$17 milijardi).

Do 1980-ih, tržište softverskih proizvoda je konačno sazrelo i raslo 30% godišnje – od agregatne prodaje vredne \$2.7 milijarde u 1980-oj do preko \$30 milijardi do kraja dekade. Kao i programerske usluge, industrija softverskih proizvoda je imala malu koncentraciju. Npr. istraživanje iz 1982. je pokazalo da je 50 ili 60 firmi činilo 50% ukupne prodaje, ostavljajući ostatak od otprilike 2,000 srednjih i malih firmi. Kao kontrast, u industriji mejnfrejm računara, manje od 20 firmi je držalo skoro kompletnu industriju, od čega jedna firma – IBM – više od 50%. Softverska industrija je ponekad karakterizovana kao “stenje, kamenje i pesak”, odnosno postoji nekoliko desetina globalnih igrača, nekoliko stotina firmi regionalnog nivoa i hiljade veoma malih firmi sa manje od 20 zaposlenih.

Vodeće firme su generalno rasle kroz konsolidaciju ili dominiranjem konkretnom oblašću softvera. Konsolidacija je bila posebno bitna u dobijanju udela u tržištu iz 2 razloga: pokazalo se izuzetno teško “imitirati” uspešan proizvod kome je trebalo više godina da evoluiru, i kao drugo, softverski proizvodi su se pokazali neosteljivim na cenu, jer kupci više vrednuju pouzdanost i bezbednost od cene. Među konsolidatorima je Computer Associates, osnovana 1976, bila najpoznatija i najuspešnija. Nakon svoje inicijalne javne ponude na berzi 1981. godine, ova kompanija je obavila veliki broj akvizicija koje su je učinile najvećim snabdevačem softvera do 1990. i održala je poziciju kao jedan od najveća tri od tad (obično se smenjujući na prvim mestima sa Microsoft-om i Oracle-om). Akvizicije ove kompanije su uključivale i neke druge konsolidatore, tako da kad bi se konstruisalo “porodično stablo”, firma bi sadržala nekoliko stotina imena (između ostalog uključujući i gore pomenute ADR i Informatics).

Do 1990-e, 2 glavna softverska proizvoda su činila polovinu prodaje korporativnog softvera: relacione baze podataka i ERP softver (Enterprise Resource Planning, odnosno softver za planiranje resursa preduzeća). Tehnologija relacionih baza podataka se pojavila u ranim 70-im u istraživačkoj laboratoriji IBM-a i Berkeley-ja. Ova tehnologija je bila veliki napredak, iako tehnički izazovna, u odnosu na ranije sisteme baza podataka. Jedna od prvih kompanija koje su iskoristile novu tehnologiju je bila Oracle, osnovan 1977. ERP softver se pojavio 80-ih kao rešenje u jednom proizvodu koje će zameniti agregaciju više aplikacija tipično korišćenih 70-ih i 80-ih. Vodeći snabdevač je bio SAP, nemačka kompanija koja je izumela ERP koncept u ranim 80-im. Iako danas postoji nekoliko konkurenata, nijedan nije prevazišao SAP-ovu početnu prednost.

Štaviše, SAP je jedina ne-američka kompanija koja proizvodi softver od najvećih 10 (i jedna od svega nekoliko u najvećih 100).

2.1.3. Paketski softver

Otkriće jeftinog mikroprocesora 1971. godine, odnosno računara na jednom mikročipu, je transformisalo celu industriju računara, kreirajući potrošački proizvod od nečeg što je ranije bilo skupo kapitalno dobro. Mikroprocesori su korišćeni i u konzolama za video igre i u personalnim računarima, a industrija tzv. paketskog softvera se razvila u 1970-im da zadovolji tražnju za programima za ove nove računarske proizvode. Proizvodi paketskog softvera su se razlikovali od enterprise softverskih proizvoda po niskim cenama, visokoj prodaji i različitim distribucionim kanalima. Potrošački softver je tipično prodavan u desetinama ili stotinama hiljada jedinica, po ceni od najviše nekoliko stotina dolara. Paketski softver je prodavan u maloprodajnim radnjama i poštanskim narudžbinama, sa malo ili bez postprodajne usluge, u poredjenju sa metodama direktne prodaje i održavanja kontakta sa klijentima kod snabdevača enterprise softvera.

Jedna od prvih firmi koja je uhvatila talas personalnih računara je bio Majkrosoft, koji su osnovali u Albukerkiju 1975. godine Bil Gejts i Pol Alen. Majkrosoft se specijalizirao kao proizvođač osnovnog operativnog softvera za ono što je tada nazivano mikro-računarima, i što je uglavnom prodavano hobistima. Personalni računari za masovno tržište, kao što su Apple II i Tandy TRS-80, su se pojavili 1977-78. U roku od nekoliko godina 2 softverske aplikacije, procesor reči i procesor tabela (spreadsheets), su učinile personalni računar generalno korisnim. Koncept procesora reči je postajao u svetu korporativnih računara, i mnogima je pala na pamet ideja kreiranja ovog proizvoda za tržište personalnih računara. Ipak, jedan proizvod, WordStar, je obezbedio dominantan udeo u tržištu. Kao kontrast, prva aplikacija za tabele na personalnim računarima, VisiCalc, nije imala jasnog prethodnika na mejnfrejmskim računarima, i odmah je zauzela celo tržište za sebe ali je u roku od nekoliko godina imala na desetine klonova.

WordStar i VisiCalc su bila 2 najprodavanija programa, svaki sa više od pola miliona prodatih kopija do 1983. WordStar, VisiCalc i drugi konkurenti su bili poznati kao "aplikacije produktivnosti" zato što su postali osnovni alati korporativnih informacionih radnika. Iako su aplikacije produktivnosti činile polovinu prodaje softvera, bilo je mnogo drugih žanrova softvera, svaka sa svojim vodećim programima – desktop izdavaštvo (npr. Adobe Systems-ov PageMaker), računarski dizajn (npr. Autodesk-ov Auto-CAD), lične finansije i mnogi drugi.

Tendencija da jedan proizvod dominira softverskim žanrom, uprkos mnogim imitatorima, je značajna karakteristika softverske industrije personalnih računara, i uglavnom se objašnjava takozvanim "efektom mreže". Pošto zajednica ili mreža korisnika vrednuje mogućnost deljenja i razmene fajlova i dokumenata, novi korisnici teže usvajanju softvera sa najvećom zajednicom korisnika – što favorizuje najpopularnije proizvode na dalju štetu njihovih konkurenata. Uredjenje u tržišnom udelu softverskih proizvoda određene kategorije se menja samo kada postoji veliki diskontinuitet u okruženju personalnih računara, pa se postepeno prevazilazi efekat mreže. Ovo se prvi put desilo nakon uvođenja IBM-ovog PC-ja u avgustu 1981. IBM-ov PC je bio moćniji od prethodnih desktop računara, i brend IBM-a je preporučio korišćenje u korporacijama, pa je ovaj računar uskoro činio 80% prodaje personalnih računara. Dve nove aplikacije produktivnosti su zasenile VisiCalc i WordStar i dominirale platformom narednu dekadu: 1-2-3 spreadsheet od Lotus Development korporacije i WordPerfect procesor reči. Majkrosoft koji se do tad preselio u rodni grad Gejtsa i Alena Sijetl, je razvio operativni system za IBM računare, ostvarujući tako enormne prihode koji su dalje podgrevali njegov rast.

Seizmički obrt na polju personalnih računara je nastao sa uvođenjem Microsoft Windows 3.0 operativnog sistema 1990 godine. Naravno Lotus i WordPerfect nisu bili kompatibilni sa Windows-om u to vreme, pa je Majkrosoft bio u poziciji da progura svoje pakete, Excel i Word, sa veoma malo konkurencije. U isto vreme, kompanija je počela da pakuje svoje aplikacije produktivnosti u jedan proizvod koji je nazvan Microsoft Office. U roku od nekoliko godina,

Majkrosoft je počeo da dominira tržištem, kako operativnih sistema za personalne računare, tako i aplikacija produktivnosti. Ovo je izazvalo restrukturiranje industrije jer su se konkurenti reorganizirali i spojili da bi ponudili odgovarajuće kancelarijske pakete. Do sredine 1990-ih, ovakvi paketi i individualne aplikacije produktivnosti su se prodavale u iznosima od 50 miliona primeraka godišnje, od čega su možda tri četvrtine bile Majkrosoftovi proizvodi.

Majkrosoftova sveprisutnost i monopolističko ponašanje su izazvali pažnju antimonopolske komisije Ministarstva Pravde SAD 1995. godine. Ovaj ekskluzivan fokus na Majkrosoft je bio iznenađujući jer je bilo sličnih monopola u sektoru enterprise softvera. Štaviše, iako Majkrosoft dominira većinom softverskih žanrova u kojima se nadmeće na tržištu, firma nema više od 10% ukupne prodaje u softverskoj industriji.

2.1.4. Internet era

Širenje PC-ja u 1980-im godinama je dramatično promenilo poslovne živote zaposlenih u kancelarijama: menadžeri su sami kucali svoje izveštaje, analitičari su se veliki deo dana bavili sa spreadsheet-ovima. Kućna mašina je poslata u prošlost.

Od sredine 90-ih Internet je imao podjednako dramatičan efekat i na živote kancelarijskih zaposlenih i kućnih korisnika računara. Iako je Internet bio u širokoj upotrebi u tehničkim zajednicama od ranih 80-ih, tek je pojava World Wide Web-a u ranim 90-im učinila internet dostupnim običnim korisnicima. Osnovnu softversku tehnologiju World Wide Web-a činili su (i čine) web čitač (browser) i web server. Prvi i najvažniji snabdaveč softvera za web je bila Netscape Communications Corporation osnovana 1994. godine. Osnivač je bio 24-godišnji Džim Klark, a kompanija je u roku od 2 godine postala milijarderska korporacija. Ipak, 1996. je uveden konkurentski proizvod, Internet Explorer, koji je brzo dostigao dominantni udeo u tržištu. Majkrosoftovo ponašanje u ovoj tržišnoj utakmici je dodalo težinu optužbama na njihovom sudjenju za monopolsko ponašanje.

U početku, web je uglavnom bio pasivno iskustvo, ali u kasnim 90-im sve više je postajao interaktivan i participativan, uz pružanje finansijskih usluga, putnih informacija, zabave, aukcija i prodajnih servisa, kao i informacija o proizvodima svih vrsta. U roku od nekoliko godina pojavili su se novi brendovi – Motley Fool, Amertrade, Yahoo, Travelocity, Amazon i mnogi drugi. Neka od ovih su nova preduzeća, dok su druga nove inicijative starih konsolidiranih firmi. Baš kao što web menja stari poredak informacionih usluga i maloprodaje, još više se menja svet softvera. Npr, metafora paketskog softvera se sve manje koristi, jer se proizvodi sve više dobijaju download-om sa interneta. Neki analitičari industrije predviđaju da će koncept softverskog proizvoda postati prevaziđen, i svi programi će biti dostupni na internetu, po zahtevu i plaćeno na osnovu korišćenja. Bez obzira da li se ovo ostvari ili ne, fraza softverska industrija će biti korišćena kao kolektivni izraz za kompanije koje se bave snabdevanjem programerskim dobrima i uslugama, bez obzira na tehnologiju koja je aktuelna u tom trenutku.

2.2. Istorija softvera otvorenog koda

Deljenje softvera je postajalo od kad postoji i sam softver. U ranim danima računara, proizvođači su mislili da je glavna kompetitivna prednost uvek u hardverskim inovacijama, pa nisu obraćali mnogo pažnje na softver kao poslovnu vrednost. Veliki broj klijenata ovih mašina su bili naučnici i tehničari, koji su bili sposobni da modifikuju i prošire softver koji je dolazio uz mašine. Klijenti su nekad distribuivali svoje zakrpe nazad i proizvođaču, ali i ostalim vlasnicima sličnih mašina. Proizvođači su često tolerisali ili čak ohrabrivali ovo: u njihovim očima su poboljšanja softvera, iz bilo kog izvora, samo činila mašinu privlačnijom drugim potencijalnim kupcima.

Iako je ovaj rani period ličio na današnju kulturu slobodnog softvera na mnogo načina, razlikovao se u 2 bitna aspekta. Prvo, bilo je vrlo malo standardizacije hardvera – to je bilo vreme kada je cvetala inovacija u dizajnu računara, ali je ta raznolikost računarskih arhitektura značila da je sve nekompatibilno sa svim ostalim. Tako, softver pisan za jednu mašinu generalno nije radio na drugoj. Programeri su težili da stiču ekspertizu na konkretnoj arhitekturi ili familiji arhitektura (kao što danas teže sticanju ekspertize u jednom programskom jeziku ili familiji jezika), pouzdani da će to biti transferabilno na bilo koji računarski hardver sa kojim dodju u kontakt. Kako je ekspertiza stručnjaka uglavnom bila veoma specifična za jednu vrstu računara, akumulacija znanja je imala efekat da su ti računari privlačniji njima i njihovim kolegama. Zato je i bilo u interesu proizvođača da se mašinski kod i znanje o arhitekturi šire što je više moguće.

Drugo, nije bilo interneta. Iako je bilo manje legalnih ograničenja za deljenje koda nego danas, bilo je više tehničkih: načini premeštanja podataka sa jednog mesta na drugo su bili relativno nezgodni. Bilo je nekih malih, lokalnih mreža, koje su bile dobre za deljenje informacija među zaposlenima iste istraživačke laboratorije ili kompanije, ali su ostajale barijere koje je trebalo prevazići ako ste želeli da delite kod sa svima, bez obzira gde su oni. Ove barijere su doduše prevaziđene na razne načine. Nekad su različite grupe komunicirale među sobom, slanjem diskova ili traka poštom, a nekad su i proizvođači bili centralno mesto za distribuciju zakrpa. Takođe je pomoglo što su mnogi od ranih programera radili na univerzitetima, gde je širenje znanja bilo očekivano. Ali fizička realnost transmisije podataka je značila da uvek postoji kašnjenje pri deljenju, koje je proporcionalno udaljenosti (pravom ili organizacionom) koju je softver trebalo da predje. Sveobuhvatno neometano deljenje, kakvo danas postoji, nije bilo moguće.

2.2.1. Uspon svojinskog softvera i slobodnog softvera

Kako je industrija sazrevala, nekoliko povezanih promena se desilo u isto vreme. Mnoštvo različitih hardverskih dizajna je postepeno ustupilo mesto nekolicini jasnih pobednika, koji su to bili kroz superiornu tehnologiju, superiorni marketing ili neku kombinaciju ova dva. U isto vreme, ali ne u potpunosti koincidentno, razvoj tzv. programskih jezika visokog nivoa je značio da je moguće pisati program jednom, u jednom jeziku, i automatski ga prevoditi ("kompajlirati") da se izvršava na različitim vrstama računara. Implikacije ovog su uticale na proizvođače hardvera – klijent je sada mogao da se upusti u veliki softverski projekat bez naročitog oslanjanja na konkretnu računarsku arhitekturu. Kada je ovo bilo kombinovano sa postepenim smanjivanjem razlika u performansama između različitih računara (jer su manje efikasni dizajni nestajali), proizvođač koji je tretirao hardver kao svoju jedinu vrednost je mogao da vidi smanjenje margina profita u budućnosti. Sirova računarska snaga je postajala šupljikava roba, dok je softver sve više postajao diferencijator. Prodaja softvera, ili barem tretiranje istog kao integralni deo prodaje hardvera, počelo je da liči na dobru strategiju.

Ovo je značilo da su proizvođači morali da počnu sa strožom primenom zaštite autorskih prava. Da su korisnici jednostavno nastavili da dele i modifikuju kod slobodno među sobom, mogli bi da nezavisno reimplementiraju neka od poboljšanja i onda ih prodaju kao vrednost dodatu na osnovnu verziju. Još gore, deljeni kod bi mogao da dodje u ruke konkurencije. Ironija je bila u tome što se sve ovo dešavalo baš u vreme kada je Internet uzimao maha. Baš kada je ničim

ometano deljenje konačno postajalo moguće, promene u računarskoj industriji su ga učinile ekonomski nepoželjnim, barem sa tačke gledišta jedne kompanije. Snabdevači su počeli da odbijaju korisnicima pristup kodu koji su ovi izvršavali na svojim mašinama, ili insistirali na ugovorima sa klauzulom ne-obelodanjivanja (non-disclosure), koja je deljenje koda činila nemogućim.

2.2.2. Svestan otpor

Kako je svet neometane razmene i menjanja koda polako nestajao, protivreakcija se kristalisala u umu bar jednog programera. Ričard Stolman je radio u laboratoriji za veštačku inteligenciju na MIT-u u 1970-im i ranim 80-im, što je bilo zlatno doba i zlatno mesto za deljenje koda. AI laboratorija je imala jaku "hakersku etiku", i ljudi nisu samo bili ohrabrivani već se od njih očekivalo da dele sva poboljšanja na sistemu. Ta edenska zajednica oko Stolmana se raspala ubrzo nakon 1980-e, kada su promene koje su se dešavale u ostatku industrije najzad došle i do AI laboratorije. Kompanije koje su pokretane u to vreme su unajmile mnoge od programera iz laboratorije da rade na operativnom sistemu koji je bio sličan onom na kom su radili u laboratoriji, ali sada pod ekskluzivnom licencom. U isto vreme, AI laboratorija je dobila novu opremu koja je došla sa svojinskim operativnim sistemom. Stolman je video šta se dešava i odlučio da se opire trendu. Umesto da ostane da radi u sada desetkovanoj laboratoriji, ili se zaposli u jednoj od novih kompanija gde rezultati njegovog rada ne bi bili dostupni svima, on je dao otkaz iz laboratorije i započeo GNU projekat i Free Software Foundation (FSF). Cilj ove organizacije je bio da razvije kompletno slobodan, besplatan i otvoren operativni sistem i telo aplikacionog softvera, u kom korisnici nikad neće biti zaustavljeni u menjanju i deljenju koda. Generalno, on je želeo da kreira ono što je uništeno u AI laboratoriji, ali na nivou cele zemlje i bez ranjivosti koje su dezintegrisale kulturu iste.

Pored rada na operativnom sistemu, Stolman je izumeo copyright licencu čiji su uslovi garantovali da će njegov kod biti zauvek slobodan. GNU General Public License govori da kod može biti kopiran i modifikovan bez ikakvih ograničenja, ali da kopije i izvedeni rad (modifikovane verzije) moraju biti distribuirane pod istom licencom kao original, bez dodatnih ograničenja. Generalno, on je iskoristio zaštitu autorskih prava da bi postigao efekat suprotan tradicionalnom copyright-u: umesto ograničenja distribucije softvera, on sprečava svakoga, čak i samog autora od ograničavanja. Za Stolmana, ovo je bilo bolje od jednostavnog ostavljanja koda da bude dostupan svima. Ako bi bio dostupan svima, bilo koja konkretna kopija koda ili njegov deo bi mogao biti inkorporiran u svojinski program (što se bez GPL-a i dešavalo). Dok takva inkorporacija ne bi umanjila dostupnost originalnog koda, ona bi značila da su od Stolmanovih napora mogli da imaju koristi njegovi neprijatelji – proizvođači svojinskog softvera. O GPL-u se može razmišljati kao o obliku protekcionizma slobodnog softvera, zato što sprečava da ne-slobodni softver iskoristi kod koji je pod ovom licencom. O softverskim licencama će biti više reči kasnije u ovom radu.

Uz pomoć mnogih programera, od kojih su neki delili Stolmanovu ideologiju, a neki su jednostavno želeli da vide puno dostupnog slobodnog koda, GNU projekat je počeo sa objavljivanjem slobodnih zamena za mnoge kritične komponente operativnog sistema. Zbog sada već opšte prisutne standardizacije u računarskom hardveru i softveru, bilo je moguće koristiti GNU zamene na drugim ne-slobodnim sistemima, što su mnogi i radili. Posebno su uspešni bili GNU editor teksta (Emacs) i C kompajler (GCC), zadobijajući veliki broj lojalnih korisnika, ne na ideološkoj osnovi, već na svojim tehnološkim kvalitetima. Do 1990, GNU je proizveo veći deo operativnog sistema, izuzev kernela (jezgra). Nažalost, GNU projekat je izabrao dizajn kernela za koji se ispostavilo da je teži za implementaciju nego što je očekivano. Kašnjenje usled toga je sprečilo da FSF izda prvu verziju potpuno slobodnog operativnog sistema. Finalni deo je sastavio Linus Torvalds, finski student računarstva koji je, uz pomoć volontera iz celog sveta, kompletirao dizajn slobodnog kernela koristeći konzervativniji dizajn. Nazvao ga je Linux (koristeći kao osnovu Minix, tako da je Linux i danas kompatibilan sa UNIX sistema), i kada je ukombinovano sa

postojećim GNU programima, rezultat je bio kompletno slobodan operativni sistem. Po prvi put je bilo moguće uključiti računar i obavljati posao bez ikakvog korišćenja svojinskog softvera.

Većina softvera na ovom novom operativnom sistemu nije bila proizvedena od strane GNU projekta, štaviše ovo nije bila jedina grupa koja je radila na stvaranju slobodnog softvera (npr, kod koji je kasnije postao NetBSD i FreeBSD je već bio u razvoju). Važnost FSF nije bila samo u kodu koji su pisali, već i u njihovoj političkoj retorici. Pričom o slobodnom softveru kao uzroku a ne pogodnosti, oni su učinili nemogućim da programeri nemaju stav o tome. Širenjem koda, tekstova i GPL-a širila se i poruka koju su želeli da pošalju.

2.2.3. Slučajan otpor

Ipak, na sceni slobodnog softvera se dešavalo još puno toga, i većina dešavanja nije bila eksplicitno ideološki usmerena kao Stolmanov GNU projekat. Jedno od bitnih dešavanja je bila aktivnost Berkeley Software Distribution grupe programera sa Berkli Univerziteta, koja se bavila postepenom re-implementacijom Unix operativnog sistema. Ovaj sistem je do kasnih 1970-ih bio svojinski istraživački projekat u AT&T kompaniji, ali je kasnije „oslobodjen“. BSD grupa nije deklarirala nikakve političke stavove o potrebi da se programeri povežu i dele kod međusobno, ali su praktikovali ovu ideju sa entuzijazmom, koordinisanjem masivnih distribuiranih napora u razvoju u kojima su volonteri napisali alate komandne linije, biblioteke koda, a najzad i sam kernel operativnog sistema. BSD je postao osnovni primer ne-ideološkog razvoja slobodnog softvera, a služio je kao mesto za obuku mnogih programera koji su želeli da ostanu aktivni u svetu open source-a.

Drugi krucijalni primer kooperativnog razvoja je bio X Window System, grafičko računarsko okruženje razvijeno na MIT-u sredinom 80-ih u partnerstvu sa proizvođačima hardvera, kojima je bio interes da budu u mogućnosti da svojim korisnicima ponude sistem sa prozorima. X licenca je namerno dozvoljavala svojinske ekstenzije na vrhu slobodnog jezgra softvera, jer je svaki član konzorcijuma želeo šansu da proširi osnovnu X distribuciju i time dobije kompetitivnu prednost nad drugim članovima. X Windows je sam po sebi bio slobodan softver, ali uglavnom kao način nivelisanja interesa između konkurentskih učesnika, a ne iz bilo kakve želje da se okonča dominacija svojinskog softvera. Još jedan primer, koji je prethodio GNU projektu nekoliko godina, je bio TeX, sistem pripreme teksta za izdavaštvo koji je napisao Donald Knut. On je TeX objavio pod licencom koja je svima dopuštala modifikaciju i distribuciju koda, ali da se rezultujući proizvod ne naziva „TeX“, osim ako prodje veoma strog skup testova (ovo je primer slobodne licence iz klase zaštite trademark-a, odnosno brenda). Knut nije zauzimao stranu u raspravi između slobodnog i svojinskog koda, njemu je samo trebalo bolji sistem za kucanje kako bi efikasnije napisao knjigu o programiranju, i nije video razlog da ne pusti svoj sistem u svet kada završi sa svojim poslom.

Bez daljeg navodjenja svih projekata i licenci, može se reći da je do kasnih 80-ih bilo puno slobodnog softvera dostupnog pod različitim licencama. Raznolikost licence je odražavala različite motivacije. Čak i neki od programera koji su birali GNU GPL licencu su bili mnogo manje ideološki vodjeni nego sam GNU projekat. Iako su radili na slobodnom softveru, većina programera nije smatrala svojinski softver socijalnim zlom. Neki su bili motivisani tehničkim izazovima, ili zadovoljstvom pri radu sa stručnim kolaboratorima, ili jednostavnom ljudskom željom da im se neko divi. Sve ove različite motivacije nisu stajale na putu stvaranju kvalitetnog softvera zato što softver u dobroj meri nije kreativna forma, već mora da prodje objektivne testove da bi se smatrao uspešnim – mora da izvršava ono što mu je namena i da bude u razumnoj meri oslobodjen od bagova. Ovo daje svim učesnicima u projektu zajedničku osnovu za razmišljanje i rad bez mnogo briga o motivacijama i netehničkim kvalifikacijama ostalih učesnika.

Programeri su imali još jedan razlog da se drže ove priče: ispalo je da svet slobodnog softvera proizvodi veoma kvalitetan kod. U nekim slučajevima je bio očigledno tehnički superioran najbližoj ne-slobodnoj varijanti, u drugim, bio je barem uporedljiv, a naravno uvek je koštao manje. Dok je

svoga nekoliko ljudi bili motivisano da koristi slobodan softver iz ideoloških ubedjenja, mnogo ljudi je želelo da koristi ovaj softver zato što je bio jako dobar. A od ljudi koji su ga koristili, uvek je postojao procenat ljudi koji su bili voljni da doniraju svoje vreme i veštine da bi pomogli održavanje i poboljšanje.

Tendencija da se proizvodi dobar kod nikako nije bila univerzalna, ali je bila sve češća u projektima slobodnog softvera svuda u svetu. Poslovna okruženja koja veoma zavise od softvera su ovo polako počela da primećuju, i mnogi od njih su otkrili da već koriste slobodan softver u svakodnevnim operacijama a da toga nisu ni svesni. Korporacije su počele da uzimaju aktivniju javnu ulogu u razvoju softvera slobodnog koda, doprinoseći vreme i opremu, a nekad čak i direktno finansirajući razvoj. Ovakve investicije su im se mogle isplatiti višestruko, zato što sponzor plaća samo mali broj ekspertnih programera da se posvete projektu kompletno, ali koristi prednosti kojima su doprineli svi, uključujući i rad neplaćenih volontera i programera koje plaćaju druge kompanije.

2.2.4. “Slobodan softver” nasuprot “softveru otvorenog koda”

Kako je korporativni svet sve više i više obraćao pažnju na slobodan softver, programeri su bili suočeni sa problemima oko prezentacije. Jedan od problema je bila i sama reč slobodan (engleski free). Do zabune dolazi zato što ova reč na engleskom može biti shvaćena i kao besplatan softver. Istina je da sav slobodan softver jeste često i besplatan, ali ne važi i obrnuto, odnosno nije sav besplatan softver slobodan. Na primer, tokom bitke web čitača (battle of browsers) u 90-im, i Netscape i Microsoft su svoje konkurentske web čitače davali besplatno, da bi dobili što veći udeo u tržištu. Ali izvorni kod ovih čitača nije bio dostupan, pa ni oni nisu bili slobodan softver. Čak i ako je bilo moguće doći do izvornog koda, nije bilo moguće menjati ga i redistribuirati. Jedina stvar koja je bila moguća je download izvršne verzije i izvršavanje. Znači web čitači nisu bili slobodni ništa više od paketskog softvera koji je bilo moguće kupiti u prodavnici, samo su bili jeftiniji.

Zabuna oko reči “free” je zbog višeznačnosti u engleskom jeziku. Većina drugih jezika razlikuje nisku cenu od slobode (besplatno nije isto što i slobodno na srpskom jeziku npr.). Ali pozicija engleskog, kao de facto jezika za internacionalnu komunikaciju na Internetu, je ovaj problem učinila u neku meru globalnim. Iako je smišljeno par krilatica kojima se razjašnjava ovaj nesporazum (“free as a speech, not free as a beer” je bila jedna od njih), i dalje je neophodnost čestog objašnjavanja dovela do toga da mnogi programeri smatraju reč “free” neadekvatnom.

Ali problem je bio dublji od toga. Ova reč je sa sobom nosila neizbežnu moralnu konotaciju: ako je sloboda cilj sama za sebe, onda nije bitno da li je slučajno slobodan softver bolji ili čak i profitabilniji za neke primene. Ovo su bili manje prijatni sporedni efekti motiva koji je, niti tehnički, niti profitni, već moralni. Šta više, pozicija „slobodnog“ softvera je zbunjivala neke korporacije koje su htele da podrže neke slobodne programe u jednom delu svog poslovanja, ali da nastave sa prodajom svojinskog softvera u drugim.

Te dileme su se pojavile u zajednici koja je već imala neke krize identiteta. Programeri koji su pisali otvoreni kod se ustvari nikad nisu slagali oko ukupnog cilja koji žele da postignu. Ne bi se čak moglo reći ni da su se mišljenja kretala između 2 ekstrema, jer je bilo toliko aspekata da su se stavovi kretali u multidimenzionalnom prostoru. Ipak, 2 široke kategorije ubedjenja mogu su se mogle primetiti, ukoliko ignorišemo male razlike. Jedna grupa je zauzimala Stolmanov pogled na stvari, da je sloboda deljenja i modifikacije najvažnija stvar, i da ako prestanete da pričate o slobodi, da ste promašili poentu. Drugi su osećali da je softver sam po sebi najbitniji argument, a da svojinski softver nije loš sam po sebi. Neki programeri slobodnog softvera, su verovali da autor (ili poslodavac ako se radi o plaćenom poslu) treba da ima pravo kontrole uslova distribucije, i da ne treba da postoji moralna osuda bez obzira na izbor kontrole.

Dugo ove razlike nisu bile pažljivo ispitane i artikulirane, ali je teret slobodnog/besplatnog softvera u poslovnim svetu učinio to pitanje neizbežnim. Termin otvoreni kod (open source) je kreiran 1998. godine kao alternativa reči "free", od strane skupa programera koji su nešto kasnije postali Open Source Initiative (OSI). OSI je tvrdio da je "slobodan softver", ne samo potencijalno zbunjujuće, već da je dilema oko izbora reči samo jedan simptom generalnog problema: da je pokretu neophodan marketinški program da bi se približio korporativnom svetu, i da priče o moralu i društvenim koristima nikada neće imati efekta u upravnim odborima korporacija. Njihovim rečima: *"OSI je marketinški program za slobodan softver. To je baziranje ovog koncepta na čvrstom pragmatičnom osnovu radije nego na ideološkom tumananju. Supstanca koja dobija se nije promenila, gubitnički stav i simbolizam jesu... Korporativni direktori nikad neće kupiti free softver. Ali ako uzmemo istu tradiciju, iste ljude i iste licence slobodnog softvera i promenimo labelu u open source, to će želeti da kupe."*[3].

Na sajtu organizacije OSI se može pročitati kompletan njihov proglas u kom objašnjavaju svoju nameru, između ostalog da dodaju ono što nedostaje pokretu slobodnog softvera – dobar marketing, gde dobar znači prihvatljiv u poslovnom svetu. OSI inicijativa je dala velikom broju ljudi baš ono što su tražili – rečnik za razgovor o slobodnom softveru kao razvojnoj metodologiji i poslovnoj strategiji, umesto kao o moralnoj borbi.

Pojava Open Source Inicijative je promenila ambijent slobodnog softvera. Formalizovala je dihotomiju koja je dugo bila neimenovana, a time je primorala pokret da obznani da ima i internu i eksternu politiku. Efekat danas je da su obe strane (GNU i OSI) morale da nadju zajedničku osnovu, jer mnogi projekti uključuju programere iz oba kampa, kao i učesnike koji ne pripadaju ni jednoj kategoriji. Zato je danas retka priča o motivaciji u svetu razvoja open source softvera, odnosno da li programer to radi iz moralnih razloga ili zato što je za to plaćen ili gradi svoj CV ili nešto drugo. Doprinosu radu se procenjuje na tehničkim osnovama i odgovara se na tehničkim osnovama, pa sve češće postoje integracije slobodnog sa ne-slobodnim kodom.

2.3. Pravna regulativa u softverskoj industriji

Postoji mnogo pravnih aspekata u softverskoj industriji. Primenjena regulativa se u velikoj meri razlikuje medju jurisdikcijama, ali ću ovde opisati elemente koji se često primenjuju.

Generalno, u većini država sveta postoji i pravno se reguliše pojam intelektualne svojine, koji uključuje zaštitu autorskih prava, pravila fer korišćenja kao i specijalna pravila prevencije kopiranja za digitalne medije. Postoji i oblast softverskih patenata, koja je kontroverzna i još uvek evoluiru u Evropi i nekim drugim delovima sveta.

Važna tema u pravnoj regulaciji softverske industrije su softverske licence. One pokrivaju veliki broj pravnih aspekata, izmedju ostalog pouzdanost proizvoda, odgovornost individualnog programera, garancije, poslovne tajne i intelektualnu svojinu.

Ostali aspekti pravne regulacije u softverskoj industriji uključuju privatnost podataka, neautorizovan pristup, cenzuru i mnoge druge teme, ali one neće biti opisane u ovom radu.

1.3.1. Licence

Softverska licenca je pravni instrument koji uredjuje korišćenje i redistribuciju softvera sa zaštićenim autorskim pravima. Tipična softverska licenca daje krajnjem korisniku dozvolu da koristi jednu ili više kopija softvera, inače bi korišćenje predstavljalo kršenje ekskluzivnih prava izdavača po zakonu o autorskim pravima. Efekat je taj da softverska licenca predstavlja obećanje da izdavač softvera neće tužiti krajnjeg korisnika za aktivnosti koje bi inače bile smatrane ekskluzivnim pravima izdavača.

Softverske licence se generalno mogu podeliti u 2 kategorije: svojinske licence i licence otvorenog koda. Karakteristike koje razlikuju ove 2 kategorije licenciranja su značajne u smislu efekta koji imaju na prava krajnjih korisnika.

Svojinske softverske licence

Žig svih svojinskih softverskih licenci je da izdavač softvera daje dozvolu na korišćenje jedne ili više kopija softvera, ali da vlasništvo nad tim kopijama ostaje izdavaču (zato se i zovu svojinske, eng. proprietary). Posledica ove karakteristike je da su sva prava nad softverom zadržana od strane izdavača, samo veoma ograničen skup dobro definisanih prava se predaje krajnjem korisniku. Zato je tipično da ove licence uključuju puno stavki koje izričito zabranjuju određene načine korišćenja softvera, čak uključujući i one koji bi inače bili dozvoljni zakonom o autorskim pravima.

Najznačajniji efekat ove forme licenciranja je što vlasništvo ostaje izdavaču softvera, a krajnji korisnik mora da prihvati licencu. Drugim rečima, bez prihvatanja licence, krajnji korisnik uopšte ne može da koristi softver.

Jedna od poznatih svojinskih softverskih licenci je licenca za Microsoft Windows. Ona se naziva End User License Agreement ili skraćeno EULA, i koristi je većina kompanija koja proizvodi svojinski softver. Kao što je obično slučaj sa svojinskim softverskim licencama, EULA sadrži opširnu listu aktivnosti koje su zabranjene, od čega su neke reverzni inženjering, simultano korišćenje softvera od strane više korisnika i objavljivanje benchmark i testova performansi.

Kada EULA ugovori prate paketski softver, oni se prezentuju korisniku na papiru ili mnogo češće elektronski, tokom postupka instalacije. Korisnik može da bira da li će prihvatiti ili odbiti ugovor, bez obaveze da ga prethodno pročita, ali se instalacija može obaviti samo ako korisnik prihvati

ugovor i time se složi sa njegovim uslovima. Kada jednom instalira softver, korisnik ima mogućnost da pročita ugovor do detalja.

Termin „upakovana licenca” se odnosi da bilo koju softversku licencu koja se nalazi unutar softverskog paketa i nije dostupna dok god se ne obavi kupovina. Obično je licenca odštampana na papiru unutar zatvorenog pakovanja, ili se prikazuje korisniku na ekranu tokom instalacije, u kom slučaju je nazivamo „upakovana klik licenca”. Nemogućnost korisnika da ima uvid u licencu pre nego što kupi softver je više puta izazvala dovodjenje u pitanje ovakvih licenci na sudu.

Mnogi EULA ugovori nameću ozbiljna ograničenja. Često EULA ogradjuje proizvođača softvera od štete koju softver može da izazove računaru ili podacima korisnika, i čak štete koja može nastati od nepravilne upotrebe softvera (npr. penali za pogrešno izračunat porez softverom za tu namenu).

Da bi se štitile poslovne tajne unutar softvera, ugovori mogu braniti reverzni inženjering korisnicima. Zabrana reverznog inženjeringa služi da oteža neodobreni razvoj softvera koji kooperira sa licenciranim softverom, i time poveća vrednost rešenja proizvođača.

Open source licence

Kod open source licenci, suprotno od svojinskih, vlasništvo nad konkretnom kopijom softvera ne ostaje kod izdavača, već se prebacuje na krajnjeg korisnika. Kao rezultat, krajnji korisnik ima sva prava koja zakon o autorskim pravima dodeljuje vlasniku kopije. Treba primetiti da “vlasnik kopije” nije isto što i “vlasnik autorskih prava” (tj. “vlasnik copyright-a”). Dok se vlasništvo nad konkretnom kopijom prenosi, vlasništvo nad autorskim pravima ostaje kod izdavača softvera. Pored toga, softverske licence otvorenog koda obično daju krajnjem korisniku i neka dodatna prava, koja su inače rezervisana samo za izdavače softvera.

Primarna posledica ovog oblika licenciranja je da je prihvatanje licence otvorenog koda u principu opciono – krajnji korisnik može da koristi softver bez prihvatanja licence. Ipak, ako želi da koristi bilo koje od dodatnih prava koja daje licenca otvorenog koda (npr. pravo da redistribuira softver), onda ga mora prihvatiti i biti vezan njim.

Postoji puno licenci otvorenog koda, od čega su najpoznatije i najkarakterističniji primeri GPL i BSD licence. Iako su obe ove licence “slobodne”, BSD je argumentovano slobodnija. Za razliku od GPL-ovog copyleft-a (popularan naziv ove licence koji ističe da je njen efekat suprotan od copyright-a), korisnici BSD softvera mogu da rade sa kodom šta god žele. BSD licenca dozvoljava programerima da koriste, modifikuju i redistribuiraju izvorni kod i izvršnu verziju originalnog softverskog programa sa ili bez modifikacija. Ovaj tip licence daje korisnicima slobodu da inkorporiraju svoje promene i redistribuiraju ih bez potrebe da objave novi izvorni kod ili da omoguće besplatnu redistribuciju. To omogućava i inkorporiranje u komercijalne proizvode zatvorenog koda. Najnoviji operativni sistem kompanije Apple, Mac OS X, uključuje BSD kod i objavljen je kao projekat otvorenog koda, ali pod drugom, restriktivnijom licencom. Microsoft Windows NT i njegovi naslednici, Windows 2000, XP i Vista, takodje uključuju BSD kod.

Sa druge strane, Free Software Foundation je izmislio koncept copyleft-a, odnosno koncept da kod nikad ne može posati privatn. Ovo znači da privatna firma ne može uzeti Linux kod i koristiti ga da kreira zatvoreni softver. Konkretna formulacija koja ovo označava u GPL-u je: “Sav rad koji se distribuira ili objavljuje, koji sadrži ceo ili deo programa ili je izveden iz istog, mora biti licenciran pod istim uslovima bez ikakve naplate za sva treća lica”. U ovome leži sva snaga, ali i sav kritikizam GPL licence. Problem je između ostalog to što, kao posledicu, ova licenca izaziva da svi programi izvedeni iz licenciranog koda moraju biti besplatni. Ako programer želi da pozajmi nešto GPL koda i koristi ga u svojinskom programu, on krši GPL i može biti tužen da otvori svoj kod. Mnogim ljudima se ne dopada to što su čak i najmanji delovi koda zaštićeni GPL-om.

Između ostalog, neiznenadjujuće protiv je i Majkrosoft, koji licencu naziva virusnom, jer ona inficira sve čega se dotakne, stavljajući ga pod svoju kontrolu. Ovaj kriticizam nije neopravdan (ali se uvek odbacuje kao tipičan za Majkrosoftovu poslovnu taktiku), jer ako neko citira pasus iz knjige, on mora da citira autora, ali konačan proizvod ne mora da bude licenciran kao originalni rad.

Neke od ostalih licenci otvorenog koda su Affero General Public License (AGPL), Apache Software License 2.0, Common Development and Distribution License, Common Public License 1.0, Eclipse Public License, GNU Lesser General Public License (LGPL), ISC license, MIT License, Mozilla Public License (MPL) 1.1...

1.3.2. Softverski patenti

Patent je skup ekskluzivnih prava koja država daje vlasniku patenta na ograničen period vremena, obično 20 godina. Ova prava se daju aplikantima za patent u zamenu za to što oni prikazuju svoje otkriće javnosti. Kada je jednom patent odobren u određenoj zemlji, nijedna osoba ne može da koristi zaštićen izum ni na koji način na teritoriji te zemlje bez dozvole vlasnika patenta. A kada se dozvola daje, to je obično u formi licence čije uslove određuju vlasnici patenata: može biti gratis, uz fiksnu materijalnu nadoknadu, ili uz pravo na procenat ostvarenih prihoda korišćenjem tog patenta.

Patenti su po prirodi teritorijalni. Da bi zaštitio otkriće patentom, pronalazač mora da podnese zahtev za odobravanje patenta u svakoj zemlji u kojoj želi da ostvari zaštitu. Na primer, različite prijave moraju biti podnešene u Japanu, Kini, SAD-u i Indiji ako želite da registrujete patente u tim zemljama. Ipak, postoje neke regionalne kancelarije, npr. European Patent Office (EPO), koji je supernacionalno telo koje ima moć odobravanja patenata koji važe u svim zemljama članicama. Postoji i internacionalna procedura za podnošenje jedne internacionalne prijave kod Patent Cooperation Treaty (PCT) organizacije, koja onda može da ostvari pravo na patent podnosiocu u skoro svim zemljama, ali ovo ne važi za sve vrste patenata. Ove različite zemlje i regionalne kancelarije imaju različite procedure i standarde za odobravanje, što posebno važi za softver ili računarski implementirana otkrića, naročito ona gde softver implementira poslovni metod.

Softverski patent nema univerzalno prihvaćenu definiciju. Ukoliko znamo da je patent jedan od načina pravne regulative sa ciljem zaštite intelektualne svojine u oblasti naučne inovacije, intuitivno je moguće razumeti na šta se pojam softverskog patenta odnosi. Ipak, zbog specifičnosti softvera kao naučno-tehničke osobine, ima mnogo neslaganja oko toga kako definisati patent u ovoj oblasti.

Takodje postoji i intenzivna debata o tome u kojoj meri softverski patenti treba da budu odobravani, ako uopšte treba da budu. Važna pitanja u vezi sa softverskim patentima obuhvataju postavljanje granice između patentabilnog i nepatentabilnog softvera, zatim da li se uslovi inventivnosti i neočiglednosti (koji su dva osnovna uslova pri ispitivanju patentibilnosti) previše lako zadovoljavaju za softver, kao i da li softverski patenti ohrabruju ili obeshrabruju inovaciju. Ova debata je posebno dugo nerešena u Evropi i nema jedinstvene politike po ovom pitanju.

Što se tiče podnošenja zahteva, u prijavi za patent mora postojati detaljno objašnjenje kako napraviti i koristiti otkriće, tako da druge osobe prosečnih veština u toj oblasti mogu da kreiraju i koriste izum bez dalje eksperimentacije. Šta više, patenti važe samo ako su otkrića koja oni štite bila nepoznata javnosti pre podnošenja prijave za patent, i ako otkrića nisu očigledna osobi prosečne veštine u toj oblasti u vreme kada je prijava podnešena (postoje neke razlike u različitim zakonodavstvima oko toga da li se posmatra trenutak kada je podneta prijava ili kada je došlo do

otkrića). Inače patent je formalni zakon, i ukoliko je prekršen može voditi ka invalidaciji patenta na sudu.

Vremensko kašnjenje od trenutka kada je podnet zahtev za patentom do kad on postane javan je 18 meseci. Ovo je kompromis koji sa jedne strane omogućava inovatorima da razviju svoj softver pre nego što budu otkriveni detalji o njemu i time konkurentima bude dat nefer pogled na rezultate tudjeg istraživanja i razvoja, i sa druge strane obaveštavanja javnosti da bi bilo vremena da drugi razviju svoje tehnologije. Format u kom su softverske inovacije date kao patenti (tekst, chart-ovi, crteži i sl.) omogućava osobi sa solidnim veštinama programiranja da kreira softver koji izvodi patentirane ideje, i to je određeno zakonom.

Kako postoji oštra debata o tome da li softverski patenti uopšte treba da postoje, nabrojaću ovde najčešće navodjene argumente za i protiv.

Agrumenti u korist softverskih patenata

Jedan od najčešće navodjenih argumenata za postojanje softverskih patenata je da on promoviše razvoj, odnosno da promoviše investiranje u istraživanje i razvoj. Osnovni principi zakona o patentima su razvijeni još pre nego što su postajali računari i važili su u svetu godinama. Ovi principi su služili promovisanju razvoja u tehničkim disciplinama. Oni koji podržavaju softverske patente tvrde da su otkrića u softveru korisna u modernom životu i zbog toga zaslužuju isti tretman koji imaju otkrića u drugim tehničkim disciplinama.

Sledeći argument govori o pravu javnosti na informisanost, odnosno da patent mora biti javno dostupan da bi time edukovao javnost i prikazao pravu sliku napretka u nauci (tzv. *state-of-the-art*). Tako se tvrdi da patenti ubrzavaju razvoj softvera, time što prethodno nepoznata i neočigledna softverska otkrića čine javnim.

Takodje, jedan od argumenata je da bi organizacije trebalo da budu u moćnosti da štite svoju intelektualnu svojinu. Iako su neki aspekti softverskog proizvoda pokriveni copyright zakonom, ovo se dosta razlikuje od zaštite ideja i inovacije koju pruža patent, a koja postoji u ostalim tehničkim disciplinama. Dok zaštita patenta pruža inovatoru zaštitu ideje, zaštita autorskih prava (copyright zakoni) samo štiti konkretnu manifestaciju te ideje. Često se kao primer navodi poredjenje između slike kao umetničkog dela (koja bi se štitila autorskim pravima), i inovativnog stila slikanja te slike (koji bi se štitio patentom). Šta više, tvrdi se da, za razliku od patenta, zakon o copyright-u inhibira inovaciju u softveru zato što pruža zaštitu softveru bez zahteva za prikazom izvornog koda ili ideje zaštićenog programa, što bi kod registracije patenta bilo neophodno.

Opšta ekonomska korist je jedan od razloga koji se često navodi kao glavna prednost softverskih patenata. SAD vode u broju kompanija, broju zaposlenih ljudi i prihodima u ovoj oblasti, verovatno između ostalog i zato što imaju najbolji sistem zaštite intelektualnih prava. Zbog toga oni koji podržavaju softverske patente tvrde da isti ohrabruju stvaranje softverskih kompanija i novih poslova, i uopšte jačanje industrije.

Zanimljiv argument je i da softverski patenti koji nastaju proizvodnjom patentabilne ideje mogu povećati vrednost malih kompanija. Tužbe za patente su jedan od načina na koji je moguće suprotstaviti se velikim igračima na tržištu (npr. Majkrosoftu) i dozvoliti da inovativne male kompanije izgrade svoje tržište ili bar prime fer kompenzaciju za svoju investiciju. Male kompanije normalno mogu da priušte patentiranje inovacije do kojih dodju (što obično košta oko \$20,000), a u suprotnom postoji mnogo banaka i drugih investitora koji su voljni da pomognu.

I na kraju, kao krajnje argument se navodi da jednom odobren patent može biti poništen ako se utvrdi da nije validan. Time razvoj novih ideja ni na koji način nije blokiran lošim patentima, i oni nemaju negativnih efekata. Ako javnost smatra da je patentirana previše generalna ideja, može

se zatražiti ponovno ispitivanje. Ipak, na ovaj način je do sad bilo izazvano veoma malo odobrenih patenata, jer ponovna ispitivanja obično traju 2-5 godina i mogu biti veoma skupa.

Argumenti protiv softverskih patenata

Jedan od najčešće navodjenih razloga protiv softverskih patenata je da se zbog ove regulative gube sredstva koja bi inače bila uložena u istraživanje i razvoj. Argumentira se da je cena utvrđivanja da li konkretan softver krši neki od registrovanih patenata previše visoka i da su rezultati previše nesigurni. Programer mora da uposli advokata za patente koji će vršiti proveru da programer u svom softveru ne koristi neki registrovani patent, a nema garancije da će ta pretraga biti kompletna jer različiti patentni mogu koristiti različite reči za opis istih koncepata. Ovakve pretrage mogu biti veoma skupe za manje budžete ili individualne inovatore. Takođe se tvrdi da je cena registracije patenata previše velika, i da se patentni odobravaju za relativno mala inkrementalna povećanja u softveru, pa je često nepohodno tražiti prava za korišćenje desetina patenata.

Kao razlog se navodi i da su cene distribucije softvera minimalne u poredjenju sa cenom proizvodnje fizičke robe, pa se samim tim ne mogu primenjivati ni metode zaštite koje su namenjene fizičkoj robi. Argumentira se da je softver poseban slučaj jer nije neophodna proizvodnja svake nove kopije, već samo kopiranje koje je generalno besplatno.

Grupe za kampanju protiv softverskih patenata tvrde da tradicionalna zaštita autorskih prava pruža dovoljnu zaštitu i da nema smetnji masivnim investicijama u razvoj softvera. Konkretno se navodi i da je generalno loše uvoditi patente na ideje, da to sprečava korišćenje ideje i time šteti društvu kao celini. A kako je sav softver sastavljen od ideja, isti navode i da nije moguće povući liniju između toga šta treba da bude patentirano a šta ne.

Argument je i da dobar deo softverskih patenata pokriva trivijalna otkrića ili ona koja bi bila očigledna osobama prosečnih veština u oblasti i u vreme u koje je došlo do otkrića. Kao problem se navodi to što oni koji ispituju patente retko imaju dovoljno znanja konkretne tehnologije, i ne mogu da procene da li je nešto trivijalno ili previše očigledno. Ovo je zbog toga što postoji ogroman broj mikro-niša u oblasti softvera, a relativno mali broj ispitivača. Ovaj argument je diskutabilan zato što je odobren patent moguće povući ako je trivijalan ili očigledan.

Mnogi programeri se žale na to što se aplikacije za patente često ne podnose sve dok novo otkriće ne postane široko korišćeno, pa programeri ne mogu da znaju da li će korisna ideja biti patentirana u budućnosti i time nedostupna. Generalno aplikacije moraju biti podnete pre nego što ideja postane javna, a izlaze u javnost 18 meseci pošto je zahtev podnet. Ipak, u SAD je ovaj uslov promenjen tako da postoji jednogodišnji period od kad otkriće postane javno, do kad inovator mora da podnese zahtev.

Kako između velikih kompanija sa bogatim portfoliom patenata postoje dogovori o uzajamnom korišćenju patenata bez ikakvih nadoknada, još jedan od čestih argumenata je da ovakav poredak utiče na mala i srednja preduzeća, jer ona nemaju veliki odbrambeni portfolio. Neke od kompanija koje međusobno dele portoflije su Microsoft, Cisco, IBM, Sun i druge.

U 2005, prosečno čekanje na odobravanje patenta u SAD-u je bilo 3 i po godine, a u Evropi 4 godine. Zato nije čudo što se kao jedan od argumenata protiv patenata navodi dugo čekanje na ostvarivanje ovog prava. Do trenutka kada se dobije odobrenje patenta, otkrića koja on pokriva se već smatraju javnim, pa su česta neovlašćena kopiranja a investitor ne može da pokrene tužbu. Ipak, ovo je redak slučaj jer je rok od 18 meseci obično dovoljan da treća lica znaju da je moguće da će ta ideja biti patentirana. Postoji i posebna aplikacija za ubrzanu prijavu patenta.

Softverski patenti omogućavaju investicionim kompanijama kupovinu patenata i zatim pokretanje parnica u cilju ostvarivanja prihoda od kupljenog patenta. Neki smatraju uvredljivim da kompanija koja nije kreirala softver može da koristi od patenta za taj softver. Drugi smatraju da patente kupuju veoma špekulativni investitori od kompanija za proizvodnju softvera kojima su potrebne investicije. Ove investicione kompanije se nekad nazivaju „patentskim parazitima“ ili „patentskim trolovima“ i posledica su potencijalno visokih nadoknada na sudjenjima za neovlašćeno korišćenje patenta.

1.3.2.3. Primeri softverskih patenata

Za kraj priče o softverskim patentima ću navesti nekoliko primera odobrenih patenata:

- Digitalni proces enkodiranja (US 5579430) – ovo je jedan od nekoliko patenata koji pokrivaju MP3 format čiji je vlasnik Fraunhofer društvo, a koji je doveo do razvoja Ogg Vorbis formata kao alternative MP3 fajlovima.
- Metod za rangiranje čvorova u povezanoj bazi podataka (US 6285999) – ovo je patent koji štiti Guglovu tehnologiju rangiranja rezultata pretrage PageRank.
- Sistem i metod uticanja na poziciju u listi rezultata pretrage (US 6269361) - patent u vezi sa pay-per-click oglašavanjem na stranama Internet pretrage. Vlasnik je Yahoo! Search Marketing.

Primer prijave za patent koja je izazvala veliku buru u softverskoj zajednici je Majkrosoftova prijava za patentiranje Emoticon-a, odnosno popularnih smajlija (US 20050156873). Prijava još uvek nije ni odobrena ni odbijena, ali objašnjava zašto su softverski patenti tako kontroverzno pitanje. Ipak, to što postoje načini zloupotrebe, može samo značiti da treba popravljati postupak revizije aplikacija, a ne automatski i ukinuti softverske patente.

2. Ekonomske paradigme razvoja softvera

2.1. Ekonomska funkcija softvera u poslovanju

Važno je pomenuti da većina kompanija nije u poslu proizvodnje softvera. One prodaju brodove, cipele, usluge, ili bilo šta drugo. Ipak, svim, pa čak i najmanjim kompanijama je neophodan softver da bi radile. Bez njega bi njihovo poslovanje bilo manje efikasno ili nemoguće. Samo treba pokušati zamisliti finansijsko planiranje bez spreadsheet-ova ili korespodenciju bez email-a. Danas nam je softver potreban u poslovanju u toj meri da čak iako većina firmi ne prodaje softver, svaka firma sa više od 50 ljudi će verovatno zaposliti programera, web dizajnera ili sistem administratora za skript programiranje. Za takve poslove, softver je esencijalna operativna tehnologija, a ne njihov proizvod.

Operativna tehnologija je esencijalna za poslovanje, ali to nije ono što ta firma prodaje. Na primer, ako kompanija prodaje knjige, knjige su izvor profita, a softver je trošak, neizbežan trošak poslovanja. Postoje 2 osnovne vrste operativne tehnologije koja je trošak: diferencirajuća i ne-diferencirajuća. Diferencirajuća je ona tehnologija koja proizvod konkretne kompanije čini privlačnijim za potrošača u odnosu na proizvode konkurenta. Npr, ako posetimo web sajt Amazon.com i tražimo knjigu, Amazon će nam takodje reći koje su još knjige kupili ljudi koji su gledali knjigu koju mi trenutno gledamo. I vrlo često, knjige koje se tu navode jesu dovoljno zanimljive da i mi poželimo da ih kupimo. Na sajtu Barnes and Noble ne postoji ova opcija, pa i nije iznenadjenje što Amazon prodaje više knjiga online. Tako, za Amazon, softver za preporučivanje jeste diferencijator poslovanja. Očigledno, bilo bi greška otvoriti kod poslovnih diferencijatora, zato što bi onda konkurenti mogli da ih iskoriste da bi postali podjednako poželjni potrošaču.

Sa druge strane, našem poslovanju ne bi smetalo kada bi konkurenti videli kako radi svaki deo našeg ne-diferencirajućeg softvera. Naprotiv, konkurent bi u tom delu mogao da bude najbolji kolaborator (ako je partnerstvo ograničeno na rad na ne-diferencirajućem softveru), zato što su njihove potrebe najbližnje vašim. Ova činjenica se demonstrira svaki dan u svetu otvorenog koda, u kome su HP i IBM partneri u razvoju softvera koji pomaže prodaju sistema oba snabdevača, dok ostaju ljuti konkurenti na višim niovima softvera gde je diferencijacija izmedju njih moguća i efektivna.

Možda 90% softvera u bilo kom poslovanju je ne-diferencirajući. Većina tog softvera se naziva i smatra infrastrukturom, odnosno bazom na kojoj se gradi diferencirajuća tehnologija. U kategoriju infrastrukture spada operativni sistem, web serveri, baze podataka, serveri za Java aplikacije i ostali middleware, grafičke radne površine korisničkog interfejsa i generalni alati za GUI radne površine kao što su web čitači, email klijenti, spreadsheet procesori i procesori reči i prezentacione aplikacije. Bilo koji softver koji pruža diferencirajuću vrednost nesoftverskoj kompaniji se gradi preko jedne ili više ovih infrastrukturnih komponenti.

Važan indikator toga da li je softver diferencirajući ili ne je da li konkurent može doći do istog softvera. Ni Windows ni Linux ne mogu pomoći da se kompanija diferencira jer su dostupni svima. Oni se razlikuju medju sobom, ali oni ne mogu biti prednost nekog drugog biznisa. Jedan ili drugi može uštedeti nešto novca ili firmu učiniti efikasnijom, ali generalno ne čine poslovanje privlačnijim potrošačima.

Drugi važan indikator da li je softver diferencirajući je da li potrošač može da vidi efekte tog softvera. Potrošača ne interesuje koji operativni sistem kompanija koristi, osim ako sistem nije oboren sve vreme. Takodje ih ne zanima da li koristi Microsoft Office ili OpenOffice. Kompanija možda imate puno razloga da bira jedan od njih, ali potrošače to ne zanima jer ih ne vide.

Tako, da bi kompanija učinila svoje poslovanje poželjnijim potrošačima, treba da troši više na diferencirajući softver koji čini njene proizvode poželjnijim, a manje na softver koji ne diferencira njenu kompaniju od konkurenata. Jedan od načina trošenja manje na nediferencijatore je korišćenje softvera otvorenog koda, jer se time distribuira cena i rizik sa samo jedne kompanije na više kompanija koje saraduju.

Naravno, za ovo je potrebna iscrpna analiza toga šta je diferencirajuće a šta nije u softveru koji kompanija koristi, a ovo nije uvek lako. Postoji puno kompanija sa NNO („Nije Napravljeno Ovde“) sindromom. To je kada menadžeri i tehničko osoblje nisu voljni da razmatraju rad ljudi van kompanije, jer ne veruju da je dobar kao onaj koji oni prave. Ovaj sindrom je skup i izazvaće dupliranje napora koji postoje na drugim mestima, umesto provodjenja vremena na diferenciranje softvera koji je najbitniji za konkretan posao (detaljnije u [1]).

2.2. Ekonomske paradigme razvoja softvera

Kako je softver neophodan za poslovanje, mora da bude razvijen na neki način. Osnovne ekonomske paradigme razvoja softvera su kupovina, razvoj unutar kompanije ili sklapanje ugovora sa nekom drugom kompanijom da to uradi za nas, zatim kolaboracija bez licenci otvorenog koda, i najzad razvoj u zajednici otvorenog koda.

Svaka od navedenih se od ostalih razlikuje u tome kako distribuira cenu razvoja, kako distribuira rizik od neuspeha, po efikasnosti u finansiranju samog razvoja softvera u poredjenju sa transakcionim troškovima procesa dobijanja softvera, kao i po stepenu u kom drugi mogu biti ograničeni u korišćenju softvera. Ovi faktori odredjuju koja paradigma je odgovarajuća za korišćenje konkretne vrste softvera. Ovde je bitno istaći da je ultimativni izvor finansiranja razvoja softvera potrošač a ne proizvođač, i on može da usmeri svoja sredstva u razvoj korišćenjem bilo koje paradigme da bi dobio softver kakav želi.

2.2.1. Paradigma kupovine gotovog softverskog proizvoda

Paradigma kupovine gotovog softvera je najbliža prosečnoj osobi, a samo 30% svog softvera se razvija i prodaje na ovaj način. Ovde cenu razvoja softvera obično iznosi jedan proizvođač. Proizvođač teži da povрати novac uloženi u razvoj, plus da ostvari profit, prodajom završnog proizvoda. Zbog toga je sav trošak razvoja neophodno uložiti unapred, pre nego što proizvođač može da počne da ostvaruje prihod. Rizik neuspešne proizvodnje profitabilnog proizvoda je kompletno na proizvođaču.

Ako se proizvod pokaže kao uspešan na tržištu, cena razvoja se distribuira potrošačima. Kako paradigma prodaje ne može direktno da distribuira cenu i rizik dok proizvod ne sazri, proizvođači često moraju da se okrenu tržištu investicija kao spoljnom mehanizmu distribuiranja cene i rizika. Potreba proizvođača za spoljnim kapitalom je dugoročna, jer mogu biti potrebne godine da bi se razvio softver i tržište za njega, i dodatno vreme pre nego što cena razvoja može biti amortizovana i dok proizvod počne da ostvaruje profit. Berze povećavaju likvidnost investicije tako što omogućavaju investitorima da monetizuju percepciju budućeg potencijala kompanije (reflektovano u ceni akcije kompanije), umesto refleksije na prihode kompanije kojih još uvek nema. Uspešne kompanije mogu reinvestirati profit u razvoj sledećeg softverskog proizvoda, radije nego da ga vrata na tržište investicija.

Transakcioni troškovi tradicionalnog modela paradigme prodaje softvera (tzv. modela „cigla i malter“) su izuzetno veliki, toliko veliki da se procenjuje da manje od 10% novca koji plaća krajnji

korisnik za softverski proizvod ide u razvoj softvera, dokumentaciju i marketing proizvoda. Majkrosoft je potrošio 16.8% svojih prihoda iz 2004. na istraživanje i razvoj proizvoda. Ostatak prihoda je otišao na stavke od kojih potrošač nema direktnih koristi, odnosno transakcione troškove kao što su advertajzing, dizajn i proizvodnja atraktivnih pakovanja, plaćanje maloprodajama za mesto na policama, osoblje za prodaju i naravno na profit. Cifra od 16.8% ne uključuje marže prodavaca, tako da se dobija da je odnos troškova razvoja softvera i cene proizvoda i manji od 10%.

Neefikasnost postoji i na strani kupovine, zbog nepoklapanja između zahteva potrošača/ kupca i funkcionalnosti softvera koji se kupuje. Kupac često kupuje softver za koji se, pri detaljnijoj proceni, ispostavlja da nije dovoljno koristan za željenu primenu. Potrošač skoro nikad ne može da povрати uloženu cenu ovog proizvoda sa police. Gubici postoje i jer softverske kompanije propadaju ili prestaju da izdaju nove verzije proizvoda, pa ukidaju korisničku podršku. Kako generalno izvorni kod nije dostupan ni za proizvode koji se ne doraduju, a nema ni podrške, potrošač može samo da otpiše svoju investiciju.

Kombinovanjem ovih faktora rizika, dolazimo da procene da se svega 50% kupljenog softvera stvarno i koristi, ili se ne koristi efektivno pa je kupovina neuspešna. Ako ovo pomnožimo efikasnošću od (manje od) 10% paradigme kupovine softvera, nalazimo da je efikasnost finansiranja razvoja softvera kupovinom gotovog proizvoda manja od 5%. Ova procena se odnosi na ukupnu ekonomsku efikasnost paradigme, a ne na efikasnost gledanu iz ugla potrošača.

Najvažnija implikacija ove izuzetno male efikasnosti je da paradigma kupovine ekonomski može da bude korišćena samo za kreiranje proizvoda za masovna tržišta. Masovna tržišta će maskirati neefikasnost paradigme, zato što svaki potrošač može platiti relativno malu cenu u poredjenju sa cenom razvoja softvera. Ukupno, oni će i dalje plaćati više nego 20 puta cenu razvoja softvera. Naravno, ukoliko funkcionalnosti ovog softvera zadovoljavaju zahteve kupca, kupac ima računicu da softver kupi.

Postoji mnogo važnih softverskih proizvoda koji jednostavno ne mogu biti kreirani korišćenjem paradigme prodaje softvera, jer nemaju dovoljno veliko tržište da amortizuju cenu razvoja i velike transakcione troškove paradigme. Šta više, mnogi novi proizvodi koji bi možda kad-tad mogli da izgrade veliko tržište neće biti razmatrani, jer kompanije i investitori ne mogu biti ubedjeni da bi se to tržište zaista razvilo, ili smatraju da je rizik neuspeha previše visok. Ovo u neku ruku onemogućuje inovaciju u softverskoj industriji koja koristi ovu paradigmu.

Jedan od primera neuspeha inovacije kod paradigme prodaje softvera je činjenica da je najvažnija inovacija softverske industrije (govorimo o komercijalnoj, ne naučnoj inovaciji) u poslednjoj dekadi, web server i web čitač, su razvijeni u open source zajednici, u istraživačkim laboratorijama državnih univerziteta. Nijedna od kompanija koja je mogla da kompletira posao nije bila ubedjena da će moći da zaradi novac na ovim projektima. Šta više, jedina kompanija koja je investirala značajna sredstva u razvoja web-a (Autodesk, investiranjem u Xanadu projekat Teda Nelsona), je prekinula projekat jer je model prihoda koji je Xanadu predvideo za web (uplate proizvodjačima sadržaja za svaku reč, sa komplikovanim sistemom praćenja izvedenih radova i referenci) bio previše komplikovan za razvoj. Tim Berners-Lee, koji se često naziva liderom i idejnim tvorcem uspešnog web-a kakav znamo danas, uopšte nije razmišljao o razvijanju bilo kakvog kanonskog modela prihoda, već je ostavio korisnicima da smisle načine za zaradu na web-u.

Takodje, paradigma prodaje softvera mora da ponudi proizvod za što širi skup kupaca, da bi generisala najveći mogući profit. Zbog ovoga je proizvod generalno dostupan svima, pa je teško ili nemoguće iskoristiti ga za diferenciranje poslovanja u odnosu na konkurente.

2.2.2. Razvoj unutar kompanije ili ugovoreni razvoj

Razvoj unutar kompanije obavljaju programeri zaposleni u kompaniji, dok ugovoreni razvoj obavlja spoljna kompanija, koja ekskluzivno biva unajmljena za dogovoreni posao. U oba slučaja, programeri su generalno plaćeni za posao pisanja softvera, pre nego za softver kao proizvod, što je bio slučaj kod paradigme kupovine.

Drugi oblik ugovorenog razvoja je detaljno proširenje i specijalizacija proizvoda snabdevača koji je u osnovnoj varijanti svima dostupan. Na primer, srednja ili veća kompanija može da kupi standardno okruženje web servera, a onda da plati proizvođaču za specijalizaciju tako da odgovara konkretnim potrebama te kompanije.

Potrošač ima odličnu kontrolu nad razvojem unutar kompanije i nad ugovornim razvojem, zato što programeri neće biti plaćeni ako ne urade ono što kupac želi. Kupac generalno može da diktira i da li će softver biti ekskluzivno samo njegov ili će biti dostupan ostalima. Pošto kupac može da kontroliše distribuciju, ovo je izuzetna paradigma za razvoj diferencirajućeg softvera. Šta više, verovatno je ovo jedina paradigma koja zaista omogućava razvoj softvera koji će diferencirati poslovanje kupca.

Generalno, kontraktor će sigurno pokušati da iskoristi neki deo posla koji je obavio za jednog korisnika da brže obavi posao za drugog korisnika, odnosno da ponovo proda posao za koji je već bio plaćen. Koliko je kontraktor uspešan u ovome, zavisi od uslova ugovora i njegove iskrenosti. Ako je kontraktor samopouzdan da može da preproda rad koji obavlja za jednog kupca, on može da ga naplati manje, i kupac može da ima koristi od distribucije cene i rizika. Ipak, ovo generalno nije zgodan scenario za diferencirajući softver: ako kontraktor preprodaje kupčeve diferencijatore posla, kupčevo poslovanje može da bude u problemu.

U zamenu za potpunu kontrolu, potrošač/kupac generalno snosi kompletnu cenu i rizik razvoja. Zbog toga, ovo možda nije najefektivniji metod, po pitanju cene, za razvoj nediferencirajućeg softvera. Ako kupac plaća 100% cene razvoja novog softvera koji duplira funkcije postojećeg softvera koji bi bio dostupan potrošačima za manje, to se naziva izmišljanjem vruće vode i predstavlja uludo trošenje sredstava. Ako kupac može da podnese da se njegov softver distribuira drugima, i nije mu potrebna apsolutna kontrola nad razvojem, paradigma kupovine ili paradigma razvoja u zajednici otvorenog koda može biti efektivnija po ceni za taj softver.

Razvoj unutar kompanije ili ugovaranjem je razumno efikasan u usmeravanju velikog procenta ulaganja ka samom razvoju softvera. Efikasnost u ovome je 50% do 80%, za razliku od 10% kod paradigme kupovine. Glavni izvori neefikasnosti su cene nalaženja novih potrošača, cene održavanja ekspertize koja se ne koristi sve vreme i sl.

Odredjeni procenat projekata razvijenih ovom paradigmom nikad ne uspe da proizvede softver koji se koristi u poslovanju kupca i da postigne ciljeve koji su planirani. Kao procenat uspeha se može uzeti 50%, slično ukupnom procentu paradigme prodaje.

2.2.3. Kolaboracija bez licenci otvorenog koda

Standardan način saradnje medju kompanijama na razvoju softvera je bio konzorcijum. Zatvoreni konzorcijumi za razvoj softvera imaju istoriju velikih neuspeha. U skorije vreme, zatvoreni konzorcijumi vredni milione dolara su zamenjeni uspešnijim projektima otvorenog koda. Kao primer mogu se navesti Taligent i Monterey, 2 konzorcijuma okupljena u nameri kreiranja zamene za Unix. Linux ih je prestigao i zamenio. Takodje, Common Desktop Environment je zamenjen open source GNOME radnom površinom u većini kompanija koje su podržavale CDE.

Pre nekoliko godina, velika agrikulturna korporacija Cargill je osnovala konzorcijum sa namerom korišćenja prednosti razvoja otvorenog koda, dok u isto vreme pruža tajnost i deli benefite softvera samo sa članovima konzorcijuma. Dve godine kasnije, Cargill je napustio projekat koji je finansirao. Zatvoreni konzorcijum se jednostavno pokazao kao pogrešna struktura za razvoj nediferencirajućeg softvera. Ima smisla širom otvoriti vrata kada ne štitite diferencijaciju, i primiti članove koji mogu da korisno doprinesu čak i kad ne mogu da pomognu u finansiranju. Takođe konzorcijum košta više od projekta otvorenog koda zato što ima manje članova koji dele cenu i rizik, a ima mnogo više strukture i transakcionih troškova nego što bi bilo u ekvivalentnom projektu otvorenog koda. Kod zatvorenih konzorcijuma, ljudi se plaćaju i postavljaju da rade u interesu kompanije koja ih plaća, a kod open source-a je to obično po pokazanim tehničkim zaslugama u razvoju. Zato se može desiti da član radi na štetu celog projekta, ako to njemu koristi. Produkciono planiranje se često svede na nerešive rasprave medju kompanijama, zato što svaka ima različitu marketinšku ideju i argumente.

Kada se uzme u obzir nesrećna istorija razvoja u konzorcijuma, i sa druge strane visok stepen uspeha velikih projekata otvorenog koda, čak i kad su u pitanju iste kompanije, izgleda da je red koji se uvodi licencama otvorenog koda esencijalna komponenta efektne kolaboracije izmedju velikog broja strana sa različitim interesima.

2.2.4. Paradigma razvoja otvorenog koda

Kod paradigme razvoja otvorenog koda, više entiteta (individue, kompanije, akademske institucije) se okuplja u cilju razvoja softverskog proizvoda. Generalno, inicijalni razvoj obavlja jedan entitet kao razvoj unutar kompanije, a softver se pušta u javnost, odnosno kod se otvara čim postane koristan drugima, a najčešće pre nego što se može smatrati završenim proizvodom i samim tim mnogo ranije nego što bi bio pušten da se radi o paradigmi kupovine. Kada je softver jednom upotrebljiv i koristan, drugi entiteti počinju da ga upotrebljavaju. Tek tada paradigma razvoja otvorenog koda počinje da funkcioniše, jer pre toga drugi entiteti nemaju interesa da koriste i proširuju softver. Proširivanjem softvera se bave individue ili zaposleni entiteta koji koristi softver.

Inkremetalna cena dodavanja karakteristike je mnogo manja nego cena celog razvoja. Strane koje kreiraju modifikacije imaju interes da te modifikacije pišu tako da one budu prihvaćene od strane drugih programera sa projekta, i da budu unešene u zvaničnu verziju izvornog koda. Ako do ovog spajanja ne dodje, cena održavanje te promene kroz naredne verzije je mnogo veća, jer je onda sa svakom novom verzijom neophodno dodavati ih i održavati kompatibilnost sa bazom koda koja se stalno menja.

Zbog toga otvoreni kod okuplja zajednicu zainteresovanih programera koji doprinose korisnom proizvodu. Cena i rizik razvoja proizvoda se distribuiraju medju programerima, a bilo koja kombinacija tih programera može da nastavi da radi na projektu ako ostali odu. Distribucija cene i rizika počinje čim je projekat dovoljno zreo da izgradi zajednicu širu od inicijalnog programera.

Projekat otvorenog koda razvijaju direktno njegovi krajnji korisnici. Npr, funkcionalnosti Apache web servera dodaju kompanije kojima su te funkcionalnosti potrebne da bi izvršavali svoje web sajtove, ili ih dodaju kontraktori koji rade za te kompanije.

Korisnici konkretnog proizvoda otvorenog koda se generalno sami identifikuju: oni traže proizvod u direktorijumu open source softvera, onda ga download-uju i testiraju. Ako su zadovoljni, počinju da ga koriste. Time stiču kontinuirani interes za proizvod, a u nekom trenutku, ako poželev dodatne funkcionalnosti imaju interes da postanu ko-programeri softvera koji koriste.

Kompanije koje se pridruže razvoju projekta otvorenog koda, koriste taj softver kao nediferencirajući, odnosno on im je trošak a ne izvor prihoda. Ovim kompanijama nije bitno što

razvoj otvorenog koda ne donosi profit: njihovi izvori profita su neke druge stvari, a softver je tehnologija koja omogućava rad. Da bi mogli da nastave sa ostvarivanjem prihoda iz svojih izvora, oni moraju da investiraju u infrastrukturu. U slučaju diferencirajućeg softvera, verovatno će morati da koriste razvoj unutar kompanije ili ugovoreni razvoj, jer će morati da spreče da diferencijator dodje u ruke konkurenta. Za nediferencijatore, postoji izbor između kupovine ili paradigme razvoja otvorenog koda. Sada ću diskutovati koji je metod od ova 2 efikasniji, u smislu opšte ekonomske efikasnosti, kao i u smislu pogodnosti sa strane kupca..

Kako se korisnici sami identifikuju, open source nema neefikasnosti paradigme prodaje, koja mora da koristi advertajzing i druge skupe mehanizme da bi privukla korisnike. U najmanjoj meri je podjednako efikasna u usmeravanju sredstava u razvoj softvera u odnosu na transakcione troškove kao razvoj unutar kompanije ili ugovoreni razvoj. Zbog ovoga open source paradigma može biti korišćena za razvoj proizvoda koji nisu namenjeni masovnom tržištu, i koji iz tog razloga ekonomski ne bi mogli biti razvijeni paradigmom kupovine. Ipak, od konkurencije na tržištu konkretnog proizvoda, ukoliko isto postoji, zavisi da li je za kupca isplativije da kupi gotovi proizvod ili se uključi u razvoj u zajednici.

Naravno, ne uspeju svi projekti otvorenog koda. Većina nezrelih projekata umre na samom početku, ili zauvek ostane solo projekat. Ipak, trošak takvih projekata je jako mali, jer oni nikad i ne privuku značajan broj korisnika ili programersku zajednicu, pa te projekte možemo isključiti iz razmatranja. Zreliji projekti otvorenog koda mogu prestati da žive kada naidje nešto bolje, ali je često moguće iskoristiti izvorni kod, podatke i veštine iz jednog projekta za drugi, pa tako investicija u razvoj projekta nije izgubljena.

Cena učestvovanja u zrelom projektu otvorenog koda se veoma razlikuje od cena kupovine ili razvoja unutar kompanije. Glavni izdatak su vreme i cena zaposlenih koji učestvuju. Detaljnije, u ovo cifru ulazi osoblje koje procenjuje i testira softver, oni koji prilagođavaju softver, odnosno rade na razvoju i oni koji rade kao podrška korisnicima unutar kompanije. Rizik predstavlja mogućnost da se ulaže u softver koji će bit zamenjen zbog nemogućnosti da se prilagodi ili da se isprate potrebe korisnika. Maksimalna cena ovakvog razvoja je u slučaju kada nema programerske zajednice osim tog konkretnog korisnika, i u tom slučaju je cena ista kao cena razvoja unutar kompanije ili ugovaranjem, gde jedan potrošač finansira ceo razvoj. Stvarna cena u skladu sa ovim zavisi od broja aktivnih participanata i potrebnog rada. Izgubljena investicija je vreme osoblja. Uzimajući sve ovo u obzir, opšta ekonomska efikasnost je velika barem kao kod paradigme razvoja unutar kompanije i ugovaranjem, a puno veća nego kod paradigme kupovine.

2.3. Zaključak o ekonomskim paradigmatama razvoja softvera

Paradigma razvoja otvorenog koda ima nekoliko značajnih ekonomskih prednosti nad ostalim paradigmatama: ona kombinuje efikasnost u usmeravanju resursa u sam razvoj softvera sa boljom distribucijom cene i rizika. Nju je moguće primeniti na razvoj proizvoda koje nije moguće razviti korišćenjem paradigme kupovine jer nema masovnog tržišta. Open source distribuira cenu i rizik proizvoda, dok razvoj unutar kompanije fokusira svu cenu i sav rizik na jedan entitet. Open source takodje ne zahteva korišćenje investicionih tržišta za distribuiranje cene i rizika, a distribucija počinje mnogo ranije u odnosu na paradigmu prodaje. Otvoreni kod daje korisniku kompletnu kontrolu nad specijalizacijom i prilagođavanjem proizvoda. Ipak, ne daje korisniku kontrolu nad tim ko ima pristup proizvodu. Zbog toga open source paradigma nije dobar mehanizam za razvoj diferencirajućeg softvera. Jedina odgovarajuća paradigma za razvoj diferencirajućeg softvera je razvoj unutar kompanije ili ugovoreni razvoj.

Poslovanja koja zahtevaju nediferencirajući softver (a to su ustvari sva poslovanja) mogu da prebace deo svoje infrastrukture i usmere razvoj na open source projekte, i time će bez sumnje postići veću ekonomsku efikasnost. Ovo je dobra poslovna strategija jer time ostaje više sredstava za ulaganje u diferencirajući softver koji utiče na profit. Ipak, kod konkretne odluke treba uzeti u obzir i ponude na tržištu odgovarajućih gotovih proizvoda.

Tabela 2. Poredjenje ekonomskih paradigmati razvoja softvera

Paradigma	Efikasnost	Stopa neuspeha	Distribucija cene	Distribucija rizika	Štiti diferencijaciju korisnika	Štiti diferencijaciju snabdevača	Potrebna veličina tržišta
Kupovina gotovog proizvoda	manje od 10%	50%	kasno, nekad nakon što počne prodaja	Ne	Ne	Da	Masovno tržište 100,000 ili više, za specijalizovana tržišta manje
Unutar kompanije i ugovaranjem	60% do 80%	50%	Ne	Ne	Da	Možda	1
Konzorcijumi i saradnja zatvorenog koda	60% do 80%	Možda 90%, neprihvatljivo visoka	Da	Da	Možda	Možda	5 i više
Razvoj otvorenog koda	60% do 100%	50% ako isključimo nezrele projekte	Rano, tokom razvoja	Da	Ne	Ne	5 i više

3. Primeri poslovnih modela u softverskoj industriji

Kao što je već pomenuto, danas nije jednostavno napraviti razliku između kompanija koje ostvaruju prihod od softvera, i onih kojima je softver samo pomoćno sredstvo. Ipak, u ovom delu rada ću pokušati da navedem što više poslovnih modela u softverskoj industriji, da ih objasnim i navedem kompanije koje uspešno primenjuju ove modele.

Modeli će biti opisani iz perspektive proizvođača i efekata koje razvoj i prodaja softvera ili usluga u vezi sa softverom imaju na poslovanje tog proizvođača. Za svaki model će biti razmatrani aspekti ulaganja, rizika, period do ostvarivanja prihoda, lakoću ulaska na tržište, konkurenciju itd. Za svaku kompaniju koja bude navedena kao primer, biće data kratka istorija, način na koji kompanija implementira konkretan model, udeo u tržištu, rast i prihod kompanije kroz istoriju, kao i imidž i kritike na njen račun.

U nastavku slede poslovni modeli, grupisani po domenima:

Modeli u domenu prodaje softverskog proizvoda

a) Prodaja softverskog proizvoda: Ovo je jedan od najčešćih i najpoznatijih poslovnih modela u softverskoj industriji. Osnovni izvor prihoda softverske kompanije je od pravljenja kopija svog softvera i prodaje uz određenu licencu. Ovaj model će biti detaljno objašnjen na primeru kompanije Microsoft.

b) Dualno licenciranje: Ovim se označava model razvoja softvera, gde je softver dostupan i pod licencom otvorenog koda, pa ga je moguće slobodno koristiti, modifikovati i distribuirati, dok god su sve promene takodje dostupne i otvorene. Ipak, softver je dostupan i pod drugom, svojinskom licencom, pa ako korisnik želi da koristi softver u komercijalnom softveru zatvorenog koda, on mora da plati za to, odnosno da kupi svojinsku licencu. Ovaj model će detaljno biti opisan na primeru kompanije MySQL AB.

c) Online korišćenje uz pretplatu: neke softverske kompanije čine softver dostupan svojim korisnicima za online korišćenje. Prihodi se u ovom slučaju ostvaruju na nekoliko načina – od periodičnih pretplata do jednokratnih uplata za uslugu. Aplikacije koje koriste ovaj poslovni model su vrlo često vodjene korisničkim podacima, odnosno predstavljaju baze podataka neke vrste informacije za koju su ostali korisnici spremni da plate. Primer su ACM digitalna biblioteka naučnih radova, online časopisi itd. Ulaganje se svodi na održavanje servera, a lakoća ulaska na tržište zavisi od toga da li sličan servis već postoji na Internetu. Obično je potrebno neko vreme dok „efekat mreže“ ne proradi, i aplikacija dobije dovoljnu količinu sadržaja da bi bila interesantna korisnicima. Kritika kompanija koje rade po ovom modelu se uglavnom odnosi na etičke implikacije prodavanja informacija.

d) Softver sa pretplatom: Kod ovog poslovnog modela, krajnji korisnik kupuje softver, i time ima pravo na bilo koja poboljšanja ili update-e softvera koje snabdevač objavi tokom perioda subskripcije. Čest primer ovog tipa softvera su antivirusni softveri koji moraju biti update-ovani regularno da bi bili efektivni. Pošto subskripcija istekne, korisnik može da zadrži softver, ali više nema prava na update-e. Ovaj model je sličan modelu prodaje softverskog proizvoda, razlika je u boljoj distribuciji cene i rizika razvoja u toku životnog ciklusa proizvoda.

e) Softver za poboljšanje poslovnog procesa: Ovaj poslovni model je sve aktuelniji i zahteva da softver demonstrira poboljšanje efikasnosti određenog poslovnog procesa pre nego što proizvođač može da dobije bilo kakvu materijalnu kompenzaciju. Softverska kompanija će analizirati poslovne procese budućeg klijenta, i identifikovati oblasti u kojima softver snabdevača može da poboljša efikasnost. Klijent ne plaća ništa direktno za upotrebu softvera ali plaća na

osnovu metrike izvedene sa obe strane. Ove metrike mogu uključivati uštede postignute kao rezultat upotrebe softvera u poslovanju klijenta. Npr, snabdevač može da dobije \$1.00 za svaki novi trag dobijen kroz web sajt snabdevača, ili može da dobije 2% uštede koju je postiglo konkretno odeljenje zbog korišćenja softvera. Ako upotreba softvera ne rezultira pozitivnom metrikom, softverska kompanija ne dobija ništa. Tok isplata može da bude iseckan ili sa jednom velikom isplatom u prvoj godini i manjim ili većim u narednim godinama, na osnovu rezultata. Kod ovog modela je sve u diferencijaciji u odnosu na konkurente, pa je nemoguće razvijati ovakav softver otvorenog koda. Takodje, sva cena i rizik razvoja se iznose unapred, odnosno nema ostvarivanja prihoda dok kompletan proizvod ne bude završen. Postoji i rizik da softver neće dovesti do poboljšanja poslovnog procesa, pa u tom slučaju prihoda uopšte neće ni biti. Ipak, ukoliko je ovaj model uspešno implementiran, ostvareni prihodi mogu biti veoma veliki.

Modeli u domenu softverskih usluga

f) Usluga programiranja: Ukoliko korisnik ne želi sam da razvija potreban softver, niti da ga kupi (zato što odgovarajući gotovi proizvod ne postoji ili zbog diferenciranja među konkurentima), on može sklopiti ugovor sa nekom programerskom kompanijom o razvoju tog softvera. Ipak, u tom slučaju je nemoguće sklopiti ugovor o kupovini gotovog proizvoda, jer to obično nije ni u interesu korisnika kome softver treba – zbog rizika od nepoklapanja zahteva sa dobijenim rešenjem, niti u interesu softverske kompanije – zbog distribucije troškova. Ovo se rešava tako što se plaća vreme programera dok razvijaju traženi softver, odnosno usluga programiranja, dok god se ne dobije zadovoljavajuće rešenje.

g) Provajeri aplikacionih usluga (Application Server Providers, ASP): Generalno, ASP poslovni model zavisi od želje krajnjeg korisnika softvera da outsource-uje trećoj strani vlasništvo i održavanje hardvera na kome se softver izvršava. Host je vlasnik računarskih servera, mrežne i komunikacione opreme na kojoj se softver izvršava. Host je odgovoran za održavanje hardvera i softvera, i naplaćuje korisniku za ovu uslugu. Postoji mnogo varijacija na ASP temu. Ponekad krajnji korisnik licencira softver direktno od snabdevača softvera i prenosi ga hostu. Ponekad, softverske kompanije izvode aktivnosti hosta za krajnjeg korisnika. U drugim slučajevima, host može licencirati softver od snabdevača softvera i naći mušterije koje kupuju pristup kapacitetima hosta.

h) Instalacija softvera: Zavisno od tipa softvera i nivoa sofisticiranosti klijenta, snabdevač softvera može da pruži uslugu instaliranja softvera na opremi klijenta. Ovaj model, kao i onaj prethodni, spada u domen usluga, a ne same proizvodnje softvera. Modeli gde se softverski proizvod daje, a prodaje se sva usluga u vezi sa softverom (instalacija, održavanje, podrška, edukacija i trening...), su sve češći u poslednje vreme.

i) Održavanje softvera: Često snabdevač i korisnik softvera sklapaju ugovor o održavanju. Ovakav ugovor obavezuje snabdevača da klijentu pruža popravke bagova, softverske pečeve i nove verzije proizvoda, kada i ako budu dostupni. Takodje, ovakvi ugovori mogu obavezati snabdevača da pruža tehničku podršku ili trening. Ponekad, snabdevač neće dati dozvolu (licencu) za korišćenje softvera ako se ne potpiše ugovor o održavanju, i tada se model naziva „obaveznim održavanjem“. Ovaj model je detaljno obrazložen na primeru Red Hat kompanije.

j) Tehnička podrška: Softverske kompanije često pružaju tehničku pomoć zaposlenima kupca softvera, kada ovi naidju na tehničke probleme u radu. Tehnička podrška je često deo ugovora o kupovini softvera, odnosno softverske licence. U suprotnom, sklapa se posebni ugovor kojim se snabdevač softvera obavezuje da će pružati tehničku podršku u dogovorenom roku, uz dogovorenu cenu, fiksno, po obradi problema ili satu pružanja podrške. I ovo spada u domen softverskih usluga.

k) Obuka, trening i sertifikacija: Kod ovog modela, snabdevač softvera pruža obuku i trening zaposlenima klijenta o korišćenju softvera. Ovo je veoma važno kada klijent kupuje poslovni sistem za menadžment informacija koji će biti potpuno nov kompletnoj radnoj snazi, i neophodna je obuka da ne bi došlo do pada produktivnosti i efikasnosti. Neki veliki brendovi u IT industriji izdaju sertifikate koji garantuju nivo veštine u korišćenju njihovih softverskih paketa ili hardverskih sistema (Cisco, Oracle, Microsoft, Red Hat, Novell). Ovaj model takodje spada u domen usluga.

Modeli kombinovanja sa drugom granom industrije

l) Poboljšanje prodaje hardvera softverom: U svetu informacionih tehnologija, gde su promene i inovacije česte, kao i veliki obrti u samoj industriji, skoro da je nemoguće dugoročno planiranje. Proizvođači računarske opreme obično imaju veća ulaganja od proizvođača softvera, a svaka promena tehnologije, do koje dolazi na svakih nekoliko godina, može dovesti do toga da čak i najveće hardverske kompanije imaju velike gubitke. Jedan od načina na koji se proizvođači kompletnih računarskih sistema bore sa ovim promenama, je vezivanjem kvalitetnog softvera za njihov hardver, nudjenjem kompletnih jeftinijih rešenja i izbora kupcima. Ovaj model je detaljno objašnjen na primeru jedne od najvećih kompanija u svetu IT-a, IBM-a.

m) Prodaja oglasnog prostora na web stranama: Ovaj model računa da će softversku aplikaciju koristiti odredjen broj korisnika, ali se prihod neće ostvarivati od tih korisnika, već od reklamiranja na web stranama aplikacije. Ukoliko je za korisnike servis besplatan, to mora biti aplikacija sa velikom bazom korisnika – pretraga, web mail, društvene mreže i sl. Ovde posebno do izražaja dolazi efekat mreže, a model će detaljno biti predstavljen na primeru kompanije Google.

Pre nego što predjem na detaljne opise nekih od navedenih modela, daću listu 20 najvećih kompanija u softverskoj industriji. Ova lista je podskup Forbes-ove liste 2000 najvećih javnih kompanija čija je delatnost označena kao „Softver i usluge“. Ipak, ovde nema nekih velikih kompanija koje proizvode softver kao što je HP ili Apple, jer je kao njihova delatnost navedeno „Tehnološki hardver“. U prvoj koloni može se videti rang kompanije na kumulativnoj listi svih delatnosti. Takodje je očigledna i velika dominacija kompanija iz SAD. Kompanije su rangirane po proizvodu prodaje, profita, imovine i kapitalizacije tržišta, koji su respektivno dati u poslednjim kolonama. Vrednosti su date u milijardama dolara.

Možemo videti da su medju najvećim kompanijama prisutni različiti poslovni modeli, od prodaje licenciranog softvera, preko softverskih usluga, advertajzinga na web stranama i pospešivanja prodaje hardvera, do kompanija koje kombinuje više od jednog modela.

Tabela 3. Najveće kompanije u softverskoj industriji u 2007. po Forbes magazinu

Rang	Rang na kumulativnoj listi	Kompanija	Zemlja	Prihod	Profit	Imovina	Kapitalizacija tržišta
1	42	IBM	SAD	91.42	9.49	103.23	139.92
2	66	Microsoft	SAD	46.06	11.91	66.37	275.85
3	205	Oracle	SAD	16.07	3.70	28.93	85.14
4	289	Google	SAD	10.60	3.08	18.47	137.60
5	357	SAP	Nemačka	12.40	2.46	12.44	58.26
6	407	First Data	SAD	7.08	1.51	34.46	19.22
7	518	Electronic Data Sys	SAD	21.27	0.47	17.95	14.48
8	562	Softbank	Japan	9.43	0.49	15.18	26.68
9	582	Yahoo	SAD	6.43	0.75	11.51	41.86
10	728	Symantec	SAD	5.07	0.45	17.90	15.81
11	743	Computer Sciences Corporation	SAD	14.68	0.36	13.10	9.10
12	855	Capgemini	Francuska	10.16	0.39	8.03	9.21
13	1047	Tata Consultancy Svcs	Indija	2.98	0.67	1.93	26.27
14	1048	Fiserv	SAD	4.54	0.45	6.21	9.06
15	1064	Adobe Systems	SAD	2.58	0.51	5.96	23.10
16	1130	Infosys Technologies	Indija	2.14	0.55	2.09	26.19
17	1166	CA	SAD	3.89	0.10	10.11	13.73
18	1130	Infosys Technologies	Indija	2.14	0.55	2.09	26.19
19	1166	CA	SAD	3.89	0.10	10.11	13.73
20	1166	CA	SAD	3.89	0.10	10.11	13.73

3.1. Komercijalni softver zatvorenog koda

Ovo je verovatno najpoznatiji poslovni model razvoja softvera. Kompanija prihod ostvaruje prodajom kopija i licenci softvera kupcima, gde kupci mogu biti krajnji korisnici softvera ili proizvođači računara koji dobijaju dozvolu da instaliraju softver na svoje računare. Takođe, neke softverske kompanije prodaju licence drugim softverskim kompanijama koje "dodaju vrednost" softveru kombinovanjem sa drugim proizvodima ili uslugama, a prvobitnim proizvođačima plaćaju po prodatoj kopiji. Nekad se jedna kopija uz odgovarajuću licencu može instalirati na veći (ali fiksni) broj računara. Kopija je nekad takođe dostupna i download-om sa Interneta.

Kod ovog poslovnog modela je nemoguće koristiti licence otvorenog koda, jer se prodaje sam softverski proizvod, što je kod open source licenci nemoguće.

Veliki broj kompanija posluje po ovom modelu, od čega su najpoznatije Microsoft, Adobe, Corel, SAP, Sybase, Autodesk, CA, Borland i mnoge druge. Većina ovih kompanija su članice Business Software Alliance, organizacije posvećene promovisanju bezbednog i legalnog digitalnog sveta, borbi protiv piraterije i uopšte angažovanja u korist komercijalne softverske industrije. Ovo su ujedno i najveće kompanije u softverskoj industriji, koje čine veliki deo ukupnih prihoda ostvarenih u ovoj industriji.

Kao primer će biti obradjena kompanija Microsoft, koja je najveća (ili druga najveća, zavisno od parametara) softverska kompanija u svetu i klasični predstavnik poslovnog modela proizvodnje i prodaje komercijalnog softvera zatvorenog koda.

3.1.1. Primer poslovanja po ovom modelu - Microsoft

Majkrosoft korporacija je američka multinacionalna korporacija za kompjuterske tehnologije. Ona razvija, proizvodi, licencira i pruža podršku u širokom opsegu softverskih proizvoda za računarske uređaje. Najpoznatiji i najprodavaniji proizvodi ove kompanije su Microsoft Windows operativni sistem i Microsoft Office skup softvera produktivnosti. Ovi proizvodi imaju značajne pozicije u svojim tržištima, sa udelima od 90% ili više za Windows i Microsoft Office. Jedna od ključnih vizija Bila Gejtsa, osnivača, je bila da se softver ove kompanije izvršava na svakom računaru i u svakom domu sveta.

Tabela 4. Rast prihoda i broja zaposlenih u Majkrosoftu

Kraj fiskalne godine	Broj zaposlenih	Neto prihod (u mrd \$)	Rast prihoda	Neto profit (u mrd \$)	Rast profita
30. jun 2007	78,565	51.12	15%	14.07	12%
30. jun 2006	71,172	44.28	11%	12.60	3%
30. jun 2005	61,000	39.79	8%	12.25	50%
30. jun 2004	57,086	36.84	14%	8.17	8%
30. jun 2003	54,468	32.19	13%	7.53	29%
30. jun 2002	50,621	28.37	12%	5.35	-28%
30. jun 2001	48,030	25.30	10%	7.35	-22%
30. jun 2000	39,170	22.96	16%	9.42	21%
30. jun 1999	31,575	19.75	29%	7.79	73%
30. jun 1998	27,055	15.26	28%	4.49	30%
30. jun 1997	22,232	11.94	32%	3.45	57%
30. jun 1996	20,561	9.05	49%	2.20	51%

Korporacija je osnovana da bi razvijala i prodavala interpretatore za BASIC na Altair 8800 uređaju. Majkrosoft se uzdigao do pozicije da dominira tržištem operativnih sistema za kućne računare svojim proizvodom MS-DOS sredinom 1980-ih. Kompanija je imala inicijalnu javnu ponudu na berzi, koja je usled velikog rasta vrednosti akcija četvoricu zaposlenih učinila milijarderima a oko 12000 milionerima.

Kroz svoju istoriju, kompanija je često bila meta kritike iz više razloga, uključujući monopolski položaj, anti-kompetitivne poslovne prakse, odbijanje oko nagodbi i ograničavanje izbora korisnicima. Ministarstvo pravde SAD-a i Evropska komisija su, među ostalima, donosili presude protiv Majkrosofta za monopolne prekršaje.

Majkrosoft ima proizvode i u drugim tržištima pored operativnih sistema i kancelarijskih paketa, kao što je MSNBC mreža kablovske televizije, MSN Internet portal i Microsoft Encarta multimedijalna enciklopedija. Kompanija takodje proizvodi hardverske proizvode kao što je Microsoft miš i proizvodi kućne zabave Xbox, Xbox 360, Zune i MSN TV. Zvanični web sajt kompanije je jedan od najposećenijih na Internetu sa više od 2.4 miliona jedinstvenih poseta stranicama dnevno.

Poslovi i proizvodi

Da bi merenje performansi svakog segmenta bilo precizno, Majkrosoft je organizovan u više poslovnih grupa, od kojih je svaka nezavisan finansijski entitet. Ovi entiteti su restrukturirani kako se menjao fokus kompanije i sama industrija, a trenutno postoje 3 divizije.

Divizija za platformske proizvode i usluge

Ova divizija proizvodi najpoznatiji proizvod kompanije, operativni sistem Windows. On je proizveden u mnogo verzija, uključujući Windows 3.1, 95, 98, 2000, Me, 2003, XP i Vista. Online servis MSN, kablovska tv stanica MSNBC i online magazin Slate su svi deo ove divizije. Krajem 1997, Majkrosoft je kupio Hotmail, najpopularniji webmail servis, a 1999 je uveden MSN mesindžer, klijent za instant razmenu poruka, da bi se takmičio sa popularnim AOL klijentom za IM. Online servisima je u toku 2006 dodat Live portal sa internet pretragom i mnogim drugim funkcionalnostima.

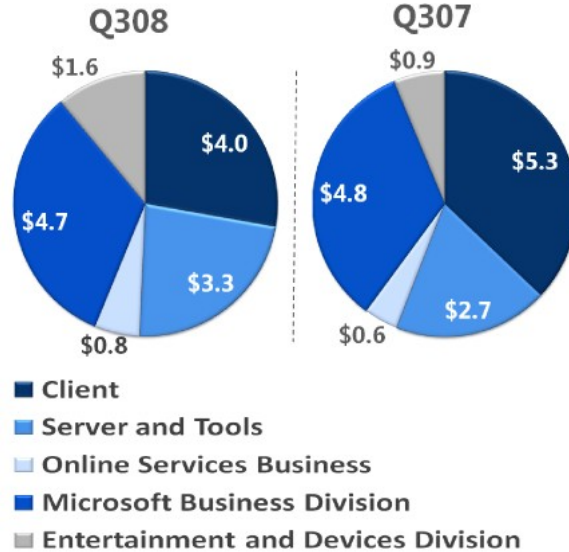
Microsoft Visual Studio, skup programerskih alata i kompajlera, takodje je deo divizije za platformske proizvode i usluge. Softverski proizvodi su GUI orijentisani i lako se vezuju sa Windows API-jem, ali moraju biti specijalno konfigurirani da bi radili sa bibliotekama koje nisu Majkrosoftove. Trenutna verzija je Visual Studio 2008. Takodje je razvijena solidna arhitektura i znatno je pojednostavljen razvoj aplikacija, kako za Windows platformu, tako i za web.

Majkrosoft nudi i skup serverskog softvera, koji se naziva Windows Server System. Windows Server 2003 je operativni sistem za mrežne servere i on je jezgro serverske linije. Drugi serverski proizvod, Systems Management Server, je kolekcija alata koji pružaju mogućnosti udaljene kontrole, menadžement pečeva, distribucije softvera i inventar računarske opreme. Ostali softverski proizvodi uključuju Microsoft SQL Server - RDBMS sistem, Microsoft Exchange Server – email poslovni server, Small Business Sever za razmenu poruka i Microsoft BizTalk Server za menadžement poslovnih procesa.

Poslovna divizija

Ova divizija proizvodi Microsoft Office, što je linija kompanijskog kancelarijskog softvera. Takodje, ovde se razvijaju finansijski i softver za menadžement poslovnih procesa kompanija. Najpoznatiji proizvod u ovoj kategoriji je Microsoft Dynamics.

SEGMENT REVENUE (in billions)



Slika 1. Prihodi Majkrosofta po segmentima poslovanja

Divizija za zabavu i uređaje

Majkrosoft je pokušao da proširi Windows brend na mnoga druga tržišta, sa proizvodima kao što su Windows CE i Windows Mobile za PDA uređaje i smartphone telefone. Fokus je takodje na uređajima gde OS nije direktno vidljiv krajnjem korisniku, npr. kućni aparati i automobili. Kompanije je prodavala i digitalni video rekorder nazvan UltimateTV, koji je dopuštao korisnicima da snimaju i do 35 sati televizijskog programa, ali nisu imali mnogo uspeha u ovome.

Majkrosoft prodaje računarske ige za PC-jeve, izmedju ostalog Age of Empires, Halo i Microsoft Fight Simulator, a ima i online servis za takmičenja u PC igrama. Ušli su i na tržište konzoli za igre kojim dominiraju Sony i Nintendo, svojim Xbox uređajem kojem je trebalo nekoliko godina da postane profitabilan. Kreirano je dosta igara za ovu konzolu. Majkrosoft takodje prodaje i nešto od računarskog hardvera – uključujući miševe, tastature, džojstike i druge kontrolere za igre, čija je proizvodnja mahom outsource-ovana.

Krajem prošle godine najavili su i kupovinu Musivawe-a, online servisa za prodaju muzike na mobilnim telefonima, koji će biti deo divizije za zabavu i uređaje.

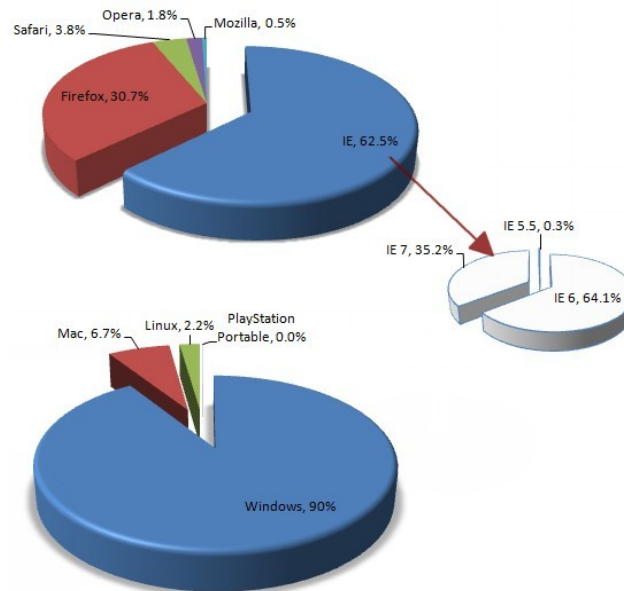
Na Slici 1 možemo videti da Majkrosoft i dalje najveći deo svojih prihoda (oko 90%) ostvaruje od paketskog komercijalnog softvera, pa je zato on najbolji primer kompanije za ovaj model. Ipak, Majkrosoft verovatno smatra da je budućnost informacionih tehnologija u Internet uslugama i mobilnim uređajima, pa ulaže i ove segmente. Šta više, on u pomenutim segmentima ostvaruje prihode od preko \$2 milijarde po kvartalu, što je daleko od zanemarljivog, ali na tim tržištima nema ni približno dominantnu ulogu kao na tržištu paketskog softvera.

Kritike

Kritike su bile usmerene na različite aspekte Majkrosoftovih proizvoda i poslovnih praksi. Problemi sa lakoćom upotrebe, stabilnošću i sigurnošću softvera su česte mete kritike. U skorije

vreme, brojni korisnici su se žalili na prisustvo trojanaca i virusa usled propusta u bezbednosti Microsoft Windows-a i drugih programa.

Kompanija je takodje optuživana za tzv. ograničavanje (lock-in) korisnika i snabdevača, u smislu da je cena prelaska na neki drugi sistem previše velika. Ovo se posebno odnosi na Windows API, jer se jednom razvijena aplikacija za Windows ne može lako prebaciti na neku drugu platformu. Slične su optužbe u vezi specifičnih formata fajlova koje Majkrosoft koristi, a koje ne želi da otvori. Zato se često vodi debata o ukupnoj ceni vlasništva Windows-a u poredjenju sa Linux-om i Mac OS X-om.



Slika 2. Tržišni udeo IE web čitača i Microsoft Windows operativnog sistema

Kompanija je bila optuživana i osuđivana nebrojeno puta, od strane nekoliko vlada i dugih kompanija, za nezakonito monopolsko ponašanje. Pored toga, zajednica slobodnog softvera i otvorenog koda kritikuje Majkrosoft zbog zatvorenog koda i korišćenja veoma restriktivne EULA licence.

Na osnovu svega navedenog može se videti da je Majkrosoft jedna od najmoćnijih kompanija u softverskoj industriji, koja svoje prihode ostvaruje kroz brojne softverske modele, od čega je bez sumnje najdominantniji model prodaje softverskog proizvoda zatvorenog koda.

3.2. Softverske usluge

Softver nije samo proizvod. U kompleksnim sistemima kakvih danas ima dosta, kupovina softvera bi bila tek početak na putu ka njegovom korišćenju. Neophodna je instalacija, zatim obuka korisnika, podrška pri korišćenju, održavanje. Tako se otvorilo celo novo tržište softverskih usluga, do te mere da postoji dosta kompanija koje svoj softver daju besplatno, a prihod ostvaruju naplaćivanjem usluga na tom softveru. U tom slučaju čest je slučaj da je softver ne samo besplatan već i otvorenog koda. Kompanija koja je pokrenula ovo tržište, da bi zatim postala glavni igrač na njemu, je svakako Red Hat.

3.2.1. Primer poslovanja po ovom modelu – Red Hat

Red Hat je jedna od najinteresantnijih kompanija na tržištu softvera danas. Njihov poslovni model je baziran na davanju njihove intelektualne svojine besplatno potrošačima. Oni razvijaju i reklamiraju svoj softver, što košta dosta, ali se nadaju da će ostvariti profit na druge načine. Softver koji oni razvijaju je otvorenog koda, i ovde ćemo opisati jedan od načina ostvarivanja prihoda od softvera otvorenog koda.

Primarni proizvod Red Hat-a je linija operativnih sistema dizajnirana oko Linux kernela, koji se mogu besplatno download-ovati i distribuirati. Red Hat puno investira u razvoj Linux-a, čak su i zaposlili Alana Coxa, koji je drugi posle Linusa Torvaldsa u hijerarhiji razvoja Linuxa. Ovo je uprkos činjenici da ne zaraduju od direktne prodaje svog glavnog proizvoda, već organizuju svoj poslovni model oko proizvoda i usluga kojima se vezuju za ovaj softver. Ovi proizvodi i usluge su izvor prihoda, a softver je sam po sebi "gubitak". Što je Linux bolji, to je poželjnije korišćenje Linuxa i sve više firmi će želeći usluge koje Red Hat pruža. U ovom smislu, Red Hat je više konsultantska firma nego firma za razvoj softvera.

Sada ću opisati linije usluga koje nudi ova kompanija:

Enterprajz rešenja

Softver otvorenog koda je i slobodan i besplatan. Ovo znači da firme biraju kupovinu softvera otvorenog koda (kakav je i Red Hat Linux) zato što vrednuju usluge koje su upakovane sa kupovinom softvera. Skoro da je nemoguće zamisliti bilo koju firmu koja stavlja svoje podatke na server sa operativnim sistemom koji nema korisničku podršku. Objavljivanjem dokumentacije i prodajom direktnih usluga kao što su podrška i update-ovi, Red Hat odgovara na ove potrebe.

Red Hat sada prodaje 3 nivoa svog softvera. Postoji Red Hat Enterprise Linux AS, Red Hat Enterprise Linux ES i Red Hat Enterprise Linux WS. Ove skraćenice su od advanced (napredni), enterprise server i workstation (radna stanica). Sva 3 proizvoda su sastavljena od slobodnog softvera, ali se razlikuju u nivou podrške i naravno ceni. AS proizvod košta \$1499 za standardno izdanje i \$2499 za premium izdanje. AS je, po sajtu Red Hat-a, "ultimativno rešenje za velike departmatske i servere centara podataka". Usluge i podrška za takav sistem su nešto za šta bi svaka kompanija drage volje platila.

ES softver košta \$349 za osnovno izdanje i \$599 za standardno. Razlika između ova 2 modela je u sledećem: osnovno izdanje pruža 90 dana instalacione i konfiguracione podrške (ponedeljak – petak 9AM-5PM ET), i jednu godinu pretplate na Red Hat Enterprise mrežu. Dostupan je samo download-om. Standardno izdanje pruža jednu godinu standardne podrške (ponedeljak – petak 9AM-9PM telefonske podrške sa 4 sata razgovora (9AM-9PM izvan Severne Amerike) i jednu godinu pretplate na RHE mrežu). Potrošači koji naručuju SE će dobiti kompletno upakovan proizvod sa CD-om i štampanom dokumentacijom. Red Hat Enterprise Linux ES pruža iste osnovne mogućnosti kao Red Hat Enterprise Linux AS. Ova vrsta cenovne diskriminacije

dozvoljava Red Hat-u da targetira svoje potrošače na osnovu važnosti platforme za njihovo poslovanje.

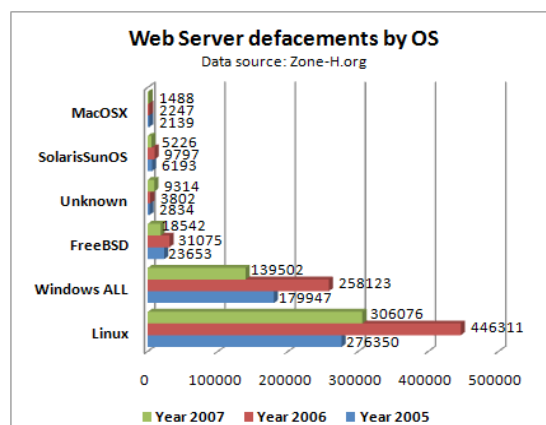
Finalna verzija, WS, je fokusirana na radne stanice. Košta \$179 za osnovno izdanje i \$299 za standardno izdanje. Razlike u usluzi između ova 2 su virtualno identične razlikama na ES nivou. WS dolazi bez mnogih softverskih aplikacija koje su dizajnirane za veće servere (na kojima bi se verovatno trebala izvršavati AS ili ES verzija Red Hat Linux-a). Softver koji targetira sličnu platformu, Microsoft Windows XP Professional, košta \$299 za punu verziju kada se kupuje direktno od Microsoft-a. Nivoi usluge se ne mogu direktno porediti, ali ovo indicira da su usluge i proizvodi koji se kupuju od Red Hat-a na mnogo načina jednaki u vrednosti kao oni koje prodaje Majkrosoft svojim kupcima.

RHE mreža je usluga koja se koristi za update softvera kod sistema sa Red Hat Enterprise Linuxom. Ovo je mnogo više nego običan update sistem, pošto omogućava kompanijama da podese kako žele da update-uju sisteme u svojoj infrastrukturi. Dozvoljava i grupisanje računara tako da različiti administratori imaju različit nivo pristupa, kao i skalabilnost pri proširenju velikog sistema. Dok proširenje jednog sistema može biti trivijalno, održavanje infrastrukture može postati ekspanzivno kompleksnije kako se veličina povećava.

Konsultantske usluge

Red Hat takodje prodaje konsultantske usluge koje idu dalje od regularnih usluga za enterprajz sisteme. Oni pružaju rešenja za kompanije koje žele da migriraju sa svojinskih Unix sistema na Linux, za one koje žele da izgrade IT infrastrukture, za velike proizvođače hardvera koji žele da dodaju Linux u svoju liniju usluga, pa čak i za kompanije koje žele da ugrade Linux u druge hardverske proizvode. Red Hat ima jedinstven pogled na Linux kao razvijač jedne od najpopularnijih distribucija Linuxa i pokušava da unovči ovaj pogled i ekspertizu kroz konsultantske usluge.

Linux je kompatibilan sa standardima za Unix sisteme. Ovo znači da može da izvršava većinu aplikacija koju mogu i konkurentne forme Unix-a. Ovi svojinski Unix sistemi se izvršavaju na skupim hardverskim platformama i softverski update-i mogu koštati nekoliko hiljada dolara. Linux, kao softver otvorenog koda, može drastično smanjiti ove cene. Jedan primer je sajt Amazon.com, koji je uspeo da smanji svoje troškove za \$17 miliona ili 24% u odnosu na godinu ranije, zbog prelaska na Red Hat Linux.



Slika 3. Raspodela tržišta operativnih sistema servera

Gradjenje IT infrastrukture može biti zahtevan zadatak, jer ove infrastrukture mogu uključivati web servere, servere baza podataka, deljenje štampača, fajlova i individualne radne stanice. Greške, u smislu nekompatibilnog hardvera i softvera mogu biti veoma skupe. Suprotno, ispravne odluke

moгу značajno da poprave produktivnost. Red Hat prodaje konsultantske usluge koje pomažu firmama da donesu ove odluke. Pomažu i firmama da isprave njihovu IT infrastrukturu standardizacijom na Linuxu i ostalim otvorenim standardima koji mogu da podrže rast ne ograničavajući firme na korišćenje vlasničkih ili nekompatibilnih standarda. Proizvodnjom sve boljih i boljih verzija Linuxa, nadaju se povećanju interesovanja za gradnjem njihovih sistema oko Linuxa.

Industrija velikih centralnih računara se dramatično promenila u prethodnih nekoliko godina: Cena računarske moći brzo opada po Murovom zakonu i kompanije pokušavaju da iskoriste prednosti ovoga. Linux se izvršava na jeftinijim računarskim arhitekturama, kao što je x86 platforma, na kojoj je i bazirana velika većina PC-ja. Većina firmi u serverskoj industriji je instalirala svojinske Unix sisteme na svojim veoma skupim hardverskim sistemima. Razvoj ovih hardverskih i softverskih linija je dovoljno skup da je bilo svega nekoliko proizvođača. Kompanije kao što je Dell su se udružile sa Red Hat-om da bi razvile liniju servera koje koriste jeftiniji hardver i Red Hat Linux kao operativni sistem. Ovo omogućava kompanijama koje traže rešenja za servere da koriste usluge Red Hat-a da bi razvile svoje linije servera. Partnerstvo sa Red Hat-om takodje omogućava lakše prihvatanje Linuxa kao načina borbe za konkurencijom u trci medju operativnih sistema u serverskoj areni. Red Hat trenutno ima veze sa IBM-om i HP-om, firmama koje su dugo u serverskoj industriji i još uvek proizvode svoje Unix sistema.

Moć računara je postala izuzetno uticajna. Čak i satovi i mobilni telefoni ne funkcionišu više na samo jedan način. Veoma često ovi uređaji su računari na kojima se izvršava ugradjeni operativni sistem kao što je Microsoft Windows, Palm OS ili čak Linux. Za firme koje razvijaju hardverska rešenja, Linux je alternativa sistemima u kojima bi morali da plaćaju proviziju za svaki uređaj koji kreiraju, ili da razvijaju svoj operativni sistem. Firme mogu da unajme Red Hat da dovede Linux na njihove uređaje i umesto da se fokusiraju na razvoj OS-a, mogu se usredsrediti na razvoj aplikacija i hardvera.

Drugi proizvodi i usluge

Red Hat takodje zaradjuje novac kroz usluge kao što su sertifikacija i edukacioni programi, a takodje i kroz prodaju Red Hat brendiranih proizvoda. Edukacioni i sertifikacioni programi, poznati kao Red Hat Certified Engineer (RHCE) i Red Hat Certified Technician (RHCT) uključuju kurseve koji se predaju širom sveta, kao i sertifikacione ispite. RHCE se smatra jednim od 10 najtraženijih sertifikata u IT industriji, i predstavlja lepu prednost na tržištu rada. Cene obuke i polaganja ispita se kreću od \$750 do \$2000 dolara.

Red Hat prodaje i brendiranu robu u svojoj online radnji. Kako je jedina intelektualna svojina nad kojom mogu da tvrde vlasništvo njihovo ime, oni prodaju majce, kape, podloge za miša i sl. Ove promocione stavke obično žele da kupe ljudi koji imaju ideološki stav prema otvorenom kodu ili tehničko poštovanje samog softvera i brenda.

Tabela 5. Raspodela tržišta serverskih distribucija Linuxa

Distribucija	Aktivnih sajtova u julu '07	Udeo u tržištu u julu '07	Šestomesečni rast
Linux – RedHat	1,465,310	50.8	10.1
Linux- Cobalt	596,743	20.8	12.3
Linux – Debian	468,502	15.5	14.5
Linux- SuSE	347,326	10.9	15.6
Linux – Mandrake	37,186	1.3	15.3
Linux – Gentoo	29,912	0.7	49.5

Kritika ka ovoj kompaniji uglavnom dolazi od ideoloških pobornika softvera otvorenog koda, koji smatraju da nije etički što Red Hat zaradjuje na osnovu rada kompletne Linux zajednice. Kontra argument je svakako to da oni Linux svojim uslugama čine popularnijim.

Na najvećem broju servera se izvršava Linux operativni sistem (Slika 3). Kao što se može videti u Tabeli 5, Red Hat je najpopularnija serverska distribucija Linuxa (najpopularnija desktop distribucija je Ubuntu, ispred Red Hat-ove desktop varijante Fedora). Uzevši u obzir koliki procenat servera u svetu je na Linuxu, ne čudi kako Red Hat ostvaruje velike prihode, i kako oni iz godine u godinu rastu. Njihov prihod 2002. godine je bio \$78.9 miliona , a u 2007. je bio \$400 miliona. Zato je ova kompanija proglašena kompanijom sa najvećim rastom u oblasti softverskih usluga.

3.3. Pospešivanje prodaje hardvera softverom

Hardver i softver u računarskoj industriji su neraskidivo povezani. Svaki veliki napredak u hardveru označavao je i promene u softverskoj industriji. Iako ove 2 industrije već 50 godina postoje kao odvojene, proizvođači su svesni da je neophodno klijentu ponuditi kompletno rešenje. Glavni snabdevači hardvera, čak iako ne proizvode svoj softver, imaju partnerstva sa softverskim kompanijama ili doprinose zajednici otvorenog koda. Oni ulažu u softver da bi po što manjoj ceni ponudili što kompletniji i kvalitetniji proizvod, da bi ostali aktuelni i plivali kroz česte promene u industriji. Ovaj model je detaljno objašnjen na primeru jedne od najvećih kompanija u svetu IT-a ikad, na primeru IBM-a.

3.3.1. Primer poslovanja po ovom modelu - IBM

International Business Machines korporacija, skraćeno IBM i sa nadimkom "Big Blue" (Veliki plavi), je multinacionalna korporacija za računarsku tehnologiju i konsalting sa sedištem u Njujorku, SAD. Kompanija je jedna od svega nekoliko kompanija u oblasti informacionih tehnologija sa kontinuiranom istorijom koja datira još iz 19-og veka. IBM proizvodi i prodaje računarski hardver i softver, a nudi i infrastrukturne usluge, usluge hostinga, kao i konsalting u raznim oblastima od mejnfrejmskih računara do nanotehnologije.

IBM je u skorijoj istoriji poznat kao najveća računarska kompanija na svetu sa više od 388,000 zaposlenih u svetu. Uprkos padu iza Hjulit-Pakarda po ukupnim prihodima u 2007-oj, ostaju najprofitabilnija kompanija u računarstvu. Takođe, IBM je vlasnik više patenata nego bilo koja druga američka kompanija u oblasti tehnologije. Imaju inženjere i konsultante u više od 170 zemalja, a IBM Research ima 8 istraživačkih laboratorija u svetu. Uposlenici IBM-a su dobili 3 Nobelove, 4 Turingove i mnoge druge nagrade. Kao proizvođač čipova, IBM je bio među najvećih 20 prodavaca poluprovodnika u svetu u prethodnim godinama, a 2007. su bili i prva ili druga najveća softverska kompanija u svetu (zavisno od parametara koji se porede).

Trenutni softverski projekti

Na IBM je uticala Open Source Inicijativa i on je počeo da podržava Linux 1998. godine. Kompanija je investirala milijarde dolara u usluge i softver baziran na Linuxu kroz IBM Linux Technology Center, koji okuplja preko 300 programera Linux kernela. IBM je takođe objavio svoj kod pod različitim licencama otvorenog koda, između ostalog Eclipse (koji je vredeo oko \$40 miliona u vreme otvaranja koda) i Java baziran DBMS, Apache Derby. Takođe, IBM dosta ulaže i u Apache web server.

Eclipse je platformski nezavisno Java-bazirano okruženje za razvoj softvera. On je prvo bio svojinski proizvod razvijen u IBM-u kao naslednik VisualAge familije alata, a kasnije je pušten kao softver otvorenog koda pod Eclipse Public License (EPL licenca).

DeveloperWorks je web sajt koji IBM održava za programere i IT profesionalce. Sadrži veiki broj how-to članaka i tutorijala, kao i download softvera i primera koda, foruma za diskusije, podcast-ova, blogova, wiki stranica i drugih resursa za programere i tehničke profesionalce. Teme se kreću od otvorenih industrijski standardnih tehnologija kao što su Java, Linux, SOA i web servisi, web razvoj, Ajax, PHP i XML, do IBM-ovih svojinskih proizvoda (WebSphere, Rational, Lotus, Tivoli i DB2).

Zakodje, IBM softverska grupa razvija svojinski softver zatvorenog koda za IBM. Ova rešenja nisu besplatna, a takodje nisu ni infrastrukturna već predstavljaju tzv. softver koji dodaje vrednost, i koji ide iznad osnovne infrastrukture. U opisu poslovnog modela sam želela da istaknem ulaganja IBM-a u razvoj softvera otvorenog koda, ali ovaj softver vredi pomenuti jer on donosi dobar deo prihoda. Svakako, ovakav softver je lakše prodati kada već prodajete kompletno hardversko i softversko infastrukturno rešenje. Neki od brendova ove grupe su:

- Information Management Software — serveri baza podataka, alati za analizu teksta, menadžement sadržaja i poslovnih procesa, kao i poslovna ineligencija.
- Lotus Software — Grupna kolaboracija i poslovni softver (kupljen 1999.).
- Rational Software — Razvoj softvera i menadžement životnog ciklusa aplikacije (kupljen 2002, kao najskuplja transakcija u softverskoj indsutriji ikad, vredna \$2.5 milijarde).
- Tivoli Software — Menadžement sistema (kupljen 1996.).
- WebSphere — Integracioni i softver za infrastrukturu aplikacije.

Koristi od open source-a

IBM je direktno koristio od svoje podrške razvoju softvera otvorenog koda. Ove koristi nastaju od efekta mreže, dugoročnog smanjivanja cena, nezavisnosti od svojinskih standarda drugih firmi i mogućnosti iskorištavanja postojećeg softvera da bi se povećala vrednost njihovog hardvera.

Efekat mreže kao pojava će kasnije biti detaljno opisan, ali ukratko on označava pozitivne efekte koji nastaju od širokog korišćenja nekog proizvoda. Na primer, što više ljudi koristi email, on postaje značajniji kao alat za komunikaciju. Kod softvera, mrežni efekat je činjenica da što više ljudi koristi određeni softver, više razvoja u toj oblasti se dešava, time proizvod postaje bolji, a onda ga koristi još više ljudi. Ovo je osnovno za teoriju razvoja otvorenog koda. Što više IBM podržava Linux, više ljudi ga koristi, i samim tim ima više logike za firme da kupe Linux rešenja. Od ovog IBM ima koristi, kao jedan od glavnih snabdevača servera, konsultantskih usluga i aplikacionog softvera koji se izvršava na Linuxu.

Pomoću otvorenog koda IBM takodje smanjuje cene. IBM ovim ima hiljade programera više po celom svetu, koji rade da bi učinili Linux stabilnijim i boljim proizvodom. Plus, već postoji solidna količina softvera slobodno dostupnog za Linux i Apache. Prelazak na ove platforme instantno daje IBM-u biblioteku softvera koji ovi mogu ponuditi svojim klijentima bez dodatne cene po IBM. Ovaj softver sada više ne mora da se proizvodi unutar kompanije, što znači da IBM može da taj višak novca prebaci u druge oblasti istraživanja i razvoja, a sam proizvod učini jeftinijim i time osvoji veći udeo u tržištu u odnosu na konkurente. Oni više ne rade posao koji je jednom uradjen i uz to dostupan. Važno je i napomenuti da je dobar deo ovog softvera veoma relevantan kupcima servera, baš onom tržištu koje IBM pokušava da uhvati. Linux se takodje često koristi kao hosting platforma za Internet aplikacije na web serveru Apache i skripting jezicima kao što su PHP i Perl. Ovo su sve postojeći programi i platforme koji se i dalje konstantno unapreduju.

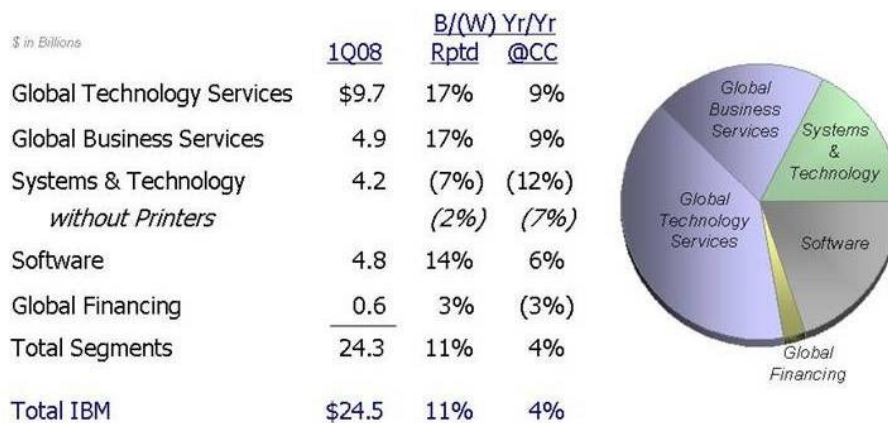
Kada je IBM razmatrao Linux, imao je 3 alternative. Prvo, mogli su da koriste drugu besplatnu distribuciju varijante Unixa, npr. BSD. Ipak, IBM nije želeo da koristi BSD jer je Linux lakše portovati sa jedne hardverske platforme na drugu, a i Linux je generalno prepoznatljiviji od BSD-a. Druge dve opcije su bile nastavak razvoja njihovog operativnog sistema ili kupovina svojinske platforme od druge firme. Analize koje su vršene u to vreme su pokazale da je dalji razvoj nove platforme neisplativ i nepotreban. A ako bi neki drugi svojinski softver bio korišćen, onda bi IBM i njegovi klijenti previše zavisili od nekog trećeg proizvođača, verovatno monopoliste. Takođe, IBM je imao loše iskustvo sa Majkrosoftom, kada su zajedno ulagali u razvoj operativnog sistema OS/2 koju je Majkrosoft napustio da bi razvijao Windows. Zato je izabran Linux.

IBM-ov ulazak na tržište razvoja otvorenog koda je imao uticaja na celo tržište servera. Iako je ova industrija kompleksna i prodaje veliki broj različitih proizvoda, relativno mali broj velikih firmi dominira. Ovo su IBM, Sun, HP i Dell. Od toga IBM, Sun i HP imaju svoje Unix sisteme, ali su svi ušli i u svet otvorenog koda.

Danas postoje 2 glavna tipa operativnih sistema za servere. Jedan je Microsoft Windows, a drugi Unix uključujući Linux. Neki softver, uključujući Apache, se izvršava na obe platforme. U skorije vreme, i Windows i Linux dobijaju udeo u tržištu u poredjenju sa Unix-om, npr. Sun-ovim Solarisom i HP-ovim HP-UX-om. Linux postaje dominantan igrač unutar polja Unixa.

Tržište servera je postalo više tržište usluge poslovnog računarstva nego samo tržište računarskih sistema. Sada paketi koje firme očekuju uključuju i konsalting usluge svih vrsta. Svojom kupovinom PwC Consulting-a od ove internacionalne firme za profesionalne usluge, IBM se odlično pozicionirao u sektoru usluga. Ideja je da će sve više IT usluga biti outsource-ovano a kompanije koja će najbolje umeti da se bavi ovim je IBM, sa svojim znanjem konsaltinga, tehnologije i otvorenih standarda i platformi. Na ovaj način IBM ima kompletno infrastrukturno rešenje, preko kog može da ponudi svoje specifične svojinske proizvode (npr. WebSphere za e-commerce) koji se izvršavaju na softveru otvorenog koda. Za ove aplikacije ne postoji alternativa otvorenog koda, već one dodaju vrednost postojećim platformama.

Da zaključim, da bi maksimizovao svoj profit i udeo u tržištu, IBM se strateški pozicionirao u zajednici otvorenog koda kroz njihovu podršku Linux i Apache platformama. Ovo je IBM-u dalo koristi u ceni, ali im je i dozvolilo da restrukturiraju industriju u njihovu korist, fokusirajući se na usluge i hardver, dok su potpuno izbacili operativni sistem i middleware iz svoje jednačine.



Slika 4. Prihod IBM-a po segmentima poslovanja

Na Slici 4 vidimo upravo rečeno, da je sektor u IBM-u koji najviše zaradjuje upravo sektor usluga.

3.4. Dualna licenca

Dualno licenciranje je praksa distribuiranja identičnog softvera pod dva različita skupa uslova. Ovo može značiti 2 različite licence, ili 2 različita skupa licenci. Softver se ponekad nudi u tri-licenciranju ili multi-licenciranju. Kada je softver dualno licenciran, korisnici mogu birati pod kojim uslovima žele da koriste ili distribuiraju softver. Distributor može ili ne mora primeniti naplatu u bilo kojoj od ove 2 opcije. Tri uobičajene motivacije za dualno licenciranje su isplativost razvoja otvorenog koda, kompatibilnost licenci i segregacija tržišta.

Poslovni modeli

Dualno licenciranje se često primenjuje da bi poslovni model slobodnog softvera bio isplativ i ekonomski održiv. Kod ovakvog modela, jedna opcija je svojinska softverska licenca, koja omogućava kreiranje svojinskih aplikacija izvedenih iz njega, dok je druga licenca otvorenog koda koja zahteva da sav izveden kod bude pod istom tom licencom otvorenog koda. Vlasnik autorskih prava softvera u tom slučaju obično daje verziju softvera otvorenog koda besplatno, a profitira prodajom licence komercijalnim kupcima koji žele da inkorporiraju softver u njihov proizvod. Ovaj model može biti upoređen sa *shareware* pristupom.

Kako uglavnom samo vlasnik autorskih prava može da promeni uslove licence softvera, dualno licenciranje uglavnom koriste kompanije koje kompletno poseduju licencirani softver.

Ovakvo licenciranje omogućava vlasniku da nudi prilagodjavanja, rana puštanja, generiše izvedeni rad ili daje prava trećim stranama da redistribuiraju svojinske verzije, sve dok svima nude besplatnu verziju softvera. Deljenje paketa kao copyleft slobodan softver može doneti korisnike i programere iz zajednice otvorenog koda. Ovi doprinosi mogu biti konkretno kodiranje, usmeni marketing ili opšta podrška. Ipak, objavljivanje i pod svojinskom licencom, pored ove otvorenog koda, nosi rizike od gubitka poverenja i podrške u zajednici otvorenog koda.

Najpoznatiji primeri aplikacija licenciranih na ovaj način uključuju MySQL bazu podataka, Gaia Ajax Widgets i Trolltech-ov Qt skup razvojnih alata.

Kompatibilnost licenci

Druga upotreba dualnog licenciranja kod slobodnog softvera je zbog kompatibilnosti licenci, dozvoljavajući kombinovanje koda iz različito licenciranih softverskih projekata slobodnog koda, ili da bi omogućili korisniku mogućnost izbora licence.

Primeri ove vrste dualnog licenciranja uključuju izvorni kod Mozilla Application Suite-a, Mozilla Thunderbird-a i Mozilla Firefox-a, koji je tri-licenciran pod Mozilla Public License (MPL), GNU General Public License (GPL) i GNU Lesser General Public License (LGPL), kao i Perl, koji je dualno licenciran pod GPL-om i Artistic licencom.

Segregacija tržišta u svojinskom softveru

Dualno licenciranje koriste i neki distributori ne-slobodnog softvera. Ponekad se ovo radi kod svojinskog softvera da bi segregirali tržište. Deljenjem ljudi u kategorije, kao što su kućni korisnici, profesionalni korisnici i akademski korisnici, vlasnici autorskih prava mogu postaviti različite cene za svaku grupu. Ipak, kod svojinskog softvera, „kućno izdanje“ i „profesionalno izdanje“ se obično razlikuju i po uključenom softveru, a ne samo po licenci.

3.4.1. Primer poslovanja po ovom modelu – MySQL

MySQL AB je osnovana 1995, a 2008. je integrisana u Sun Microsystems. Ova kompanija je kreator i vlasnik MySQL, menadžement sistema relacionih baza podataka, kao i povezanih proizvoda kao što je MySQL Cluster.

Sa oko 400 zaposlenih u 25 zemalja, MySQL AB je bio jedna od najvećih kompanija otvorenog koda. Oko 70% zaposlenih je radilo za MySQL od svojih kuća. U januaru ove godine, kompanija je objavila da su postigli dogovor da ih otkupi Sun Microsystems kompanija za oko \$1 milijarde, što predstavlja jedan od 10 najvećih transfera u softverskoj industriji ikad.

Zajedno sa Linux-om, Apache-om i PHP-om, MySQL Server formira jedan od gradivnih blokova LAMP skupa tehnologija. Kompanija tvrdi da ima bazu korisnika od 5 miliona MySQL instalacija u svetu i preko 10 miliona download-a proizvoda u 2007. godini.

Ovu kompaniju često nazivaju šampionom „druge generacije“ kompanija otvorenog koda. Prihodi kompanija otvorenog koda prve i druge generacije se ostvaruju prodajom podrške, konsultantskih usluga i treninga za njihove proizvode. Ono što generalno razlikuje ovu „druhu generaciju“, koju čine kompanije kao što je MySQL AB i Trolltech, od ranijih poslovnih modela otvorenog koda je dualno licenciranje – iako je softver dostupan i pod open source licencom, on je vlasništvo komercijalne kompanije koja želi da ostvari prihod prodajom tradicionalnih softverskih licenci.

Kao primer: MySQL je dostupan pod GPL-om besplatno, ali se prodaje pod drugim tradicionalnijim licencama klijentima koji ne misle da je GPL idealan za njihove poslovne namere, npr. uključivanje MySQL tehnologije u projekat zatvorenog koda.

Drugi izvori prihoda za MySQL AB su pružanje podrške i konsulting, kao i trening i sertifikacija za MySQL Server. Zajedno sa nekim dodatnim uslugama MySQL takodje prodaje MySQL Enterprise kao subskripcioni proizvod, koji preprodaju i druge kompanije (Dell, HP i Novell) sa kojima MySQL ima ugovor.

MySQL AB je od svog osnivanja 1995. rasla izuzetnim tempom da bi danas bila najpopularnija baza podataka otvorenog koda. Kompanija tvrdi da se sa njihovog sajta izvrši 50,000 download-ova dnevno, pa ne čudi zašto se ova kompanija smatra uspešnom i u svetu komercijalnog, i u svetu slobodnog softvera.

3.5. Prodaja oglasnog prostora na web portalima

Internet je promenio svet, a bez imalo sumnje, promenio je i softversku industriju iz korena. Neometano deljenje informacija, bolja komunikacija i pojednostavljenje većine svakodnevnih zadataka koje čovek obavlja su samo neke od funkcionalnosti koje je Internet omogućio. Softver kao usluga (software as a service, SaaS) je postao sve prisutniji model korišćenja softvera, pa mnogi progoniziraju smrt standardnog paketskog softvera i tvrde da će se uskoro sav softver koristiti na Internetu. Mnoge snažne i moćne softverske kompanije nisu izborile veliko tržište na Internetu. Ipak, postoje kompanije koje su kreirale nove poslovne modele i pronašle načine ostvarivanja ogromnih prihoda na ovom novom mediju. Najuspešnija kompanija u tom proboju je svakako Gugl, koji je svoje kvalitetne usluge (SaaS) spojio sa prodajom reklamnog prostora i time se za manje od 10 godina popeo na četvrto mesto najvećih softverskih kompanija.

3.5.1. Primer poslovanja po ovom modelu – Google

Google Inc. je američka javna korporacija koja ostvaruje prihod od advertajzinga na stranama svoje Internet pretrage, web baziranog e-maila, online mapa, web aplikacija kancelarijske produktivnosti, društvenog umrežavanja, deljenja video sadržaja, kao od i prodaje personalizovanih verzija istih tehnologija bez reklamnog sadržaja. Sedište Google-a je u Kaliforniji, SAD. Kompanija ima oko 20,000 zaposlenih, a predstavljaju i najveću američku kompaniju (po kapitalizaciji tržišta) koja nije deo Dau Džons industrijskog indeksa.

Gugl su osnovali Leri Pejdz i Sergej Brin dok su bili studenti Stanford univerziteta 1998. godine. Inicijalna javna ponuda je bila 2004. godine na NASDAQ berzi, čineći Gugl vrednim \$23 milijarde. Kompanija je nastavila rast kroz seriju razvoja novih proizvoda, akvizicija i partnerstava.

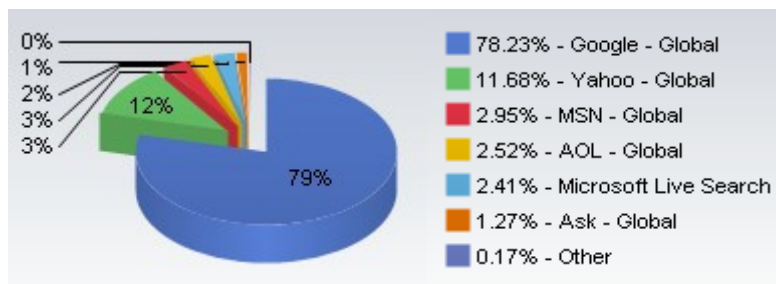
Prvi (a i danas najvažniji) proizvod ove kompanije, engine za pretragu Interneta, je brzo privukao veliki broj korisnika zbog svog jednostavnog dizajna i upotrebljivosti. U toku 2000, Gugl je počeo da prodaje reklame povezane sa ključnim rečima pretrage. Reklame su bile tekstualne, da bi se zadržao jednostavan dizajn stranice i da bi se maksimizirala brzina otvaranja stranice. Ključne reči su prodavane bazirano na kombinaciji izborne cene i broja klikova, gde je nadmetanje počinjalo sa 0.05\$ po kliku. Ovaj model reklamiranja pomoću ključnih reči je prvi razvio Goto.com, kompanija koju je kasnije otkupio Yahoo!. Dok mnogi dot-com rivali nisu uspeali na ovom novom Internet tržištu, Google se tiho izdigao generišući ogromne prihode.

Ime "Google" je nastalo od česte greške u pisanju reči "googol", koja označava 10^{100} , broj koji predstavlja jedinicu koju sledi 100 nula. Ova reč je postala deo svakodnevnog govora, označavajući servis kompanije Google za pretragu Interneta, ali i uopšte pretragu Interneta. Kao glagol je ušla u mnoge rečnike uz objašnjenje "koristiti Google engine za pretragu da bi pronašli informaciju na Internetu".

Prvi patent koji opisuje Google-ove mehanizme rangiranja (PageRank tehnologija) je odobren u septembru 2001. Nakon ovoga, kompaniji je odobreno još dosta patenata za njihove inovacije u softverskim tehnologijama.

Gugl je kreirao usluge i alate za generalnu javnost i poslovna okruženja, uključujući Web aplikacije, mreže za advertajzing i rešenja za kompanije.

Iako je osnovno poslovno interesovanje kompanije arena web sadržaja, Gugl je počeo da eksperimentiše i na drugim tržištima, kao što su radio i štampane publikacije. U januaru 2006, kompanija je objavila kupovinu radio advertajzing kompanije "dMarc" koja pruža automatizovani sistem koji omogućava kompanijama reklamiranje na radiju. Ovo je omogućilo kombinovanje 2 medija za reklamiranje, uz mogućnost fokusiranja na ukuse potrošača. Gugl je počeo i da eksperimentiše u prodaji reklama u novinama i magazinima.



Slika 5. Tržišni udeo Guglove Internet pretrage

Advertajzing

Većina Guglovih prihoda se ostvaruje iz advertajzing programa. Za 2007. fiskalnu godinu, kompanija je prijavila \$10.6 milijarde prihoda od advertajzinga i samo \$112 miliona od licenci i ostalih prihoda. Google AdWords omogućava oglašivačima na webu prikaz reklama u rezultatima pretraga i u Guglovoj mreži sadržaja (Google Content Network), kroz šemu cene po kliku ili po pogledu (cost-per-click i cost-per-view). Vlasnici Google AdSense web sajtova takodje mogu da prikazuju reklame na svom sajtu, i zaradjuju novac svaki put kada posetilac klikne na reklamu.

Web-bazirani softver

Guglov engine za web pretragu je najpopularniji servis kompanije. Već duže vreme je Gugl najkorišćeniji engine za pretragu weba sa 78.2% tržišnog udela, ispred Yahoo!-a (11.7%) i Majkrosoftovih pretraga (5.9%). Gugl indeksira milijarde web stranica, tako da korisnici mogu da pretražuju informacije koje žele, kroz upotrebu ključnih reči i operatora. Gugl je ugradio svoju tehnologiju pretrage weba u druge usluge pretrage, uključujući pretragu slika (Google Image Search), vesti (Google News), poredjenje cena (Google Product Search), interaktivne grupe (Google Groups), mape (Google Maps) itd.

U toku 2004, kompanija je pokrenula svoj besplatni web bazirani e-mail servis, poznat kao Gmail (ili Google Mail u nekim jurisdikcijama). Gmail uključuje tehnologije filtriranja spama odnosno nepoželjne pošte i pretrage e-maila. Servis generiše prihod prikazivanjem reklama i linkova sa AdWords servisa koji su vezani sa izborom korisnika i/ili sadržaja email poruka prikazanih na ekranu.

U 2006-oj godini, kompanija je pokrenula Google Video, koji ne samo da omogućava korisnicima pretragu i besplatno gledanje video sadržaja, već nudi korisnicima i medijskim izdavačima mogućnost objavljivanja sadržaja. Kasnije te godine, kompanija je odlučila da kupi sajt YouTube koji se bavi istom aktivnošću za \$1.65 milijarde. U februaru 2008, pokrenuli su Google Sites wiki kao deo Google Apps.

Gugl je takodje razvio nekoliko desktop aplikacija, uključujući Google Earth, interaktivni program mapiranja koji koristi satelitske i vazdušne snimke koji pokrivaju veći deo planete. Ova aplikacija se obično smatra izuzetno preciznom i detaljnom. Većina velikih gradova imaju tako detaljne snimke koji se mogu zumirati da se jasno vide pešaci i vozila. Jasno, bilo je nekih briga oko implikacija po nacionalnu sigurnost. Konkretno, neke zemlje i vojske su se žalile da softver može biti korišćen da se dovoljno tačno odredi fizička lokacija kritične infrastrukture, komercijalnih i rezidencijalnih zgrada, baza, vladinih agencijai sl. Ipak, satelitski snimci se ne update-uju često, i svi oni su dostupni besplatno i kroz druge proizvode, pa čak i iz vladinih izvora.

Mnogi drugi proizvodi su dostupni kroz Google Labs, koji su kolekcija nekompletnih aplikacija koja se još uvek testira za korišćenje generalne javnosti.

Takodje, u 2007. je objavljen Android, otvorena platforma za mobilne uređaje i skup alata za razvoj aplikacija na ovoj platformi. Ovo je prva platforma za mobilne uređaje nezavisna od proizvođača i predstavljala bi značajan pomak u ovom segmentu industrije.

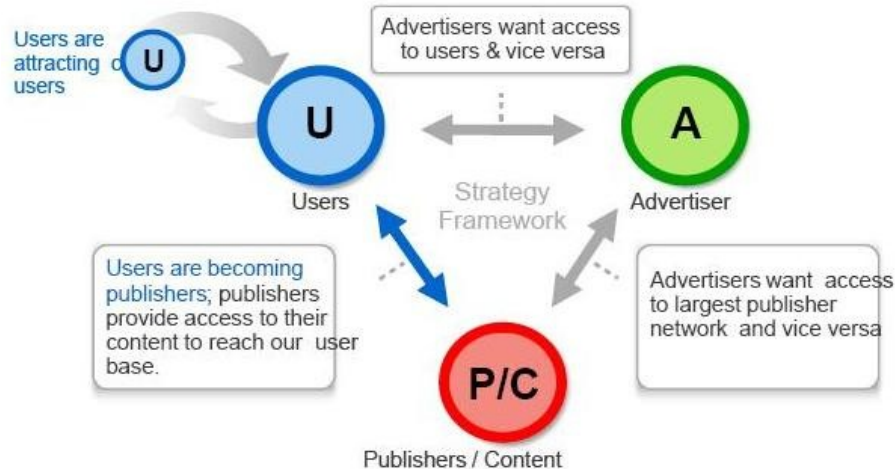
Enterprajz proizvodi

U toku 2007, Gugl je pokrenuo Google Apps Premier Edition, verziju Google Apps skupa aplikacija namenjenu primarno poslovnim korisnicima. Ona uključuje dodatke kao što su više prostora za email, API pristup i premium podrška, za cenu od \$50 po korisniku godišnje. Velika implementacija Google Apps sa 38,000 korisnika je na Lakehead univerzitetu u Kanadi.

Guglov ekosistem

Uspeh ove kompanije se može objasniti efektom mreže. Ovaj efekat označava činjenicu da dobro ili usluga imaju veću vrednost za potencijalnog korisnika zavisno od broja korisnika koji već poseduju to dobro ili koriste tu uslugu. Konkrento, to znači da je ukupna vrednost dobra ili usluge koja ima efekat mreže približno proporcionalna kvadratu broja korisnika koji već imaju to dobro ili koriste tu uslugu.

Korisnici koriste Google Search (ili kroz Google.com ili na nekom od partnerskih sajtova) zato što Gugl ima informacije; oni koriste druge proizvode ove kompanije zato što su korisni, moćni i fleksibilni. Provajderi informacija i sadržaja traže Gugl zato što Gugl ima mnogo korisnika, pa tako njihov indeks raste zato što imaju više korisnika. I oni koji žele da se reklamiraju se obraćaju ovoj kompaniji zato što ima mnogo korisnika, a onda kompanija dobija novi prostor za reklamiranje na partnerskim sajtovima zato što ima mnogo oglašivača. Na ovaj način Gugl ostvaruje sve veće prihode.



Slika 6. Guglov ekosistem

Kritike

Najviše kritika Gugl je primio po pitanju zaštite autorskih prava i privatnosti.

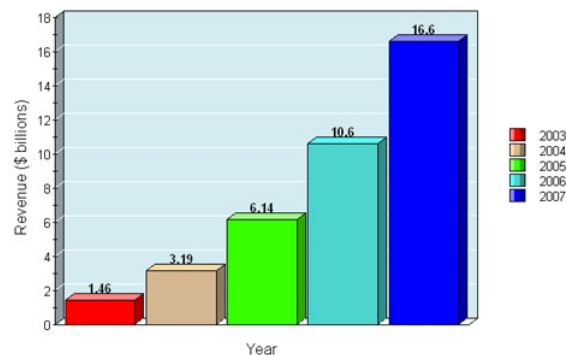
Neke organizacije su koristile Digital Millenium Copyright Act da zahtevaju od Gugla da ukloni reference ka materijalu zaštićenom autorskim pravima na drugim sajtovima. Gugl obično ovo rešava tako što ukloni linkove kao što je traženo i prikaže obaveštenje o uklonjenim stavkama u

rezultatima pretrage. Bilo je žalbi i sa keširanjem, u smislu da ono krši autorska prava. Ipak, Gugl pruža mehanizam zahteva za isključivanje keširanja. Naravno, žalbe oko kršenja copyright-a postoje i na Guglovom sajtu YouTube, ali je proces uklanjanja sadržaja po žalbi automatizovan i dobro funkcioniše.

Druga osnovna kritika se tiče opasnosti i implikacija po privatnost činjenice da postoji centralizovano široko popularno skladište podataka miliona pretraga Internet korisnika, i da li Google po zakonu može/treba da preda te podatke vladi SAD (ili vladi druge jurisdikcije). Takođe, neki korisnici misle da procesiranje email poruka u Gmail-u krši privatnost. Ipak, kompanija tvrdi da ovaj sadržaj nikad ne čita nijedno ljudsko biće i da se on koristi samo u cilju popravljavanja relevantnosti reklama. Šta više, niko ne može da tvrdi da to ne rade i ostali email provajderi. Sličnih žalbi je bilo za Google Streets i još neke servise ove kompanije.

Ipak, kompanije generalno ima dobar imidž, jer je veliki podržavaoc zaštite životne sredine i ulaže velika materijalna sredstva u dobrotvorne svrhe. Takođe, Gugl podržava razvoj softvera otvorenog koda time što je i sama otvorila deo svojih baza, i što finansira razvoj nekih konkretnih open source projekata. Svojom kompanijskim sloganom "Don't be evil" („Ne budite zli“), oni tvrde da se mogu ostvarivati prihodi i na sveopšte dobro.

Kao jedna od kompanija koje su nastale, uzdigne se i postoje zahvaljujući Internetu, ova firma unosi inovacije i ulaže napore da informacija bude univerzalno pristupačna i dostupna. Pronašli su veoma dobar način da ovu svoju ideju i unovče, kako možemo videti na Slici 7.



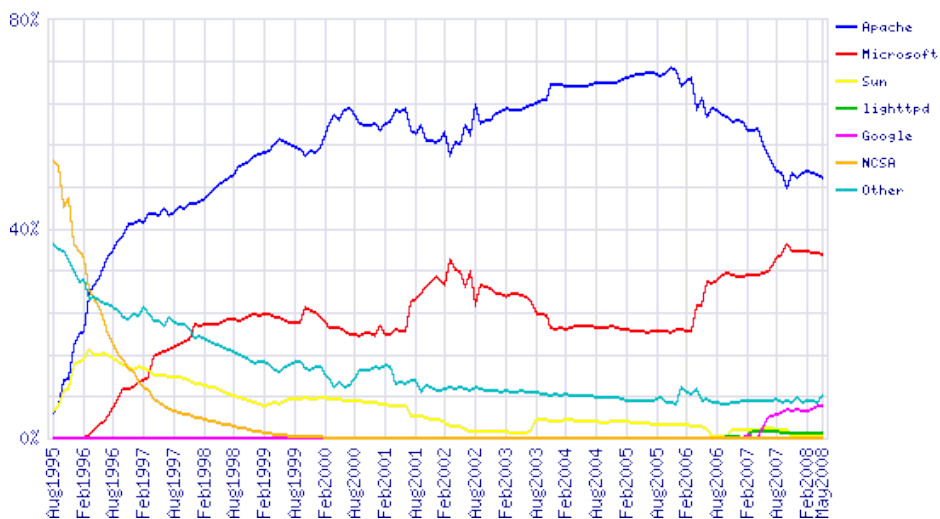
Slika 7. Rast prihoda kompanije Gugl u poslednjih 5 godina

3.6. Održivi nekomercijalni softverski proizvod

I pored svih ideoloških pristalica razvoja softvera otvorenog koda, teško je poverovati da uopšte može da se razvije masivna softverska aplikacija bez podrške neke kompanije ili organizacije koja bi to radila iz interesa. Još je teže poverovati da takva aplikacija može uspešno da postoji duže vreme i osvoji dominantan udeo u tržištu. Ipak, ovakvi slučajevi su se desili par puta u kratkoj istoriji softverske industrije, gde su najbolji Apache, Mozilla i Linux kernel (iako ove 3 aplikacije imaju u mnogome različitu priču). Ja ću opisati uslove u kojima je nastao i razloge zašto se održao Apache web server.

3.6.1. Primer poslovanja po ovom modelu – Apache

Apache je posebno zanimljiv slučaj razvoja softvera. Apache je razvijen tako što su ljudi periodično dodavali zakrpe, i on se razvio u funkcionalan, kvalitetan i izuzetno popularan proizvod. Ima široku bazu korporativne podrške, uključujući masivnu podršku IBM-a. Da bi utvrdili zbog čega je Apache tako uspešan, osvrnućemo se kratko na razvoj ovog servera, izbor licence i strukturu grupe koja nadgleda razvoj ovaj softver – Apache Software Foundation. Važno je naglasiti da se termin „Apache“ koristi u više konteksta – kao organizacija koja nadgleda Apache web server, kao sam server i kao ostale komponente oko servera. Ovde će se pod terminom Apache podrazumevati web server.



Slika 8. Tržišni udeo Apache-a medju web serverima

Razvoj Apache HTTP Servera

Jedan od prvih razvijenih web servera ikad je bio NCSA HTTP Server koji je razvijen u Nacionalnom Centru Superkompjuterskih Aplikacija (NCSA) na Univerzitetu Illinois. Server je bio pušten pod licencom javnog domena koja je omogućavala korisnicima da sa kodom rade šta god žele. Ovaj server je odmah postao najpopularniji web server. Kada je jedan od glavnih programera napustio centar 1994, rad na projektu je stagnirao, i uskoro je odredjen broj korisnika počeo da dodaje svoje zakrpe i da ih širi svima kao da više nema centralne glave projekta. Uskoro je jedan broj ovih ljudi rešio da kreira svoju distribuciju servera sa zakrpama (patchevima) koje su razvili. Odlučili su da je nazovu Apache jer je bio „a patchy server“ a i iz poštovanja za Apače, indijansku naciju koja je imala reputaciju kao veoma vešta i istrajna. Za

manje od godinu dana kako je grupa formirana, Apache je postao najpopularniji web server. On se sada izvršava na 67% web sajtova, a na drugom mestu je Majkrosoftov server sa samo 25%.

Odgovor na pitanje zašto je Apache odmah dobio tako veliki udeo u tržištu je veoma jednostavan – on je tada bio jedini pristojan web server sa obećavajućom budućnošću. Komercijalne kompanije u to vreme nisu ulagale u web servere jer nisu imale poverenja u njegovu isplativost. Početna prednost nikad nije prevaziđena, ne samo zbog halo efekta ili ograničavanja korisnika, već zato što je Apache i danas veoma kvalitetan web server, koji razvija hiljade ljudi a koriste milioni. Uz to, on je potpuno besplatan i slobodan.

Izbor licence

Apache je trenutno dostupan pod šemom licenciranja sličnom BSD licenci. Ovo znači da kod Apache-a može biti integrisan u komercijalni proizvod, a da kod promena ne mora da bude podeljen sa zajednicom. Mnoge firme zaista i prodaju komercijalne web servere bazirane na Apache-u, izmedju ostalog IBM i Covalent. Ove firme su takodje doprinele Apache projektu i objavile dosta koda koji su razvile za svoje servere nazad u zajednicu. Interes firmi da rade ovo je više puta objašnjen u radu, ali ukratko, kompanije kao IBM imaju interes da Apache bude najbolji mogući proizvod, jer će tada više firmi želeti da ima rešenja bazirana na njemu, izmedju ostalog i ona koja IBM nudi kroz svoj konsalting i softverske pakete koji dodaju vrednost.

Struktura grupe

Struktura Apache grupe se razlikuje od većine softverskih projekata otvorenog koda. Mnoge OSS projekte nadgleda jedan "diktator" ili individua koja ima uvid u ceo projekat duži niz godina. Linux je verovatno najbolji primer ovoga – Linus Torvalds lično nadgleda šta se uključuje u Linux kernel. Pečevi kernela se unose kroz proces koji zahteva da oni prodju kroz više nivoa proveru, dok sam Torvalds ne dozvoli uključivanje.

Organizacija Apache-a je više imperijalni senat nego vladavina diktatora. Inicijalno je grupa od osmoro skupila svu dokumentaciju i popravke bagova i izdala tako popravljenu verziju. Ova grupa sad glasa šta će biti, a šta neće biti uključeno u finalne pečeve Apache-a koji se zvanično puštaju u javnost. Pored ove osmočlane grupe, postoji mnogo više članova šire Apache Software Foundation organizacije. Članstvo se sastoji od ljudi koji odlučuju o http server projektu, na koje još projekte organizacija treba da se fokusira, kao i šta raditi sa prikupljenim novcem organizacije. Kako je organizacija rasla, od originalnog servera ona se proširila i na druge projekte. Ovo uključuje SSL podršku (secure socket layer) koja omogućava sigurne veze za osetljive transakcije, kao i mod_perl projekat, koji omogućava da se Perl skripte izvršavaju na Apache serverima zbog dodatnih funkcionalnosti.

U svakom slučaju, Apache je dokazao da i nekomercijalni proizvodi koje razvija zajednica otvorenog koda mogu biti veoma ozbiljni. Ipak, ostaje činjenica da se na ovaj način mogu razvijati samo infrastrukturni proizvodi neophodni velikom broju korisnika i organizacija.

4. Zaključak

U ovom radu sam opisala poslovne modele trenutno aktuelne u softverskoj industriji. Najpre je postavljen okvir za priču o poslovnim modelima opisom okruženja i faktora koji utiču na razvoj poslovanja u ovoj oblasti.

Ukratko je bila izložena istorija ove mlade industrije, kroz opis njenih ključnih segmenata od početaka u 1950-im godinama pa do danas. Zatim je bila opisana istorija pokreta za razvoj softvera otvorenog koda, koji su imali uticaj na stanje industrije danas. Takođe, u ovom delu je opisana i pravna regulativa u ovoj oblasti, jer ona postavlja okvir funkcionisanja. Konkretno, objašnjene su softverske licence kojima se štiti intelektualna svojina, kao i kontroverzno pitanje softverskih patenata oko koga još uvek nema saglasnosti u zajednici.

U drugom delu rada su izložene ekonomske paradigme razvoja softvera. Ovde je razvoj posmatran sa strane korisnika softvera, kako profesionalnog (odnosno korporativnog) tako i kućnog, i razmatrani se svi aspekti razvoja iz ove perspektive – troškovi i rizik razvoja, efikasnost finansiranja, mogućnost diferencijacije u odnosu na konkurentne proizvode i stepen u kom drugi mogu biti ograničeni u korišćenju softvera. Detaljno se opisane 4 osnovne ekonomske paradigme razvoja softvera.

U trećem delu rada su nabrojani i opisani poslovni modeli u softverskoj industriji, sa strane proizvođača softvera. Svaki model je opisan na primeru neke poznate kompanije koja proizvodi softver da bi ostvarivala prihod. Vredi naglasiti da kao primer nisu korišćene samo kompanije kojima je osnovna delatnost razvoj softvera, već mnoge od njih pružaju usluge (između ostalog usluge konsaltinga, treninga, sertifikacije, podrške) ili koriste softver za unapredjenje prodaje (računarske opreme, oglasnog prostora itd.). Postoji veliki broj modela, a ovde je bilo nabrojano desetak, i detaljno opisano nekoliko njih.

Nema sumnje da će se u budućnosti stvarati novi poslovni modeli u softverskoj industriji. Oni će pratiti razvoj i inovaciju u oblasti računarstva, ali biće i puno novih originalnih okvira za zaradu ulaganjem u razvoj softvera.

5. Literatura

- [1] *The Emerging Economic Paradigm of Open Source*, Bruce Perens, 2005.
- [2] *The Magic Cauldron*, Eric S. Raymond, 2002.
- [3] *Producing open source software*, Karl Fogel, 2005.
- [4] *The business of software*, Michael Cusumano, 2004.
- [5] *An imperical look at software patents*, Bessen & Hunt, 2004.
- [6] *Open source software development case studies*, Robert L. Greenberg, 2003.
- [7] Software & Information Industry Association, www.siaa.net
- [8] http://en.wikipedia.org/wiki/Software_patent
- [9] <http://www.answers.com/topic/software-industry>
- [10] <http://www.microsoft.com>
- [11] <http://www.ibm.com>
- [12] <http://www.redhat.com>
- [13] <http://www.google.com/corporate>
- [14] <http://www.mysql.com>