

**UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA**

DIPLOMSKO DELO

**PRIMERJAVA ODPRTO-KODNEGA IN LASTNIŠKEGA PROGRAMJA S
PODROČJA POSLOVNE INTELIGENCE**

Ljubljana, oktober 2007

LUKA KURALT

IZJAVA

Študent Luka Kuralt izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom prof. dr. Jurija Jakliča in da dovolim objavo diplomskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne: 01. 10. 2007

Podpis:

KAZALO

1	Uvod	1
2	Odperta koda	2
2.1	Prosto programje	3
2.2	Odperto kodno programje	3
2.3	Primerjava prostega in odperto-kodnega programja	4
2.4	Prikaz delovanja odperto-kodnih projektov	4
2.5	Zgodovina	6
3	Licenciranje	7
3.1	Odperto-kodne licence	7
3.2	Lastniške licence	8
3.3	Povezava med licenciranjem in dostopom do programja	9
4	Poslovna inteligenca	9
4.1	Definicija	10
4.2	Arhitektura poslovne inteligence	12
4.3	Zgodovina	15
5	Pregled programja poslovne inteligence	16
5.1	Odperto-kodno programje	16
5.1	Lastniško programje	19
5.2	Primerjava programja	21
6	Kriteriji za izbiro programja poslovne inteligence	25
6.1	Strateška merila	28
6.2	Tehnična merila	28
7	Izbira programja poslovne inteligence	29
7.1	Način ocenjevanja	30
7.2	Primerjava po strateških merilih	30
7.3	Rezultat analize	31
7.4	Primerjava po tehničnih merilih	33
7.5	Rezultat analize	35
8	Sklep	37
	Literatura	39
	Viri	40
	Priloga	1

KAZALO SLIK

Slika 1: Delovanje odprto-kodnih projektov	5
Slika 2: Načini licenciranja programja	9
Slika 3: Piramida spremembe od podatkov do znanja	10
Slika 4: Prikaz merjenja učinkovitosti delovanja procesa	11
Slika 5: Arhitektura poslovne inteligence	13
Slika 6: Strateška merila za izbiro programja	26
Slika 7: Tehnična merila za izbiro programja	27
Slika 8: Dejavniki, ki vplivajo na uspešnost spremembe organizacije	27

KAZALO TABEL

Tabela 1: Primerjava programja s področja poslovne inteligence po strateških merilih	31
Tabela 2: Primerjava programja s področja poslovne inteligence po tehničnih merilih	34

1 Uvod

Globalizacija in posledično vse večja konkurenca sili podjetja, da se bolje organizirajo, ter se nato tudi prilagajajo spremembam. Pri organizaciji pa običajno veliko vlogo igrajo tudi službe informatike v posameznih podjetjih, z informatizacijo dejavnosti in procesov. Poleg tega pa se informatiki v podjetjih ukvarjajo tudi z dostavljanjem informacij odločiteljem na posameznih dejavnostih. S ciljem, da imajo odločitelji na voljo kvalitetno informacijo, v trenutku, ko jo potrebujejo. Po drugi strani pa službe informatike velikokrat ustvarjajo velike stroške s svojimi investicijami.

Vodje služb za informatiko so se posledično začeli ozirati za cenejšimi možnostmi informatizacije poslovanja podjetja. V svoje povpraševanje po informacijskih tehnologijah so začeli uvrščati tudi odprto kodno programje, ki naj bi bilo na prvi pogled precej cenejše. Vprašanje, ki pa se tu takoj postavi, pa je, ali je odprto kodno programje dovolj kvalitetno, da podpre poslovne procese podjetja. Postavi se tudi vprašanje, kakšno podporo nam nudi ponudnik odprto-kodnega programja. Še posebej bo zanimivo ugotoviti, kakšne so možnosti ponudbe odprto-kodnega programja na precej ozko specializiranih in hitro razvijajočih se panogah, kakršna je panoga poslovne inteligence.

Namen diplomske naloge je predstaviti odprto kodno programje s področja poslovne inteligence ter ga nato tudi primerjati z vodilnimi ponudniki lastniškega programja s področja poslovne inteligence. Izraz programje se v nadaljevanju nanaša na programje s področja poslovne inteligence. Če pa se izraz ne bo nanašal na navedeno programje, bo to tudi napisano.

Diplomsko nalogo bom razdelil na dva glavna dela: teoretični in praktični del. V teoretičnem delu diplomske naloge bom razložil odprto kodo (*ang. open source*) ter njen razvoj skozi čas. Nato bom predstavil poslovno inteligenco¹ (*ang. business intelligence*) ter poglede nanjo s strani različnih avtorjev. Na kratko bom predstavil tudi zgodovino poslovne inteligence. V drugem, praktičnem delu pa bom prikazal primerjavo med odprto-kodnim in komercialnim programjem s področja poslovne inteligence na konkretnem primeru. Tako bom najprej izbral in predstavil merila, s katerimi bom ocenjeval programje. Nato bom izbral ponudnike programja in naredil tudi njihovo predstavitev. V zadnjem delu sledi še primerjava izbranega odprto-kodnega in lastniškega programja na podlagi izbranih meril.

Cilj je ugotoviti, ali je odprto kodno programje s področja poslovne inteligence primerljivo z istovrstnim lastniškim programjem. Poskusil bom tudi ugotoviti, na katerih področjih lahko odprto kodno programje konkurira lastniškemu programju.

¹ V slovenski literaturi najdemo tudi izraz poslovno obveščanje

Navedeno problematiko sem preučeval predvsem na analitičen način. Po posameznih poglavjih sem v večini uporabljal opisno metodo preučevanja. V praktičnem delu pa sem večinoma uporabljal primerjalno metodo, kar je tudi cilj te diplomske naloge.

Ker je literature v knjižni obliki za to področje malo, sem večino podatkov pridobival z interneta. Oziral sem se za uglednimi revijami, neodvisnimi raziskavami in članki avtorjev s tega področja. Izogibal pa sem se raziskavam, ki jih predstavljajo ponudniki sami ali pa so financirane z njihove strani. Uporabil pa sem tudi znanja, ki sem jih pridobil pri študiju informatike, na konferencah in pri dosedanjem delu v praksi.

2 Odprta koda

Odprtakoda.org opiše odprto kodo na naslednji način (Odprta koda, 2007): "Odprta koda ali "open source" pomeni programje, ki podpira skupinski razvoj le-tega, saj je izvorna koda na voljo javnosti in si jo lahko vsak prilagaja svojim potrebam." Razloženo je, da izraz "prosto" v odprti kodi pomeni "prosto", kot pri pravici izražanja mnenja ("free as in free speech") in ne zastonj, kot denimo, zastonj pijača. Toda odprto ali prosto (da bolje razumemo to trditev, povejmo še, da v angleščini beseda "free" pomeni hkrati "prosto" in "zastonj") programje se običajno dobi brezplačno, zato je večina narobe razumela cilj odprte kode kot zelo ne-komercialnega.

Razlaga, ki jo najdemo v Wikipediji (2007) pa pravi, da programje, ki je na voljo s svojo kodo pod odprto kodno licenco, dovoljuje vsem uporabnikom preštudirati, menjati, izboljševati in razpošiljati odprto kodno programje. Edina dejavnika, ki nam povesta, ali je določeno programje odprto kodno ali ne, sta dostopnost kode in licenčni pogoji za razpošiljanje programja. Trenutno se odprta koda deli na dve veji: prosto programje (*ang. free software*) in odprto kodno programje (*ang. open source software*).

Ker je običajno koncept odprte kode težje razumljiv, še posebno pa za ljudi, ki niso računalniški zanesenjaki, bom idejo odprte kode prikazal na primeru piva. Recept za pivo, imenovan Vores Øl (*slo. naše pivo*), najdemo ga tudi pod imenom Free Beer (*slo. zastonj, prosto pivo*), so iznašli študentje Kopenhagenske Univerze za informacijsko tehnologijo, z namenom dokazovanja, da se lahko zasnova odprte kode uporablja tudi zunaj digitalnega sveta. Recept za izdelavo piva je dostopen vsakomur preko spleta, tako, da si lahko vsakdo, ki ima željo, zvari to pivo sam. Vendar je tu potrebno dodati, da zvarjeno pivo, ki ga že izdeluje nekaj manjših pivovarn po svetu ni zastonj in je v redni prodaji kot vsa druga "komercialna" piva. Seveda, če pa želite, lahko naredite pivo po receptu ter ga prosto ali zastonj razpošiljate naprej. Omenjeno pivo pa lahko kupite tudi preko njihove spletne strani. Prvih sto litrov piva, ki so ga zvarili danski študentje, so poimenovali inačica 1.0. V času, ko pišem diplomsko nalogo, pa so že prišli do inačice 3.0.

2.1 Prosto programje

Free Software Foundation (2006) je definirala prosto programje kot programje, ki uporabnikom omogoča, da ga uporabljajo, kopirajo, razpošiljajo, preučujejo, spreminjajo in izboljšujejo. Poznamo štiri "svobode" za uporabnike programja (Free Software Foundation, 2006):

1. Svoboda uporabe programja za katerikoli razlog.
2. Svoboda preučevanja delovanja programja in njegova prilagoditev za lastne potrebe.
3. Svoboda razpošiljanja programja z razlogom pomagati drugim.
4. Svoboda do izboljšanja programja in predstavitev lastnih izboljšav javnosti, z razlogom, da ima koristi od tega celotna skupnost.

Če ima programje vse štiri naštetosti, ga označimo kot prosto programje. S to definicijo pa Free Software Foundation skrbi, da si nihče ne lasti izvirne kode.

2.2 Odprto kodno programje

Izraz odprta koda, ki se uporablja bolj pogosto, se je pojavil veliko kasneje, ko se je skupina ljudi iz Free Software Foundation ločila od zgoraj opisane originalne zasnove. Začetki so pogosto povezani z odprto kodno skupnostjo, ki pravi, da je odprto kodno programje vsakršno programje, ki se drži naslednjih desetih načel (Open Source Initiative, 2006):

1. Prosta prerazdelitev: Licenca ne sme omejevati nikogar pri prodaji ali brezplačnem razpošiljanju programja kot komponente skupnega razpošiljanja programja, ki je sestavljen iz več programov. Za takšno vrsto posredovanja se ne sme zahtevati nikakršnega plačila programja.
2. Izvirna koda: Programje mora vsebovati izvorno kodo in mora dovoljevati razpošiljanje tako v izvorni kodi, kakor tudi v obliki sestavljenih obrazcev. Tam kjer katerakoli oblika proizvoda ni razpošiljana z izvorno kodo, morajo obstajati dobro opisani razlogi za zadržanje izvirne kode. Ta pa se ne sme zadržati za več kot razumno reprodukcijo, ki je stroškovno sprejemljiva. Snemanje (*ang. download*) z interneta pa mora ostati brezplačno. Izvirna koda mora biti v sprejemljivi obliki tako, da lahko programer prilagodi programje. Namerno oteževanje razumevanja kode ni dovoljeno. Vmesne oblike, kot npr.: prevajalnik, niso dovoljene.
3. Izpeljana dela: Licenca mora dovoljevati modifikacije in izpeljana dela na programju. Prav tako mora dovoljevati nadaljnjo razpošiljanje pod enakimi pogoji kot to velja za original.
4. Poštenost avtorjeve izvirne kode: Licenca lahko omeji razpošiljanje v spremenjeni obliki samo, če licenca istočasno dovoljuje razpošiljanje datotek za nadgradnjo, skupaj z izvorno kodo, z namenom spreminjanja programja v času izgradnje. Licenca mora

jasno dovoliti razpošiljanje programja, zgrajenega iz spremenjene izvorne kode. Licenca lahko dovoli drugo ime ali številko innačice v primerjavi z izvornim programjem.

5. Ne sme se zapostavljati posameznikov ali skupin: Licenca ne sme zapostavljati posameznikov ali skupin.
6. Ne sme se zapostavljati področij uporabe: Licenca ne sme omejevati nikogar pri uporabi programja, ne glede na področje uporabe. Npr., licenca ne sme omejevati uporabo istega programja v poslovne namene ali za genetske raziskave.
7. Razdeljevanje licence: Pravice, ki pripadajo programju, morajo veljati za vse tiste, katerim je bilo programje posredovano, brez potrebe po dodatni licenci za omenjene osebe.
8. Licenca se ne sme prilagajati programju: Pravice, ki pripadajo programju, ne smejo biti odvisne od tega, da je program del določenega programja, ki se razpošilja skupaj. Če je program izvzet iz tega razpošiljanja in uporabljan ali razpošiljan v skladu z licenco tega programja, morajo vsi, katerim je bil program posredovan, imeti enake pravice kot tisti, ki jim je to omogočeno skupaj z originalnim programjem.
9. Licenca ne sme omejevati drugega programja: Licenca ne sme omejevati drugega programja, ki se razpošilja skupaj z licenciranim programjem. Npr., licenca ne sme postavljati za pogoj, da je vso drugo programje, ki se istočasno distribuira na istem mediju, odprto kodno.
10. Licenca mora biti tehnološko nevtralna: Noben člen licence ne sme omejevati procedure kode na katerikoli individualni tehnologiji ali obliki vmesnika.

2.3 Primerjava prostega in odprto-kodnega programja

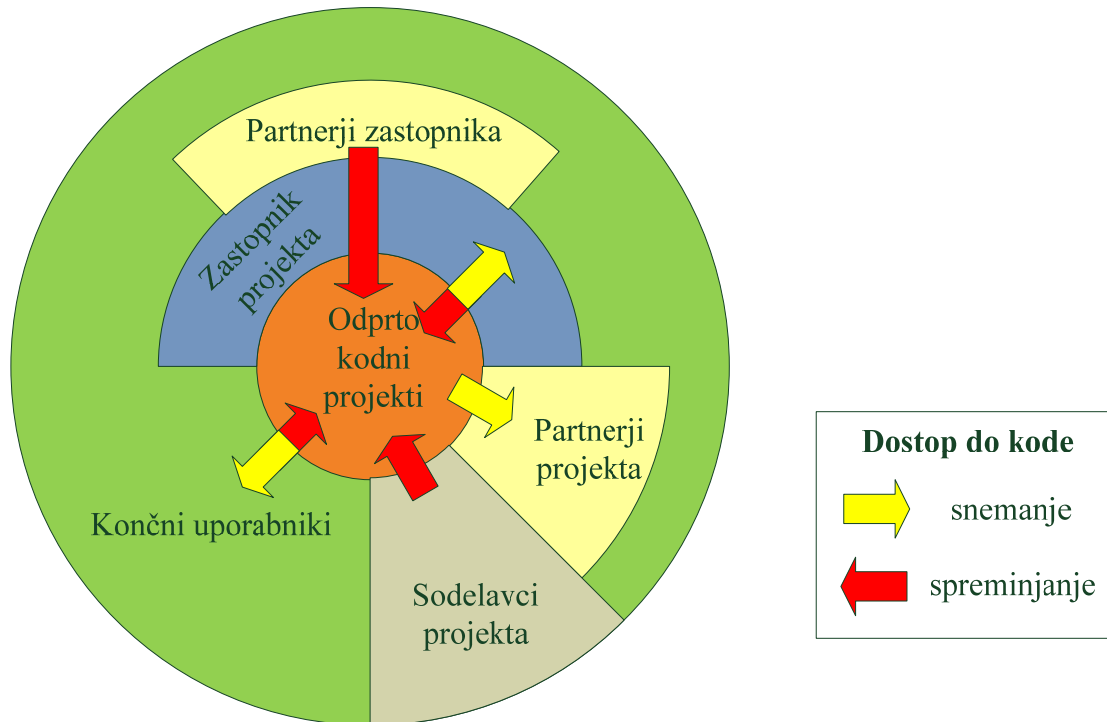
Kljub temu, da sta si zgoraj navedeni definiciji, prostega in odprto-kodnega programja na prvi pogled podobni, se vseeno razlikujeta med seboj. Zagovorniki odprto-kodnega programja slonijo na ideji, da pravice intelektualne lastnine ovirajo tehnični napredek, delajo proti koristi skupnosti ter da etičnost ali njeno pomanjkanje obstaja samo v povezavi z razpošiljanjem programja. Na drugi strani pa gibanje prostega programja gradi na temelju ekonomskih in tehničnih odlik, raje kot na moralnih in etičnih načelih, kot to počne gibanje odprto-kodnega programja.

2.4 Prikaz delovanja odprto-kodnih projektov

Slika 1 prikazuje način delovanja odprto-kodnih projektov oziroma dostop do izvorne kode projekta različnim skupinam. Puščice kažejo, kakšne pravice uporabe projekta imajo različne skupine. Če je puščica obrnjena proti projektu, pomeni to, da lahko določena skupina spreminja izvorno kodo projekta. Če je puščica obrnjena stran od projekta, pomeni, da lahko

določena skupina le snema (*ang. download*) projekt z interneta. Obojestransko obrnjena puščica pa skupini dovoljuje uporabo obeh pravic.

Slika 1: Delovanje odprto-kodnih projektov



Vir: Open Source Software Ecosystem, 2007.

Končni uporabniki, ki so zasebni, javni, neprofitni uporabniki in podjetja, lahko snemajo projekt z interneta ter tudi spreminjajo izvorno kodo.

Zastopnik projekta je podjetje, ki uradno zastopa projekt. Predvsem je to običajno za odprto-kodne projekte, ki so namenjeni podjetjem ali kakšnim drugim podobnim organizacijam. To podjetje nato skrbi za izdajo različic, ima nalogo povezovanja skupnosti, trženje in promocijo projekta in še nekatere druge naloge. Dokaj pogosto pa je tudi, da je to podjetje nosilec razvoja odprto-kodnega projekta.

Partnerji zastopnika so druga podjetja ali posamezniki, ki delujejo kot podporna mreža za uvajanje in vzdrževanje projekta. Za svoje delovanje pa so pridobili licenco s strani zastopnika projekta.

Sodelavci projekta so osebe, ki sodelujejo pri razvoju in drugih dejavnostih pri projektu. Če je zastopnik projekta podjetje, in je to podjetje nosilec razvoja, spada nekaj teh sodelavcev pod okrilje zastopnika projekta. Sodelavci projekta pa so lahko tudi drugi ljudje, ki delujejo preko skupnosti.

Partnerji projekta pa so lahko podjetja ali posamezniki, ki projektu nudijo komponente za njegovo delovanje, in jih ta tudi vključuje. To je zelo pogosta praksa v odprto-kodni skupnosti. Projekti tako velikokrat vključujejo druge odprto-kodne projekte, ki so že narejeni in jih uporabljajo kot sestavine za svoje delovanje.

2.5 Zgodovina

Na začetku je obstajalo samo prosto programje. Zasnova razvoja fizične kode programja, ki bi lahko tekla na računalniku in bi jo lahko replicirali, se je začela istočasno v različnih znanstvenih laboratorijih v poznih 60-ih letih prejšnjega stoletja, po ZDA (Stanford, Berkeley, Carnegie Mellon in MIT). Laboratoriji so bili med seboj v tesnih stikih in so med seboj delili rezultate raziskav, publikacije ter tudi organizirali strokovne posvete. Programiranje v tistih časih je bilo zelo sorodno drugim znanstvenim disciplinam v univerzitetnem okolju. Ekonomska upravičenost razvoja kode je bila zanemarljiva, saj je bilo v tistem času majhno število računalnikov, na katerih bi se ta koda lahko izvajala. Pa še ti računalniki so se v glavnem nahajali v raznih inštitutih in niso imeli velike povezave s poslovnim svetom. Majhni napredki v učinkovitosti in funkcionalnosti so se hitro replicirali po drugih sodelujočih laboratorijih, saj je bila skupnost majhna in tesno povezana. Ko je programer uspel narediti izboljšavo na operacijskem sistemu, je kodo hitro posredoval celotni skupnosti. S tem je dobil priznanje skupnosti in je veljal za "očeta" tega dela skupne kode. Lastništvo kode pa je bila v tistem času samo formalnost, saj nihče ni zahteval pravic za svoj prispevek. Vsi v skupnosti so vedeli, kdo je izdelal kateri del kode (Barausse, Jermol, Stevens, 2006, str. 11).

V tistem času se je zgodil še en izjemen del razvoja odprte kode. Danes je to ogrodje GNU operacijskega sistema, in sicer gre za Linux kernel, ki ga je napisal študent na Univerzi v Helsinkih, po imenu Linus Torvalds. Torvalds je napisal kernel, podoben Unixu, ki je imel osnovo v Minix-u, manjšem klonu Unixa, ki se uporablja kot orodje za učenje. Torvalds je poslal svoj kernel, imenovan Linux (Linus + Unix) v pregled različnim skupinam in preko e-poštnih seznamov. Kar nekaj drugih programerjev je začelo spreminjati in dopolnjevati kodo. To kodo so sproti pošiljali nazaj Torvaldsu, z namenom, da jo vključi v naslednjo inačico kernel-a. Sčasoma je Linux postal de facto kernel za GNU operacijski sistem (Barausse, Jermol, Stevens, 2006, str. 12).

Leta 1997 je Eric Raymond objavil esej z naslovom "The Cathedral and The Bazaar". V njem je pojasnil razloge, zakaj je po njegovem mnenju programje pod odprto kodno licenco kvalitetnejše in cenejše. Esaj se je hitro razširil po programerski skupnosti. Vsi so se strinjali, da se mora odprto-kodna skupnost bolje odzvati na komercialne konkurente, če se želi širiti in obnavljati "odprtega" duha. Ugotovili so, da se morajo bolj usmeriti na potrebe industrije in nekoliko bolj zanemariti ideološke cilje, ki so do sedaj prevladovali v odprto-kodnem gibanju. Zagotoviti morajo neke vrste certificiranje, ki bi potrdilo aplikacije, ki se izdajajo za odprto-

kodne, resnično za odprto-kodne. Na sestanku "delničarjev", februarja 1998, so se dogovorili o pogojih odprte kode, izdelali poslanstvo in ustanovili Open Source Initiative. Open Source Initiative je neprofitna organizacija, katere poslanstvo je ustvariti "odprte" licence, certificirati izvorno kodo in zagotoviti informacije v zvezi z odprto-kodno programsko opremo. Po objavi Netscapa in ustanovitvi Open Source Initiative, je še nekaj podjetij oznanilo podporo Linuxu, med njimi tudi Oracle, IBM in Corel. Intel in Netscape pa sta ustanovila podjetje za distribucijo Linuxa, z imenom Red Hat (Barausse, Jermol, Stevens, 2006, str. 13).

3 Licenciranje

Licence, ki jih sprejmemo z namestitvijo izbranega programja, so lahko zelo različne, načeloma pa se ločijo na dve večji skupini: licence za lastniško in licence za odprto-kodno programje.

Za programje pod lastniško licenco je značilno, da uporabnik prejme le izvedljivo inačico programja, medtem ko je za programje pod odprto-kodno licenco značilno, da uporabnik prejme tudi izvorno kodo (Jerič, 2004, str. 19). Ker so razlike v licencah najbolj očitna in verjetno tudi največja razlika med odprto-kodnim in lastniškim programjem, bom pojasnil najpogosteje uporabljane licence iz obeh navedenih skupin.

Podobnost med obema skupinama licenc je ta, da ko namestimo programje, sprejmemo pogoje uporabe tega programja, ne glede na to ali je programje lastniško ali odprto kodno. Obe imata licenco. Če pa se pogojev ne držimo, lahko izgubimo pravico do uporabe programja ali pa se celo znajdemo na sodišču.

Razlika med obema skupinama licenc pa je v tem, da v licenci za lastniško programje piše, česa vse ne smemo delati, v licenci za odprto-kodno programje pa piše, da lahko delamo vse kar želimo, razen stvari, ki so določene pogoji (Pingo Linux, 2004).

Za izdajo odprto-kodnih licenc določa pogoje Open Source Initiative – OSI (*slo. Odprto-kodna iniciativa*). Pogoje, ki jih določa OSI, sem navedel v poglavju odprto-kodno programje, kjer so tudi razloženi.

3.1 Odprto-kodne licence

Najbolj razširjeni in edini odprto-kodni licenci, ki ju uporabljajo ponudniki poslovne inteligence sta:

- GNU GPL (General Public License),
- GNU LGPL (Library or "Lesser" General Public License),

Obstaja še precej drugih podobnih odprto-kodnih licenc, vendar se zaradi vsebine diplomske naloge ne bom spuščal v njihovo razlago. Uradna razlaga in licence so prosto dostopne preko spletnih strani krovnih organizacij FSF in OSI.

GNU (Gnu's Not Unix) se je imenoval operacijski sistem, ki ga je napisal Richard Stallman, tako pa ga je poimenoval, da bi ga ločil od drugih plačljivih različic. Da bi zagotovil, da bi bilo GNU kodo vedno možno brezplačno spreminjati in razpošiljati, je Stallman ustvaril licenco, ki je to omogočala. Licenca se je imenovala GNU General Public License (GPL), ter se še danes imenuje enako. GPL licenca s svojimi pogoji omogoča, da lahko uporabnik pregleduje, spreminja ali dodaja kodo, pod pogojem, da so spremembe v kodi dostopne drugim pod pogoji iste licence GPL. Za razvoj sistema GNU, GPL licence in ustanovitev fundacije Free Software Foundation je bil Stallman leta 1990 odlikovan z redom MacArthur – ja (Barausse, Jermol, Stevens, 2006, str. 14).

Licenca GNU LGPL pa je kasneje nastala kot izpeljanka iz licence GNU GPL. Glavna razlika med GPL in LGPL licenco je v tem, da lahko odprto-kodno programje, ki je izdano pod LGPL licenco, vključimo tudi v lastniško programje. Torej lahko odprto-kodno programje, ki je izdano pod LGPL licenco, dodamo lastniškemu programju in tudi odprto-kodni del programja naprej prodamo pod lastniško licenco (Free Software Foundation, 2006).

3.2 Lastniške licence

Na področju lastniških licenc ne obstaja nobena krovna organizacija, ki bi določala pogoje, pod katerimi se lahko izda kakršnakoli lastniška licenca. Komercialno programje lahko glede na tip licence razdelimo v naslednje skupine (Jerič, 2004, str. 22):

- oglaševalno programje,
- skrbniško programje,
- lastniško programje,
- lastniško-demonstracijsko programje,
- demonstracijsko programje,
- zastonjsko programje,
- dopisniško programje in
- programje na pokušino.

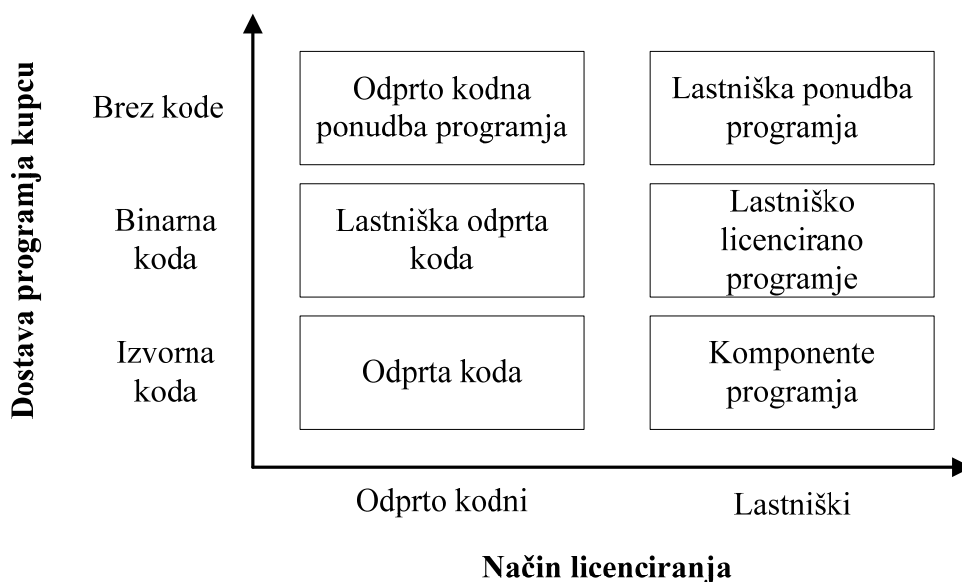
V nadaljevanju se bom osredotočil samo na lastniško programje. Druge vrste programja pa niso vsebina te naloge in jih zato tudi ne bom razlagal. Lastniško-demonstracijsko programje pa je pri ponudnikih dostopno kot del njihovih spletnih strani.

Lastniško programje je najbolj znano od vseh naštetih vrst programja. Tovrstno programje deluje tako, da ga lahko uporabljamo, če kupimo licenco. Cena licence je običajno določena s strani proizvajalca. Licenca je torej neke vrste pogodba med proizvajalcem programja in uporabnikom oziroma kupcem. V tej licenci so tudi navedeni pogoji, ki se jih mora uporabnik držati, če želi uporabljati programje. Uporaba komercialnega programja brez nakupa licence ni mogoča oziroma je nelegalna.

3.3 Povezava med licenciranjem in dostopom do programja

Slika 2 prikazuje, pod kakšnimi pogoji lahko kupec v različnih fazah pride do programja glede na način licenciranja. Namen slike je prikazati razlike v licenciranju programja. Kot vidimo iz analize tudi ponudniki odprto-kodnega programja uporabljajo lastniške licence.

Slika 2: Načini licenciranja programja

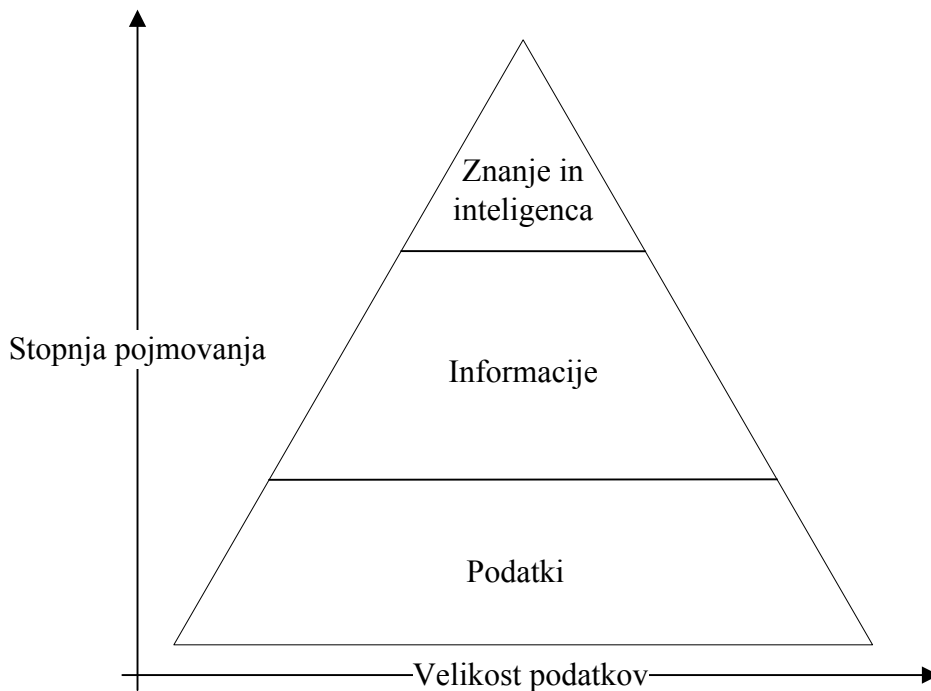


Vir: IDC, 2007.

4 Poslovna inteligenca

Ker je pojem poslovna inteligenca razmeroma nov, panoga pa se je šele dobro začela razvijati, je najbolje, da najprej ugotovimo, kaj poslovna inteligenca pomeni.

Slika 3: Piramida spremembe od podatkov do znanja



Vir: Loshin, 2003, str. 4.

Slika 3 prikazuje spreminjanje podatkov v informacije in nato naprej v znanje in inteligenco. Velikost podatkov so vsi podatki in informacije, ki so v obliki zajete v procesu delovanja podjetja. Ti podatki in informacije pa so lahko v kakršnikoli obliki. Stopnja pojmovanja pa nam prikazuje, kako dojemamo podatke oziroma njihovo spreminjanje glede na nam potrebne informacije. Prikazan je način, kako naj bi deloval sistem poslovne inteligence oziroma kako deluje izkoriščanje podatkov z namenom odločanja in ukrepanja v določenih položajih. S pomočjo današnjih sodobnih tehnologij in metodologij pa si le olajšamo prenos med podatki in znanjem.

4.1 Definicija

Steve in Nancy Williams (2007, str. 2) pravita, da poslovna inteligenca vsekakor ni posamezni proizvod, tehnologija ali metodologija. Je pa poslovna inteligenca kombinacija proizvodov, tehnologij in metod, s katerimi organiziramo ključne informacije, ki jih potrebuje vodstvo, da izboljša dobiček in učinkovitost poslovanja. Na poslovno inteligenco gledata kot na poslovno informacijo ali poslovno analizo v sklopu poslovnih procesov, ki vodijo do odločitev in dejanj. Te odločitve in dejanja pa izboljšajo izvajanje poslovnih procesov.

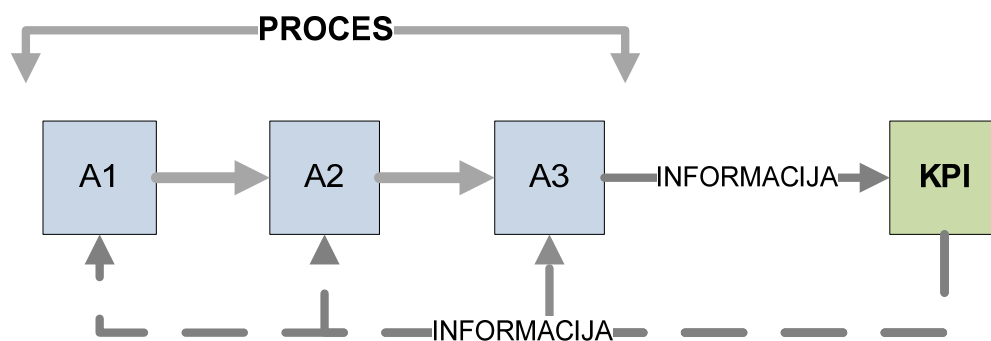
Data Warehouse Institute (2006), ki skrbi za izobraževanje in certificiranje vodilnih kadrov in informatikov za procese pri skladiščenju podatkov in poslovni inteligenca, definira poslovno inteligenca takole: "Poslovna inteligenca so procesi, tehnologije in orodja, potrebna za

transformacijo podatkov v informacije, informacij v znanje in znanje v plane, ki vodijo dobičkonosne poslovne dejavnosti. Poslovna inteligenca vključuje skladičenje podatkov, poslovna analitična orodja in upravljanje vsebine ter znanja."

Wikipedia (2006) pa opredeli poslovno inteligenco kot strokovno informacijo, znanje in tehnologije, ki učinkovito pomaga pri upravljanju organizacije ali samostojnega posla. Iz tega izhaja, da je poslovna inteligenca skupek programja in tehnologij za zbiranje, dostop in analiziranje podatkov, z namenom pridobivanja znanja o dejavnostih, ki vplivajo na poslovanje podjetja. Zelo pomembno je, da imamo v podjetju znanje o dejavnostih kot so npr.: kupci, konkurenca, poslovni partnerji, ekonomsko okolje podjetja in notranje operacije. Vse z namenom, ustvariti učinkovite in kakovostne poslovne odločitve. Poslovna inteligenca omogoča sprejemanje takih odločitev.

Vsekakor se poslovna inteligenca močno navezuje na poslovne procese v podjetju. Ključno pri poslovni inteligenci je, da se kar najbolje definira poslovne procese ter določi merila, s katerimi bomo spremljali učinkovitost delovanja teh procesov. Merila učinkovitosti delovanja procesov spremljajo različni ljudje na različnih nivojih poslovanja oziroma dejavnostih podjetja. Pri tem je potrebno dodati še, da se poslovni procesi lahko spreminjajo v času, zaradi enega ali več različnih dejavnikov.

Slika 4: Prikaz merjenja učinkovitosti delovanja procesa



Vir: Pentaho Open Source Business Intelligence Platform Technical White Paper, 2007, str. 4.

Slika 4 prikazuje merjenje učinkovitosti delovanja določenega poslovnega procesa. Učinkovitost delovanja procesa merimo s ključnim procesnim indikatorjem (*ang. key process indicator - KPI*). Aktivnosti so obarvane modro in imajo oznako A. Številka zraven oznake A pa pomeni zaporedno številko dejavnosti. Skupaj pa te dejavnosti tvorijo poslovni proces. Znotraj tega procesa se po dejavnostih prenaša tudi tok informacij, dokumentacijski tok, komunikacijski tok in še drugi kazalniki, s katerimi merimo učinkovitost posamezne dejavnosti. Na koncu je z zeleno obarvan kvadrataček z oznako KPI, ki meri učinkovitost delovanja vseh dejavnosti skupaj oziroma celotnega procesa. Povratno pa daje informacije

izvajalcem ali upravljalcem dejavnosti, da ti prilagodijo njihovo delovanje, če so rezultati delovanja procesa nezadovoljivi.

Največja težava večine podjetij v praksi je ta, da imajo slabo ali pa nimajo definiranih poslovnih procesov. Posledica tega je, da niso definirani kazalniki, s katerimi se meri učinkovitost delovanja procesa. Zato se posledično ne more spremljati učinkovitosti delovanja procesov in učinkovitost operativnega poslovanja podjetja. Rezultat tega pa so navadno visoki stroški poslovanja, neorganiziranost posameznih oddelkov znotraj podjetja in ustvarjanje slabega imena pri kupcih ter dobaviteljih. Dolgoročno pa to lahko privede tudi do propada podjetja.

4.2 Arhitektura poslovne inteligence

Ker sem kot središče diplomske naloge postavil odprto-kodno programje poslovne inteligence, sem se odločil, da bom predstavil arhitekturo poslovne inteligence ponudnika Pentaho. Arhitektura se bistveno ne razlikuje od arhitekture, ki jo uporablja drugo podobno programje, tako odprto-kodno kot tudi komercialno.

Po mnenju njihovih strokovnjakov je poslovna inteligenca razdeljena v štiri plasti (Pentaho, 2007):

1. Zunanji viri podatkov
2. Integracija podatkov in aplikacije
3. Platforma poslovne inteligence
4. Predstavitvena plast

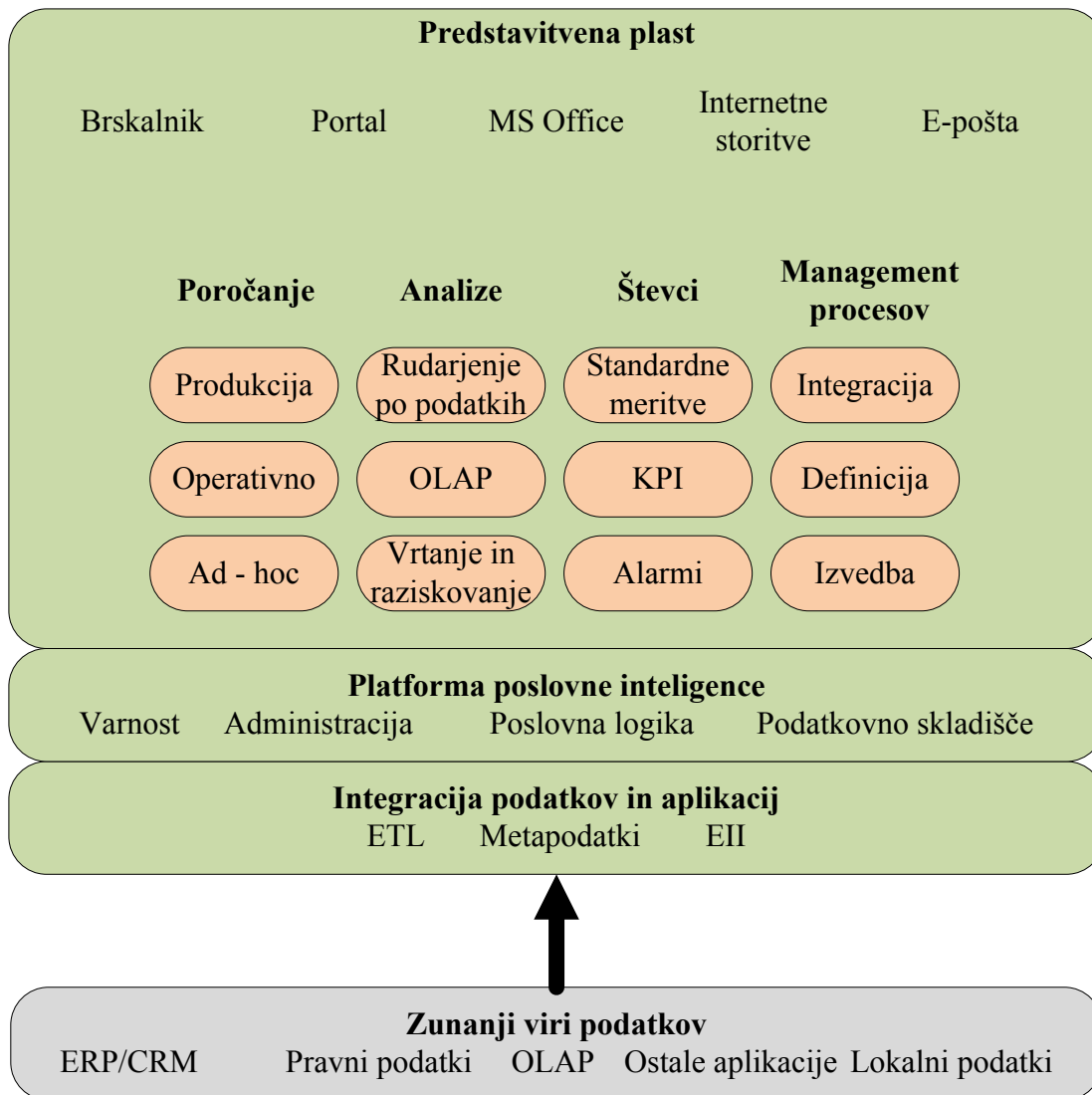
Plasti so našteje v zaporednem vrstnem redu od spodaj navzgor (*glej sliko 4*).

David Loshin pa v svoji literaturi ponudi nekoliko drugačno sestavo arhitekture poslovne inteligence (Loshin, 2003, str. 49): Obstajajo tri logične sistemske platforme, ki so potrebne za delovanje programa poslovne inteligence. Skrbijo pa za prenos informacij od vira do končnega cilja. Ti sistemi so:

1. Področje priprave podatkov: ciljno področje, kjer se podatki iz različnih virov zbirajo in integrirajo. Tako so nato pripravljeni za nalaganje v podatkovno skladišče.
2. Skladišče podatkov: centralizirano in stalno dostopno skladišče podatkov.
3. Analitično okolje: je sestavljeno iz področnih podatkovnih skladišč, ki črpajo podatke iz glavnega podatkovnega skladišča. Skupaj z analitičnimi vmesniki in orodji pa zagotavljajo vmesnik za uporabnika.

Kot vidimo iz navedenih definicij, je arhitektura podobna pri različnih sistemih poslovne inteligence. Določena področja so ponekod združena, druga pa razdružena. Potrebno pa je še poudariti, da včasih različni avtorji uporabljajo drugačne izraze za enako stvar, kar se vidi tudi iz navedenih definicij.

Slika 5: Arhitektura poslovne inteligence



Vir: Pentaho odprto kodno orodje za poslovno inteligenco, 2007.

Zunanji viri podatkov

Kot vidimo na sliki 5, gre pri aplikacijah in drugih virih podatkov za elektronske baze podatkov, ki zajemajo podatke iz operativnega delovanja podjetja. Cilj je zajeti vse operativne podatke, ki obstajajo v neki organizaciji v elektronski obliki.

Integracija podatkov in aplikacij

V tej plasti se podatki iz transakcijskih sistemov združijo, pregledajo in se nato prenesejo v naslednjo plast oziroma v skladišče podatkov. Za to skrbijo postopki ETL, EII ter tudi metapodatki. Slednji zgolj opisujejo operativne podatke.

ETL je proces izvlečenja (*ang. extract*), preoblikovanja (*ang. transform*) in nalaganja (*ang. load*) podatkov. Podatki se izvlečejo iz drugih operativnih elektronskih baz podatkov. Nato se preoblikujejo, po potrebi se dopolnijo manjkajoči podatki, odstranijo se slabi podatki in še druge funkcije. Po izvlečenju in preoblikovanju se podatki naložijo v podatkovno skladišče in so pripravljeni za uporabo preko drugih orodij poslovne inteligence. Za proces ETL obstaja tudi različno programje, ki nam olajša samo delo.

Meta podatki (*ang. metadata*) so podatki, s katerimi opisujemo druge podatke. Definicije podatkov se tako lahko imenujejo meta podatki. Primeri za meta podatke so: sheme, tabele, indeksi, pogledi in definicije stolpcev (Naude, 2006).

EII (Enterprise Information Integration) je tehnologija integracije, ki vleče in kombinira podatke iz več sistemov v realnem času. Pri tem pa jih ne skladišči na disk ampak ustvarja virtualno podatkovno skladišče. S tem pa odstrani potrebo po shranjevanju ali premikanju podatkov (Informatica, 2007). Postopek EII je novejši postopek in je zelo podoben postopku ETL.

Platforma poslovne inteligence

V podjetju Pentaho predstavljajo platformo poslovne inteligence nekoliko drugače kot jo predstavlja večina drugih ponudnikov, ter tudi nekoliko drugače kot je opredeljena v teoriji. Platforma so orodja, s katerimi zgradimo sistem poslovne inteligence. Običajno platforma poslovne inteligence združuje vse navedene ali pa vsaj večino plasti kot jih navaja Pentaho. Tako je platforma sestavljena iz povezav na zunanje vire podatkov, proces integracije in nalaganje podatkov v skladišče, ne glede na to, kateri tehnični postopek se za to uporabi, ter okolja, ki omogoča predstavitev in analizo podatkov. V nadaljevanju bom zato predstavil platformo poslovne inteligence kot si jo predstavljajo pri Pentahu. Druge navedene procese, ki spadajo v platformo, pa sem opisal že prej ali pa jih bom na koncu tega poglavja.

V platformo poslovne inteligence so vgrajena varnostna pravila, administracija celotnega programja poslovne inteligence, poslovna logika podjetja in podatkovno skladišče. Vsi navedeni pojmi se navezujejo na celoten sistem poslovne inteligence.

- Varnost lahko definiramo kot stanje brez nevarnosti in stanje brez izpostavljenosti škodi od nesreč ali napadov. Lahko pa jo definiramo tudi kot proces doseganja tega stanja. Cilj informacijske varnosti je optimizacija učinkovitosti delovanja organizacije

z vključevanjem možnosti tveganja, ki mu je organizacija izpostavljena (Kabay, Seymour, 2002, str. 2).

- Sistemska administracija je vzdrževanje in upravljanje računalnika, sistema ali mreže (Wikipedia, 2007).
- Poslovna logika je sestavljena iz poslovnih procesov in poslovnih pravil. Predstavlja pa modele poslovnih objektov kot so: računi, posojila, zaloge, itd. Predpisuje, kako se ti objekti med seboj povezujejo, ter uveljavlja poti in metode, ki nam omogočajo dostop in spreminjanje objektov (Wikipedia, 2007).
- Podatkovno skladišče je podatkovna struktura, ki je optimizirana za razpošiljanje. Zbira in hrani integrirane podatke iz več operativnih sistemov in jih shranjuje v enega ali več področnih podatkovnih skladišč (Williams, 2007, str. 201).

Predstavitvena plast

Predstavitvena plast je namenjena predstavljanju in analiziranju podatkov, ki so shranjeni v podatkovnem skladišču. Pentaho nam tako ponuja naslednje načine predstavitve podatkov: preko spletnih strani, izvozi podatkov v Microsoft Office aplikacije ali prednastavljena elektronska pošta, ki se pošlje, ko se zgodi nek dogodek. Najbolj pogosto uporabljane in dograjene so prav spletne strani, ki so navadno dostopne samo preko intranetnega omrežja v podjetju. Preko njih je možno predstaviti podatke v obliki tabel, števecv, grafov, možno je izdelovanje poročil, itd. Poleg samih predstavitev podatkov pa je omogočen tudi analitični del, ter izdelava in upravljanje s procesi v podjetju. Prenosi podatkov v Microsoft Office aplikacije, predvsem Excel, so nezaželeni in večinoma tudi nepotrebni, vendar je kljub temu funkcija prenosa omogočena, saj obstaja možnost, da neka določena analiza, ki bi jo želel izvesti uporabnik, ni na voljo.

4.3 Zgodovina

Začetki poslovne inteligence, v takšni obliki kot jo poznamo danes, segajo približno 20 let nazaj. Najti jo je bilo mogoče samo v velikih podjetjih, uporabljali pa so jo samo najbolj izučeni uporabniki. Programske rešitve so bile omejene na drage in zapletene statistične pakete ali standarde za poročila. Preglednice (npr.: Lotus 1-2-3) so postajale bolj priljubljene, vendar niso omogočale neposredne povezave s podatki. Za izdelavo analize je tako uporabnik moral ročno prepisati podatke iz natisnjene poročila, kar je imelo za posledico veliko izgubo časa in veliko napak pri prepisovanju. Kasneje se je pojavil že prvi sistem Executive information systems (EIS), ki je obljubljal analitiko za vodilne kadre v podjetjih. Pred 20-imi leti prav tako ni bilo skladišč podatkov. Danes se uporabljajo že v večini srednjih in velikih podjetij. Kapaciteta shranjevanja podatkov je do danes zrastle iz nekaj 100 megabitov (MB) na

več terabitov (TB) prostora. Prav tako je danes večje število strokovnjakov s tega področja, ki jih je možno dobiti preko trga dela ali preko zunanjih podjetij. Upravljalci podjetji pa so se začeli zavedati uporabnosti te tehnologije. Danes se prav tako z večino vodilnih kadrov v podjetjih, ki uporabljajo take sisteme, ni več težko pogovarjati o uravnoteženem sistemu kazalnikov (*ang. balanced scorecard*) ali sproti analitični obdelavi podatkov (*ang. online analytical processing – OLAP*). To seveda ni bilo mogoče pred 20-imi leti. Danes so orodja in tehnologije lažje uporabni, imajo več funkcij in so integrirani. Standardizacija internetnih vmesnikov in jeziki za delo s podatki (*ang. Data manipulation language – DML*) pa so poenostavili implementacijo in uporabo orodij za poslovno inteligenco (Dresner, 2007; Wikipedia, 2007).

5 Pregled programja poslovne inteligence

Za primerjavo programja sem si izbral naslednja podjetja oziroma njihove proizvode: Pentaho, JasperSoft, Spago BI, Cognos, Hyperion in Business Objects. Prva tri našeta nudijo odprto-kodno programje, zadnja tri podjetja pa so ponudniki lastniškega programja. Izbrana podjetja so specializirana za programje s področja poslovne inteligence in so tudi vodilna v svoji panogi. V nadaljevanju je kratka predstavitev navedenih podjetij, ter njihovega programja s področja poslovne inteligence.

5.1 Odprto-kodno programje

Pentaho

Podjetje Pentaho ima sedež v Orlandu, na Floridi v ZDA. Ustanovljeno je bilo leta 2004, ima 20 zaposlenih, njegov kapital pa znaša 13 milijonov ameriških dolarjev. Investitorja v podjetje sta Index Ventures in New Enterprise Associates. Izvršni direktor podjetja pa je Richard Daley. Ustanovili so ga strokovnjaki s področja poslovne inteligence, ki so pred tem razvijali programje za vodilne ponudnike lastniškega programja kot so: Business Objects, Cognos, Hyperion, IBM, Oracle in SAS. So pionirji na področju odprto-kodnega razvoja programja s področja poslovne inteligence. Njihov projekt poslovne inteligence se imenuje enako kot podjetje, torej Pentaho. Podjetje ima vizijo prehiteti vse tekmece na trgu poslovne inteligence, tako odprto-kodne kot komercialne. Njihovi konkurenti rastejo skozi prevzeme drugih podjetij in ne toliko zaradi samega razvoja programja, v čemer vidijo priložnost za rast. Kot pravijo sami, želijo, da se kupec za njih odloči ne samo zato, ker so odprto-kodni ampak zato, ker so boljši (Cubarrubia, 2006).

Pentaho se je dobro uveljavil in ima vse možnosti za hitro rast na trgu poslovne inteligence, katerega vrednost ocenjujejo na 23 milijard dolarjev. Za konkurenčnost proti drugim velikim podjetjem v panogi bo moral Pentaho predvsem pritegniti še kakšnega večjega kupca. Glede

na ogromno izkušenj, ki jih imajo, menijo, da s tem ne bodo imeli prevelikih težav (Cubarrubia, 2006).

Programje Pentaho je odprto-kodni projekt in je bilo razvito s pomočjo skupnosti razvijalcev povsod po svetu. Povprečno pa se sname z interneta 50.000-krat na mesec. Stroški implementacije programja naj bi znašali od 20.000, za normalne implementacije, in do 50.000 ameriških dolarjev, za velike implementacije (Cubarrubia, 2006).

Razvoj sestavin programja sloni tako na skupnosti razvijalcev kot na lastnem razvojnem oddelku. Razvojni oddelek tudi pregleda ter vgradi sestavine, ki jih je razvila skupnost, ter skrbi za izdaje novih inačic. Poleg tega želijo s prirejanjem narediti programje uporabniku čim bolj prijazno (out of the box). S tem zmanjšajo stroške implementacije in drugih podpornih dejavnosti pri uporabniku. Nudijo pa tudi tehnično podporo, šolanje ter svetovanje pri razvoju in implementaciji. Programje je mogoče brezplačno sneti preko spletnih strani podjetja. Podporne dejavnosti s strani podjetja Pentaho ali pooblaščenega partnerja pa so plačljive (Splošne informacije. b, 2007).

Programje je razdeljeno na več pregledov podatkov: poročila, rudarjenje po podatkih, analize, tok dela (workflow), digitalne delovne table (dashboard), ter platforma. Slednja ni namenjena pregledovanju podatkov ampak zagotavlja delovanje celotne aplikacije. Pentaho je v vključil tudi druge že dobro razvite odprto-kodne projekte: Mondrian (OLAP strežnik), JFreeReport (poročila), Kettle (integracija podatkov), Weka (rudarjenje po podatkih), drugi deli platforme so bili razviti v Pentahu ali njihovi skupnosti in spadajo neposredno pod samo programje Pentaho (Splošne informacije. b, 2007).

JasperSoft

Podjetje JasperSoft ima sedež v San Franciscu, Kalifornija, ZDA. Ustanovljeno je bilo leta 2005, njegov izvršni direktor pa je Paul Doscher. Ima nekaj manj kot 50 zaposlenih, začetni kapital pa znaša 23,5 milijona ameriških dolarjev. Glavni investitorji v podjetje so Partech International, Discovery Ventures, Morgenthaler Ventures in Doll Capital Management (Cubarrubia, 2006).

Ciljni kupci JasperSofta so večja podjetja, kljub temu pa so njihovi kupci tudi srednje velika in manjša podjetja. Pri tem je potrebno velikost podjetja razumeti v svetovnem in ne slovenskem merilu. Njihovo programje je mogoče integrirati v obstoječe informacijske sisteme v podjetjih ali pa jih postaviti kot samostojne sisteme s povezavo na vire podatkov. Glavna prednost pred konkurenco je cena in manjša kompleksnost za uporabo. Imajo že približno 5.000 kupcev v 81-ih državah. Kot največji uspeh pa štejejo, da je podjetje Siemens Molecular Imaging zamenjalo rešitev Oracle reports z rešitvijo JasperSoft (Cubarrubia, 2006).

Odrpto-kodni projekt JasperSoft je na voljo preko portala JasperForge.org. Skupnost, ki podpira projekt, šteje več kot 33.000 članov. Trenutno se v okviru JasperSofta izvaja 150 projektov (Splošne informacije. a, 2007).

JasperSoft sestavljajo naslednje komponente: JasperReports (poročila), iReport (poročila), JasperServer (strežnik), JasperAnalysis (analize), JasperETL (integracija podatkov), JasperBabylon (podpora za različne jezike), JasperIntelligence (platforma na osnovi spletnih tehnologij). Poleg navedenih komponent pa so na voljo tudi nekateri drugi odrpto-kodni projekti, ki pa niso neposredno povezani z glavnim projektom. Vsi omenjeni projekti so na voljo kot odrpto-kodni ter prosto dostopni preko interneta. JasperServer in JasperAnalysis pa sta na voljo tudi v lastniški različici (Opis produktov. c, 2007).

Poleg proizvodov podjetje nudi tudi tehnično podporo, šolanje ter svetovanje, ne glede na to ali smo namestili komercialni ali odrpto-kodni proizvod. Kakršnakoli podpora s strani podjetja je plačljiva (Opis podjetja. c, 2007).

Spago BI

Podjetje Engineering Ingegneria Informatica s.p.a, iz Italije, je zadolženo za vodenje projekta Spago BI. So vodilni ponudnik na področju informatike v Italiji in imajo že 27-letno tradicijo. Letni promet je znašal 425 milijonov evrov, ter raste po približno 10-odstotni stopnji letno. Zaposlenih imajo približno 200 ljudi, poleg tega pa imajo še 50 partnerskih podjetij. Večino teh podjetij je v Italiji, nekaj pa še drugje po Evropi. Podjetje se v večini ukvarja z implementacijo in razvojem programja. Spago BI pa je le eden od proizvodov, ki ga ponujajo.

Spago BI je prvo odrpto-kodno programje, narejeno v Italiji. Namenjeno je državni upravi, ter večjim podjetjem. Sestavljeno je iz več drugih odrpto-kodnih sestavin. Projekt se navezuje na več odrpto-kodnih skupnosti, od katerih vsaka razvija določen del, ki je vgrajen v rešitev. Poleg tega pa lahko namesto odrpto-kodnih v ponudbo vključijo lastniške sestavine (npr. Microsoft Analysis Server). Različne rešitve so odvisne od potreb in želja stranke. Spago BI je izdan pod licenco GNU LGPL.

Programje je sestavljeno iz naslednjih sestavin: Poročila, rudarjenje po podatkih, števci, QBE (poizvedbe po primeru), tok dela, analize po geografskih ločenostih z zemljevidi, integracija podatkov (ETL), metadata upravljanje s podatki ter podpora za delo z dokumenti.

Vir: Opis podjetja. d, 2007, Opis produktov. d, 2007.

5.1 Lastniško programje

Cognos

Cognos je eden od večjih ponudnikov programja s področja poslovne inteligence. Njihove proizvode in storitve uporablja več kot 23.000 kupcev, v 135-ih državah. Po svetu pa imajo še približno 3.000 lokalnih partnerjev in prodajalcev (Opis produktov. b, 2007).

Trenutna aplikacija, ki jo ponujajo, je Cognos 8 Business Intelligence. Temelji na isti arhitekturi kot prejšnja inačica. Vključuje pa tudi funkcijo iskanja informacij po celotnem informacijskem sistemu podjetja. Prilagodi posamezne preglede podatkov po uporabnikih. V prvi polovici leta 2007 je Cognos začel ponujati tudi prejemanje informacij preko mobilnih naprav (Opis produktov. b, 2007).

Z izdajo rešitve Cognos 8 je Cognos precej izboljšal svojo platformo za poslovno inteligenco. Predvsem se izboljšave poznajo na področju poslovno inteligenčnega meta podatkovnega modela, ki sedaj podpira več različnih področij kot so: poročanje, ad hoc poizvedbe, OLAP, opomniki dogodkov in uravnotežene sisteme kazalnikov. Do konca leta 2006 je tako Cognos pri 2.900 svojih kupcih že nadgradil starejše inačice v inačico 8. V zadnjem času pa je Cognos zaradi precejšnjega napredka konkurentov, predvsem Microsofta, Oracla in SAP, zapadel v manjšo prodajno krizo. Poskušajo se rešiti z izdajo priročnika za hitro gradnjo rešitev s področja poslovne inteligence. Izdelek se je že pokazal kot dober dodatek k prodaji aplikacije. Nove priložnosti s področja prodaje pa so se odprle s prevzemom Celequesta. Podjetje Celequest je specializirano za področje shranjevanja podatkov in dodatnega programja, ki je v pomoč pri obvladovanju platforme. Razvili so tehnologijo, ki omogoča shranjevanje podatkov v spominu in ne v bazi podatkov. Posledica tega so hitrejše poizvedbe (Magic Quadrant for Business Intelligence Platforms, 2007; Ferguson 2007).

Aplikacija Cognos 8 je sestavljena iz naslednjih sestavin: poročila, analize, uravnoteženega sistema kazalnikov, števec, upravljanja poslovnih dogodkov in integracije podatkov (Opis produktov. b, 2007).

Hyperion

Hyperion se je leta 2004 uvrstil na lestvico FORTUNE 100 podjetij, za katera je najboljše delati. Svoje kupce ima v 90-ih državah po vsem svetu. Mreža partnerjev je sestavljena iz približno 600 podjetij, ki nudijo podporo ali razvoj proizvodom Hyperion. Letni promet je v prejšnjem fiskalnem letu znašal 765 milijonov ameriških dolarjev. Aprila leta 2007 je bil Hyperion prevzet s strani Oracla (Opis podjetja. a, 2007).

Z izdajo nove inačice System 9, je Hyperion odpravil tradicionalne težave, ki jih je imel s poročili ter integracijo. Kljub temu pa je Hyperion zadržal zelo močan analitični del

programja. Z nadgraditvijo sistemov, na System 9, in z odpravo omenjenih tehničnih pomanjkljivosti je Hyperion pri kupcih spet zasedel mesto med vodilnimi v panogi. Glavna prednost Hyperiona pred drugimi konkurenti je zelo močna povezava na finančno poslovanje oz. povezava neposredno na finančnega direktorja podjetja. Ta rešitev vleče za sabo ves drugi sistem poslovne inteligence. Poleg tega je Hyperion nakazal, da želi v prihodnosti združiti vse podatke iz različnih podatkovnih virov v eno podatkovno skladišče. Od tu pa bodo na voljo za različne prikaze, ne samo za ad hoc poizvedbe, kot je bilo to do sedaj. To je Hyperion storil z prevzemom podjetja Razza. Hyperion lahko zaradi tega sedaj ponudi OLAP možnosti za veliko večje količine podatkov. To pa je ključna sestavina, ki bo omogočila Hyperionu, da zavzame položaj vodilnega ponudnika poslovne inteligence tudi na drugih področjih in ne samo na finančah. Zgoraj omenjeno tehnologijo, ki jo je Hyperion prevzel od Razze, je potrebno še integrirati v celotno platformo. Tako lahko pričakujemo, da bodo funkcije, vključno z poročanjem, števci in OLAP, začele popolnoma delovati v sredini leta 2007 (Magic Quadrant for Business Intelligence Platforms, 2007).

Business Objects

Business Objects je ponudnik programja s področja poslovne inteligence. Ima več kot 42.000 kupcev in 3.000 lokalnih partnerjev in prodajalcev razpršenih po vsem svetu. Več kot 80 % podjetij s seznama Fortune 500 je njihovih strank. V podjetju je zaposlenih več kot 5.000 ljudi po vsem svetu. Gartner uvršča podjetje Business Objects med vodilne ponudnike rešitev s področja poslovne inteligence za podjetja. IDC pa jih označuje kot vodilno podjetje na področju poslovne inteligence (Opis produktov. a, 2007).

Podjetje je prvo v panogi, ki je uspelo preseči milijardo ameriških dolarjev prometa. To ga postavlja na prvo mesto med ponudniki programja s področja poslovne inteligence. Uspeh je posledica integracije rešitve za poročila v platformo poslovne inteligence in izdaja nove inačice BusinessObjectsXI. V bližnji prihodnosti je pričakovati nadaljnjo rast podjetja, kar bo predvsem posledica dobrega partnerskega programa ter prevzemov specializiranih proizvajalcev s področja poslovne inteligence (Magic Quadrant for Business Intelligence Platforms, 2007).

Njihov glavni proizvod je BusinessObjectsXI. Sestavljen je iz naslednjih sestavin (Opis produktov. a, 2007):

1. Platforma: Sestavlja običajne storitve za čim preprostejšo vzpostavitev in nadzor nad delovanjem sistema
2. Napredno poročanje: Omogoča dostop do velike količine podatkov, ki jih je nato preprosto pretvoriti v lepo oblikovana poročila.
3. Poizvedbe in analize: Omogoča uporabniku, da poizveduje za katerokoli informacijo, ki mu je dovoljena, brez kakršnegakoli naprednega znanja o delovanju sistema ali načina hranjena podatkov.

4. Upravljanje z učinkovitostjo delovanja podjetja (EPM): Prikazuje informacije z različnih področij o ciljih, ki so napisani v strategiji poslovanja podjetja.
5. Upravljanje z informacijami v podjetju (EIM): Skrbi za integracijo ter kakovost podatkov, ki vstopajo v sistem.

Programje omogoča povezovanje z vsemi najbolj razširjenimi formati izmenjave podatkov (Microsoft Office formati, Pdf Reader, XML,...)

5.2 Primerjava programja

V zadnjem času se je razvila razprava o programju oziroma modelih delovanja, ki jih uporabljajo ponudniki oziroma razvijalci. Na eni strani so se znašli zagovorniki prostega in odprto-kodnega programja in na drugi strani zagovorniki lastniškega programja. V središču dogajanja pa so se znašli ponudniki oziroma razvijalci programja. V tem poglavju bom zato predstavil prednosti in slabosti posamezne vrste programja, ter tudi modele delovanja velikih računalniških podjetij na eni in razvijalskih skupnosti na drugi strani. Predstavitev bo kratka in zelo okvirna, saj to ni glavna tema diplomske naloge. S tega področja pa že obstajajo tudi raziskave, ki temeljijo na matematičnih modelih in skušajo skozi številke prikazati prednosti in slabosti določene vrste programja.

Dogajanje na trgu

Zadnje raziskave trga kažejo, da je odprto kodno programje prevzelo vodstvo pred lastniškim programjem na nekaj proizvodnih področjih. Raziskava, ki jo je opravil Netcraft kaže, da je Internetni strežnik Apache vodilni na področju internetnih strežnikov. Omenjeni strežnik ima kar 67-odstotni tržni delež, kar je precejšnja prednost pred drugouvrščenim Internet Information Server, ki ga ponuja Microsoft in ima 21-odstotni tržni delež. Obstaja še nekaj drugih odprto-kodnih proizvodov, ki dosegajo zavirljive tržne deleže, kot so npr.: SendMail, FreeBSD, MySQL, itd. (Ohlhorst, 2004, str. 1).

Raziskava organizacije IDC kaže, da podjetja letno kupijo za 98 milijonov ameriških dolarjev odprto-kodnih storitev in proizvodov. Ta številka pa naj bi se do leta 2008 povzpela kar na 288 milijonov ameriških dolarjev letno (Ohlhorst, 2004, str. 1).

Skupne točke

Lastniško programje je včasih poslano na trg kot preizkusno programje (*ang. shareware*). Značilno zanj je, da je uporabno brezplačno, vendar samo v določenem obdobju. Po tem obdobju pa moramo kupiti licenco. Lahko pa ga dobimo tudi kot zastonjsko programje (*ang. freeware*), kar pa pomeni, da je brezplačno uporabno v nedoločenem obdobju. Kakorkoli pa pri nobeni od navedenih različic programja nimamo dostopa do izvorne kode. Licenca pa nam

prepoveduje kopiranje, modifikacije in preprodajo (Torfinn, 2007, str. 5). Druge vrste lastniškega programja so našteje v poglavju o licenciranju.

Prednosti odprto-kodnega programja

Glavne prednosti odprto-kodnega programja, v primerjavi z lastniškim programjem, so (Torfinn, 2007, str. 3, 6):

Neposredna primerjava med odprto-kodnim programjem in programjem Microsofta je pokazala, da so stroški implementacije in začetka uporabe nižji pri uporabi odprto-kodnega programja.

V svoji osnovi temelji razvoj odprto-kodnega programja na načelu deljenja. Posledično pa pospešuje izmenjavo komunikacij in znanja.

Raziskave so pokazale, da izmenjava programja pospešuje lokalno sposobnost, ustvarjalnost, lokalizacijo in neodvisnosti od komercialnih podjetij. Istočasno pa se lahko zmanjšujejo tudi stroški. Kitajsko ministrstvo za šolstvo je v sodelovanju s podjetjem IBM začelo projekt povezovanja 200.000 študentov v 100 univerzah. Podjetje IBM je znano po razširjanju odprto-kodnega operacijskega sistema Linux.

Odprto-kodno programje je alternativa ilegalnemu piratskemu programju: Odprto-kodno programje nudi legalno in cenovno ugodno alternativo za tiste, ki si ne morejo privoščiti dragega programja ali pa ne želijo uporabljati piratskega programja. Ta težava je še toliko večja v državah, ki vstopajo v Svetovno trgovinsko organizacijo (World Trade Organization – WTO), preko katere velike multinacionalke izvajajo pritisk glede uporabe intelektualnih pravic.

Odprto-kodno programje je boljše od lastniškega pri zagotavljanju lokalne jezikovne podpore: Glavni jezik v današnji računalniški industriji je angleščina. Večja udeležba neangleško govorečih je mogoča le, če so njihovi jeziki uporabljeni v programju. Proces, ki se imenuje lokalizacija, skrbi za to, da se dokumentacija in vsebina objavljata v lokalnih jezikih. Razlog je v tem, da je lastniško programje zaprto in zato ne moremo lokalizirati jezikovne podpore. To lahko stori samo podjetje, ki ga je proizvedlo. Podjetje pa bo to storilo le, če bo ugotovilo, da je tržišče dovolj veliko, da je lokalizacija ekonomsko upravičena.

Slabosti odprto-kodnega programja

Glavne slabosti odprto-kodnega programja so (Torfinn, 2007, str. 4, 6):

Stroški odprto-kodnega programja: Potrebno je razmisliti o celotnem projektu vpeljave informacijskega sistema. Tako je potrebno k ceni licence prišteti tudi stroške nadaljnega razvoja, vzdrževanja in šolanja uporabnikov. Ker je navadno odprto-kodno programje

ugodnejše z vidika nakupa licence, to še ne pomeni, da bomo zaradi tega ceneje izpeljali projekt.

Odvisnost: Kljub navidezni svobodi pri razvoju in prilagajanju programja, ki jo dobimo z odprto-kodno licenco, je potrebno upoštevati, da veliko razvijalcev uporablja svoje standarde. Ti standardi velikokrat niso usklajeni z drugimi razvijalci. Zato je potrebno paziti, da podjetje, ne postane preveč odvisno od enega samega ali od skupine razvijalcev.

Odprto-kodno programje ni in ne bo nujno vedno zastoj: Na trgu obstaja precej ponudnikov odprto-kodnega programja, ki zaračunavajo tako licenco kot tudi nadaljnje prilagoditve in vzdrževanje. Eden takih večjih ponudnikov je podjetje Red Hat, ki ponuja celo vrsto komercialnih različic odprto-kodnega operacijskega sistema Linux. Cene odprto-kodnega programja pa se po njihovem ceniku gibljejo tudi do 18.000 ameriških dolarjev.

Spreminjanje zakonodaje in poseganje na trg: V prihodnosti se lahko zgodi, da bodo vlade po svetu začele, zaradi kakršnihkoli pritiskov, sprejemati zakonodajo, ki bo omejevala odprto kodno skupnost in razvoj. Posledično lahko pride do višanja cen odprto-kodnega programja. Čeprav je verjetnost, da se kaj takega zgodi zelo majhna, so se odprto-kodne organizacije začele pripravljati na poganja v primeru, da bi do tega prišlo.

Odprto-kodna skupnost ni sestavljena samo iz zanesenjakov: Običajno se jo predstavlja kot skupino zanesenjakov, ki jih ne vodi motiv dobička. Za svoje delo ne zahtevajo plačila. V resnici so velika večina teh zanesenjakov, razen samih razvijalcev, velika podjetja, ki uporabljajo Linux in potrebujejo dodatne rešitve, ki bodo podprle njihove procese.

Ponudniki lastniškega programja se odzivajo na izzive s strani odprto-kodne skupnosti: Microsoft je leta 2001 kot odgovor na širjenje odprto-kodnega programja, ustanovil Shared Source Initiative. To je program, preko katerega bodo začeli izmenjevati kodo operacijskega sistema Windows z vladami, podjetji in izobraževalnimi ustanovami. Kljub temu pa kode operacijskega sistema uporabniki ne bodo smeli spreminjati.

Udeležba v razvoju in dostop do odprto-kodnega programja zahtevata povezanost z internetom: Internet je za odprto-kodno programje ključnega pomena. Z vidika razvoja je pomembno, da omogoča razvijalcem skupno pisanje kode, oziroma sočasni razvoj. Poleg tega pa omogoča tudi izmenjavo programja. Po drugi strani pa je internet pomemben distribucijski kanal. Večino odprto-kodnega programja ne moremo kupiti v trgovinah, ampak jih lahko dobimo samo preko interneta. Mnoge države po svetu nimajo vzpostavljene ustrezne infrastrukture, ki bi odpravila omenjeno težavo.

Tehnično osebje je bolj podkovano pri uporabi lastniškega programja: Če se bodo podjetja v prihodnosti odločala za uporabo večjega dela odprto-kodnega programja, bo prišlo tudi do pomanjkanja osebja, ki bo lahko vzdrževalo omenjeno programje.

Stanje v velikih podjetjih

Podjetje Google, ki je najbolj znano po svojem iskalniku, uporablja za platformo svojega iskalnika operacijski sistem Linux. Google se opira na odprto kodno skupnost pri pridobivanju novih inačic Linuxa, ki ga nato prilagaja svojim potrebam, s čimer ustvarja tudi svojo konkurenčno prednost. Vendar obratno Google ne deli prilagojenega dela Linuxa z drugo skupnostjo. V zameno pa Google finančno podpira odprto kodno razvijalsko skupnost (DeLong, 2006, str. 4).

IBM je v svoji poslovni strategiji zapisal, da bo v prihodnosti povečeval del svetovalnih storitev. Svetovanje v IBM-u predstavlja več kot 50 % proračuna podjetja. V podjetju so tako kot osnovo postavili razpošiljanje operacijskega sistema Linux. Na tej podlagi pa bodo nato prodajali svoje svetovalne storitve in drugo lastniško programje (DeLong, 2006, str. 3, 4).

Primerjava modelov razvoja programja

Razvoj interneta je v industrijo razvoja programja prinesel nov dejavnik. Spremenil je dejstvo, da lahko sedaj razvijalci in uporabniki sestavljajo skupno celoto in ne razdeljenih otokov. Razvijalci so lahko vsi ljudje, ki jih neka težava zanima in ne samo določena skupina ljudi, ki dela na specifični težavi. Razvoj odprto-kodnega programja je nekoliko spremenil organizacijo ljudi, ki so zadolženi za rešitev določene razvojne težave. Sam razvojni proces združuje različne ljudi z vsega sveta glede na težavo, ki jo je potrebno rešiti. Pri tem pa se ne omejuje z organizacijo. V nadaljevanju bom zato predstavil nekaj dejstev v številkah (Tiemann, 2007, str. 2, 3):

Sprememba modela razvoja je zelo spremenila pogled na razvoj in odstranila ovire, ki so obstajale pri običajnem razvojnem postopku in industrializaciji panoge. Za primer si pogledjmo portal sourceforge.net. Sourceforge.net je odprto-kodni razvojni portal, ki gosti več kot 100.000 projektov in ima več kot 1.200.000 registriranih uporabnikov. Obsežna raziskava, ki je bila opravljena leta 2002 in dopolnjena v letu 2005, je pokazala, da ima omenjeni portal več kot 490.000 dejavnih razvijalcev, ki za razvoj svojih projektov porabijo v povprečju več kot 10 ur na teden. Zmnožek nam pokaže številko približno 5 milijonov ur na teden. Trije najpogostejši odgovori na vprašanje, zakaj razvijalci porabijo toliko svojega časa za razvoj, pa so:

1. Zato, ker je zabavno
2. Zato, ker nam omogoča razvoj naših spretnosti in sposobnosti
3. Zato, ker je dobro za družbo

V tem izračunu niso upoštevani razvijalci, ki sodelujejo na nekaterih drugih odprto-kodnih projektih kot so: Linux, Apache, GNU in še nekateri drugi.

Sedaj pa naredimo primerjavo z najuspešnejšim ponudnikom lastniške programske opreme. To je podjetje Microsoft. Če bi vseh 61.000 zaposlenih pri Microsoftu pisalo kodo za

programsko opremo, bi morali delati 80 ur na teden, da bi dosegli količino 5 milijonov ur. Če bi pri Microsoftu plačali programerje po ceni 25 ameriških dolarjev na uro, bi za 80-urni delavnik na teden porabili svoj celoten proračun, ki ga ima na voljo oddelek za raziskave in razvoj in znaša 6,6 milijarde ameriških dolarjev na leto.

Iz tega lahko sklepamo, da se je sourceforge.net izenačil ali celo prehitel Microsoft proizvodni potencial. Pri tem pa je bil uporabljen socialni in ne industrijski model. Pri tem je industrijski model mišljen kot proizvodno podjetje, ki razvija in nato prodaja svoje proizvode na trgu. Če bi tu zraven priključili še vse druge odprto-kodne razvijalce, pridemo do ugotovitve, da je nastal nov razvojni model, ki je presegel meje tradicionalnega industrijskega pristopa.

Poglejmo še, kako je z številom napak pri razvoju odprto-kodnega in lastniškega programja. Lastniško programje ima približno 20 do 30 napak na vsakih 1000 vrstic kode. Številka je več ali manj nespremenjena od 60-ih let prejšnjega stoletja. Pri meritvah števila napak v izvorni kodi Linuxa pa so rezultati naslednji: leta 2004 je bilo v 5,7 milijona vrstic kode najdenih 985 napak. Leta 2005 je izvorna koda Linuxa zrastle za 4,7 %, pogostost napak pa se je zmanjšala za 2,2 %. Vse večje napake pa so bile odpravljene v roku 6-ih mesecev.

Model je šel tako daleč, da je Fundacija Gates, ki jo vodi Melinda Gates objavila, da ne bo več financirala raziskav s področja HIV, če raziskovalci ne bodo pripravljeni deliti svojih ugotovitev z drugimi sodelavci. Fundacija skrbi za financiranje raziskav s področja HIV. Odločitev pa je posledica dejstva, da v zadnjih 20-ih letih kljub intenzivnim raziskavam še niso iznašli zdravila za AIDS. Ustanovitelja fundacije sta Melinda in Bill Gates. Bill Gates pa je lastnik Microsofta, največjega ponudnika lastniškega programja, kar je mogoče celo malo ironično.

6 Kriteriji za izbiro programja poslovne inteligence

Pri kakovostni odločitvi o izbiri programja morajo podjetja ali organizacije upoštevati strateška in tehnična merila ter upoštevati morebitne težave, ki se lahko pojavijo pri implementaciji.

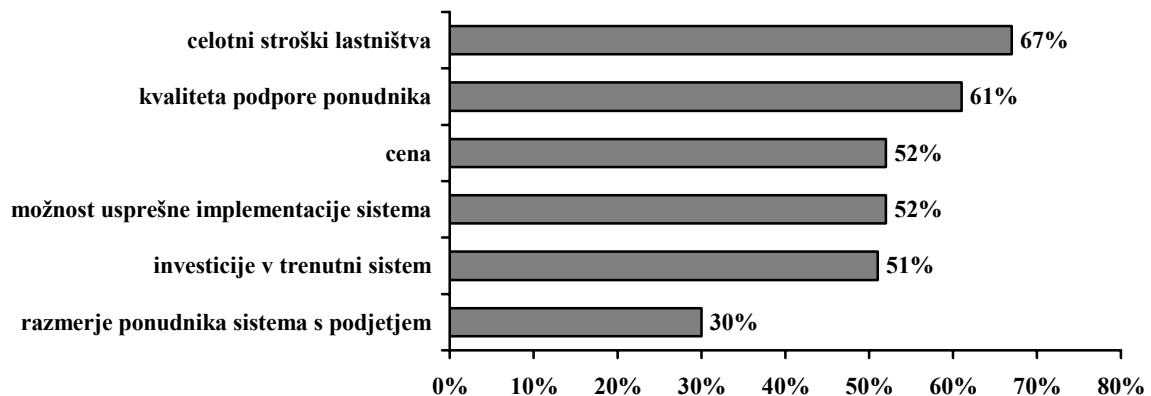
Ena od možnosti izbire programja je ta, da oddelek informatike najprej določi stanje tehnične podpore v podjetju. Tu je tehnična podpora mišljena kot strojna oprema, na katero se bo namestilo programje in znanje kadrov na področju informatike, ki bodo sodelovali pri projektu izbire programja. Znanje kadrov je pomembno z vidika prirejanja programja pri odprto-kodnem programju, kjer imamo dostop do izvorne kode. Po določitvi tehnične podpore v podjetju se izbere skupina zaposlenih, ki je sestavljena iz sedanjih in prihodnjih uporabnikov programja. Ta skupina pa nato izbere programje, ki ustreza tehnični podpori v podjetju. V večini podjetij je ta možnost skoraj neizvedljiva.

Druga možnost pa je ocenitev mogočih rešitev različnih ponudnikov po že vnaprej pripravljenih strateških in tehničnih merilih. Pri tem pa upoštevamo tudi možnost, da pride do same zamenjave tehnične podpore.

Obstaja še vrsta drugih pristopov, kako se lotiti izbiranja različnih ponudnikov programja. Vendar namen naloge ni predstavljati različnih pristopov. Odločil sem se, da uporabim drugi navedeni pristop.

Na podlagi raziskave oziroma ankete, ki jo je opravila organizacija TDWI, so ključni strateški kriteriji za izbiro programja naslednji (glej sliko 6): Celotni stroški lastništva (*ang. total cost of ownership*) 67 %, kvaliteta podpore ponudnika 61 %, cena 52 %, možnost uspešne implementacije sistema 52 %, investicije v trenutni sistem za podporo odločanju 51 %, razmerje ponudnika sistema s podjetjem 30 %.

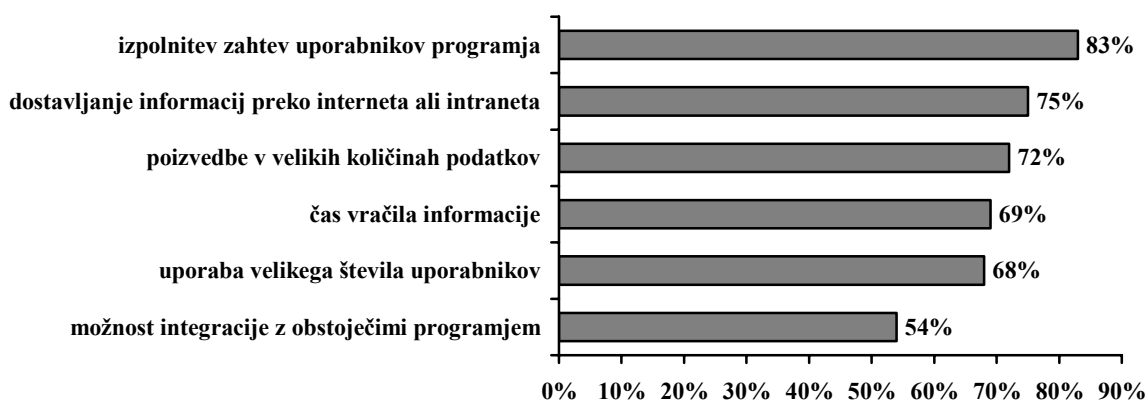
Slika 6: Strateški kriteriji za izbiro programja



Vir: Eckerson, 2005.

Tehnična merila za izbiro programja so naslednja (glej sliko 7): izpolnitev zahtev uporabnikov programja, in sicer z 83 %. Sledijo še dostavljanje informacij preko interneta ali intraneta z 75 %, nato možnost poizvedovanja v velikih količinah podatkov z 72 %, čas vračila informacije (*ang. response time*) 69 %, možnost uporabe velikega števila uporabnikov z 68 % in možnost integracije z obstoječim programjem 54 %.

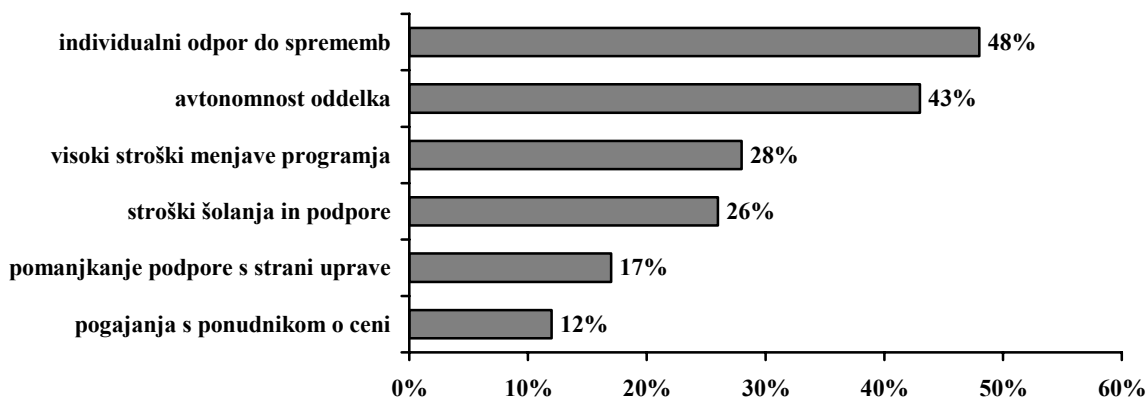
Slika 7: Tehnična merila za izbiro programja



Vir: Eckerson, 2005.

Poleg strateških in tehničnih meril je potrebno pred samo izbiro programja upoštevati tudi nekatere druge dejavnike, ki se bodo pojavili ob sami spremembi organizacije delovanja podjetja. Ti dejavniki so po anketi pomembnosti razvrščeni po sledečem vrstnem redu (glej sliko 8): individualni odpor do sprememb 48 %, sledi avtonomnost oddelka 43 %, visoki stroški menjave programja 28 %, stroški šolanja in podpore 26 %, pomanjkanje podpore s strani uprave podjetja 17 %, pogajanja o ceni s ponudnikom 12 %.

Slika 8: Dejavniki, ki vplivajo na uspešnost spremembe organizacije



Vir: Eckerson, 2005.

Zgoraj omenjena merila sem razčlenil na bolj podrobna. Razdelil sem jih na dve skupini: tehnična in strateška merila. Merila sem razdelil na strateška in tehnična, ker želim prikazati, da se morajo o takem projektu zbrati tako mnenja vodstva podjetja kot tudi tehničnih strokovnjakov. Vsak pa izvede analizo za svoje področje. Lahko pa merila tudi drugače razdelimo, npr. po modulih ali drugih različnih kategorijah, lahko pa jih pustimo tudi skupaj. Za bolj kompleksne analize je pametno uporabljati tudi združevanje meril v različne nivoje. V tem delu bom merila samo opisal, da bodo razumljiva, kasneje pa jih bom uporabil pri prikazu izbire programja.

6.1 Strateška merila

Strateška merila so v prvi vrsti namenjena splošni oceni programja, ki ga izbiramo. Cilj analize po strateških merilih pa naj bo ocena, ki bo povedala, ali se določena ponudba sklada s poslovno strategijo poslovanja podjetja (Seznam meril za izbiro ponudnika informacijskega sistema, Agregata, d. o. o., 2007).

1. Cena: Cena se navezuje na nabavno ceno programja brez morebitnih stroškov implementacije, šolanja in drugih možnih dodatnih storitev s strani ponudnika.
2. Licenca: Pod kakšno licenco se nahaja programje, ki ga želimo implementirati. Merilo je pomembno pri odprto-kodnih projektih.
3. Podpora pri implementaciji: Merilo je pomembno predvsem pri odprto-kodnih aplikacijah, kjer nas zanima, kakšne vrste pomoč nam ponudnik nudi pri implementaciji programja.
4. Lokalna podpora: Ali imamo možnost lokalne podpore. Zanima nas, ali ima ponudnik programja svojega predstavnika v Sloveniji za slovensko tržišče. Merilo pride v poštev predvsem pri podjetjih, ki imajo dislocirane poslovne enote. Tako nas zanima, če nam ponudnik nudi podporo v različnih državah, kjer imamo naše poslovne enote.
5. Podpora pri razvoju: S tem merilom je mišljeno, da nam ponudnik nudi podporo pri načrtovanju kot samem programiranju programja. Lahko na našo zahtevo z njegovimi kadri ali pa samo kot svetovanje pri določenih delih razvoja.
6. Svetovanje: Ali ponudnik pri vpeljavi programja tudi svetuje pri rešitvi težav, ki se pojavijo z novim programje. Svetovanje lahko izvaja sam ali pa ga izvajajo pooblaščenih svetovalci.
7. Šolanje uporabnikov: Šolanje uporabnikov je pomembno, saj se sicer lahko zgodi, da programja ne bo nihče uporabljal. Merilo ocenjuje ponudbo šolanja s strani ponudnika ali njegovih pooblaščenih oseb.
8. Razširljivost: Merilo nam pove, kakšno možnost imamo, da sami implementiramo programje, ki je že del našega informacijskega sistema na druge sisteme znotraj organizacije ali podjetja.

6.2 Tehnična merila

Tehnična merila uporablja za ocenitev programja predvsem služba informatike, tako z vidika implementacije, same operativne rabe kot tudi vzdrževanja programja (Seznam meril za izbiro ponudnika informacijskega sistema, Agregata, d. o. o., 2007).

1. Vrsta podpore: Način, kako nam ponudnik programja nudi pomoč pri uporabi. Lahko nam jo nudi preko telefona, elektronske pošte, foruma, ... Merilo se oceni glede na vrsto podpore, ki je za podjetje najuporabnejša.

2. Možnost lastnega razvoja: Ocena možnosti lastnega razvoja ali spreminjanja programja. Zanima nas, ali lahko sami prilagodimo programje po svojih željah. Seveda je predpostavka, da ima podjetje zaposlenega programerja, ki pozna kodo in logiko programja.
3. Programiranje na zahtevo: Ali imamo po namestitvi programja možnost, da nam ponudnik na našo zahtevo in po naših željah spremeni ali doda določen del programja.
4. Možnost podpore s strani skupnosti: Merilo se presoja glede na velikost skupnosti, ki sodeluje pri projektu. Merilo je namenjeno predvsem ocenjevanju odprto-kodnega programja, kjer nam ponudnik ne nudi neposredne podpore. Lahko pa se uporablja tudi za ocenjevanje drugih aplikacij.
5. Velikost skupnosti: Kako velika je skupnost, ki sodeluje pri projektu. Zanima nas s stališča, ali je skupnost dovolj velika, da nam lahko nudi kakovostno podporo.
6. Poosebitev: Poosebitev pomeni prilagoditev programja posameznemu uporabniku ali skupini uporabnikov. Merilo ocenjuje stopnjo poosebitve. V podjetju ni zaželeno, da vsi vidijo vse informacije o poslovanju podjetja.
7. Materiali za šolanje: Merilo ocenjuje materiale, ki nam jih ponudnik posreduje na kakršenkoli način za učenje uporabe aplikacije. Ocenjuje tudi, ali nam ponudnik sploh nudi materiale za šolanje.
8. Tehnična dokumentacija: Kakšno tehnično dokumentacijo nam nudi ponudnik. Tehnična dokumentacija je zelo pomembna pri odprto-kodnem programju, ki je prosto dostopen preko interneta. Ponavadi je potrebno tako programje še dodatno prilagoditi, da ga lahko vključimo v lasten informacijski sistem.
9. Povezljivost: Programje je možno dodati drugemu programju. To programje pa je lahko obstoječe programje v podjetju ali pa se bo šele začelo uvajati.

7 Izbira programja poslovne inteligence

Sestavlil sem dve tabeli, v katerih so v stolpcih navedeni ponudniki programja s področja poslovne inteligence, v vrsticah pa so navedena merila. Prvi trije so ponudniki odprto-kodnega programja, drugi trije pa so ponudniki lastniškega programja. Tabeli se razlikujeta glede na merila, ki so upoštevana v analizi. V prvi tabeli so navedena strateška merila, v drugi tabeli pa so navedena tehnična merila.

Predpostavljam, da se programje izbira za srednje ali veliko slovensko podjetje. Velikost podjetja je vezana na slovenske razmere, merilo za določanje velikosti podjetja pa je število zaposlenih. Podjetje je lahko iz katerekoli panoge in ima lahko svoje prodajne ali poslovne enote v državah razvitega sveta (Evropa, ZDA, Kanada). Dobavitelji so lahko iz kateregakoli dela sveta. Zaposleni v podjetju so v večini slovenski državljani, kar pomeni, da se v podjetju za komunikacijo uporablja slovenski jezik.

7.1 Način ocenjevanja

Na postavljena merila sem pri vsakem ponudniku odgovoril z DA ali NE. S tem sem poskušal poenotiti odgovore ter tako narediti analizo čim bolj preprosto. Tam kjer nisem našel enoznačnega odgovora, sem dodal tudi opombo.

V primeru, da bi delal analizo izbire programja za konkretno podjetje bi zraven dodal še prioriteto vsakega posameznega merila. S tem bi določil pomembnost merila za določeno podjetje. V praksi je prioriteta posameznega merila, ne glede na to, ali je merilo strateško ali tehnično, tista, ki loči merila glede na organizacijo. Ker pa je namen te naloge bolj primerjati odprto -kodne in komercialne rešitve s področja poslovne inteligence, sem prioriteto meril izpustil. Prioriteta izbranih meril je lahko različna za vsako posamezno podjetje, tudi za podjetja znotraj iste panoge.

7.2 Primerjava po strateških merilih

Programje sem ocenjeval tako, da sem si izbral merilo ter nato preveril ponudbo pri vsakem izmed ponudnikov. Ker so strateška merila bolj splošne narave, sem odgovore na merila pridobil na spletnih straneh ponudnikov ali njihovih pooblaščenih zastopnikov. Nekaj pa tudi s pomočjo splošne dokumentacije ponudnikov, ki je na voljo preko spletnih strani.

Tabela 1: Primerjava programja s področja poslovne inteligence po strateških kriterijih

	Pentaho	JasperSoft	Spago BI	Cognos	Hyperion	Business Objects
Cena	Ni navedena	0 - 35.000 \$ ⁵	Ni navedena	Ni navedena	Ni navedena	Ni navedena
Licenca	GNU LGPL ali lastniška	GNU GPL ali GNU LGPL ali lastniška	GNU LGPL	Lastniška	Lastniška	Lastniška
Podpora pri implementaciji	DA	DA ^{2,3}	DA ¹	DA	DA	DA
Lokalna podpora	DA	NE	NE	DA	NE	DA
Podpora pri razvoju	DA ²	DA ^{2,3}	DA ¹	NE ⁴	NE ⁴	NE ⁴
Svetovanje	DA	NE	DA ¹	DA	DA	DA
Šolanje uporabnikov	DA	DA ¹	DA ¹	DA	DA	DA
Razširljivost	DA	DA	DA	NE	NE	NE

Legenda:

Št.	Komentar
1	Samo v angleškem jeziku.
2	Podpora pri implementaciji je s strani odprto-kodne skupnosti, preko foruma ter z tehnično dokumentacijo.
3	Čas odzivnosti je definiran z paketom, ki ga kupimo (od 3 do 5 dni).
4	Merilo se ne upošteva pri lastniškem programju, saj nimamo možnosti lastnega razvoja.
5	Cena 0 pomeni, da je programje prosto snemljivo z interneta. Cena 35.000\$ je za popolni paket. Programje, vzdrževanje, tehnična podpora, itd..

Vir: Lastni vir.

7.3 Rezultat analize

Ponudniki odprto-kodnega programja ponujajo dve različici svoje rešitve. Ena je odprto-kodna, ki je brezplačna in je tudi snemljiva z interneta. Druga rešitev pa je izdana pod lastniško licenco in ima bistveno več funkcionalnosti in večjo podporo s strani ponudnika kot

odprto-kodna različica. Ne glede na to, ali nam odprto-kodni ponudniki ponujajo odprto-kodno ali lastniško programje, je programje sestavljeno iz drugih odprto-kodnih sestavin, ki so v večini primerov brezplačne in so prav tako brezplačno snemljive z interneta. Vendar je potem potrebno opraviti še integracijo v lastni informacijski sistem, kar pa je običajno zelo zahtevno opravilo. Drugi trije ponudniki lastniškega programja delujejo po enakem običajnem principu. Najprej je potrebno plačati licenco za programje, nato še vzdrževalnino, ter razvoj ali prilagajanje programja posebej. Cen svojih rešitev ponudniki ne objavljajo javno, le JasperSoft ima objavljene cene za svoje sestavine. Cena je odvisna od različnih sestavin, ki jih kupec potrebuje ali želi imeti.

Pentaho ponuja programje pod odprto-kodno in lastniško licenco, pri tem pa omejuje odprto-kodno programje z njegovo funkcionalnostjo ter manjšo podporo. JasperSoft je storil še korak dlje ter popolnoma oklestil odprto-kodno različico. Tako ima rešitev veliko novih izdaj, nima nobene pravne podlage, ni je mogoče razpošiljati, ni je mogoče vgraditi v kakšno drugo rešitev, poganja se samo na odprto-kodnih strežnikih ter tudi nima naprednih funkcionalnosti kot sta poročanje in ad hoc poizvedbe. Videti je, da JasperSoft uporablja odprto-kodno skupnost le za podporo pri razvoju in pomoč pri testiranju svojega programja. Odprto-kodno ponudbo pa le kot dobro tržno potezo. Spago je v tem pogledu najboljši. Prav tako kot omenjena ponudnika nudi programje pod odprto-kodno in lastniško licenco, vendar ne omejuje funkcionalnosti in podpore kupcu. Razlika izhaja iz različnih sestavin, ki jih je mogoče vgraditi v programje. Sestavine programja so tako lahko odprto-kodne ali lastniške. Za lastniške je potrebno plačati licenčnino, odprto-kodne pa so brezplačne. Katere sestavine programja bodo izbrane, pa je odvisno od potreb in želja kupca.

Vsi ponudniki nam nudijo podporo. Večinoma pa je ta podpora v sklopu dokumentacije na voljo preko spletnih strani, forumov, ali preko pooblaščenih lokalnih podjetij. Pentaho, Cognos in Business Objects nam tako nudijo tudi vse vrste podpore preko svojih pooblaščenih podjetij v Sloveniji. V analizi sem napisal, da Hyperion ne nudi podpore v Sloveniji, ker pa je pred kratkim Oracle prevzel podjetje Hyperion, predpostavljam, da podporo za to rešitev nudi podjetje Oracle Slovenija ali kakšno drugo od Oracla pooblaščenno podjetje. Pentaho v Sloveniji zastopa mariborsko podjetje Agenda, d. o. o., ki je specializirano za odprto-kodne projekte. Cognos je za svoje zastopanje v Sloveniji izbral podjetje NPS, d. o. o. iz Ljubljane. Business Objects pa zastopajo podjetja: ITS Intertrade sistemi iz Ljubljane, SINABIT, d. o. o. iz Ljubljane ter podjetje UT informacijski sistemi, d. o. o., ki pokriva celotno srednjo ter Vzhodno Evropo in ima svoje prostore prav tako v Ljubljani. JasperSoft v Sloveniji ne nudi podpore oziroma jo nudi v primeru, da kupec plača prevoz, prenočitev in druge stroške svetovalca. Gledano z vidika stroškov pa ta način zelo podraži same stroške podpore. Enako velja tudi za Spago. Vendar je Spago usmerjen bolj lokalno, samo na italijanski trg. Vsi ponudniki odprto-kodnega programja nudijo tudi razvoj programja po želji, vendar je to potrebno dodatno plačati. Odprto-kodno programje pa imamo možnost razvijati tudi sami s svojimi razvijalci in s pomočjo tehnične dokumentacije ter razvojnih forumov.

Šolanje uporabnikov je zelo pomemben kriterij, saj je v praksi mnogokrat prav od šolanja odvisen uspeh implementacije določenega projekta. Vsi ponudniki nam nudijo šolanje uporabnikov. Od tega ga JasperSoft in Spago ponujata samo v angleškem jeziku. Spago po želji organizira tudi tečaje, vendar je zopet potrebno plačati prevoz iz Italije in druge stroške svetovalca, ki bo izvajal šolanje v Sloveniji. Ker šolanje navadno traja vsaj 3 dni, pa lahko spet nastane težava s previsokimi stroški. Drugi ponudniki, ki jih zastopajo slovenska podjetja, imajo organizirano tudi šolanje uporabnikov v slovenskem jeziku. Seveda je potrebno pri vseh ponudnikih šolanje tudi plačati. Ponudniki lastniškega programja so pred kratkim tudi že začeli organizirati šolanja uporabnikov preko interneta ali virtualna šolanja.

Pri vseh ponudnikih odprto-kodnega programja imamo tudi možnost sami izvajati razširljivost. Pri tem predpostavljam, da izberemo odprto-kodno različico programja, ki je omejena po funkcionalnosti. To različico pa nato tudi sami prilagodimo. Ponudniki lastniškega programja ne nudijo te možnosti, oziroma moramo kupiti dodatne licence za vsako na novo nameščeno programje. Stvarno gledano, se razširljivost ne izplača, saj potrebujemo tudi svoje razvijalce, poleg tega pa smo vedno omejeni po funkcionalnosti in podpori. Razen mogoče v primeru, da smo zelo veliko podjetje, v svetovnem merilu, in se lahko pogajamo za boljše pogoje pri ponudniku.

7.4 Primerjava po tehničnih merilih

Programje sem ocenjeval tako, da sem si izbral merilo, ter nato preveril ponudbo pri vsakem izmed ponudnikov. Ker se tehnična merila vežejo predvsem na operativni nivo, sem poleg uradnih spletnih strani za informacije uporabil še tehnično ter drugo dostopno dokumentacijo. Tehnična dokumentacija pri lastniškem programju ni dostopna, kar sem tudi pričakoval. Poleg tega sem pregledal tudi samo delovanje programja. Pri odprto-kodnem programju so bile na voljo za testiranje večinoma vse sestavine programja. Medtem, ko je lastniško programje na voljo samo kot demonstracijsko "demo". Tako pa je možno videti samo uporabniški grafični vmesnik in že vnaprej nastavljene podatke.

Tabela 2: Primerjava programja s področja poslovne inteligence po tehničnih merilih

	Pentaho	JasperSoft	Spago BI	Cognos	Hyperion	Business Objects
Vrsta podpore	E pošta, telefon, internet portal, forum ⁶	E pošta, telefon, internet portal, neposredni dostop, forum ⁷	E pošta, forum ⁸	E pošta, telefon, internet portal, forum, baza znanja, skupine uporabnikov ⁵	E pošta, telefon, internet portal, podpora tehnične ekipe ⁹	E pošta, telefon, internet portal, forum, podpora na kraju, ... ¹
Možnost lastnega razvoja	DA	DA	DA	NE	NE	NE
Programiranje na zahtevo	DA	NE	DA	NE	DA ¹	NE
Podpora skupnosti	DA	DA	DA	DA ⁵	DA	DA ⁵
Velikost skupnosti	15.300 1.100	NP Srednja	NP 8	NP	NP Majhna	NP
Poosebitev	DA	DA	DA	DA	DA	DA ³
Materiali za šolanje	DA	DA	DA	DA ⁵	DA	NE ⁴
Tehnična dokumentacija	DA	DA	DA	DA ⁵	DA	NE
Povezljivost	DA	DA ²	DA	NE	NE	NE

Legenda:

Št.	Komentar
1	Odvisno od vzdrževalnega paketa, ki ga plačamo pri nakupu programja.
2	Samo lastniška različica.
3	Lahko naredi kupec sam pri poročanju. Drugače je plačljiva storitev.
4	Samo tečajji v razredu ali preko interneta, ki so vsi plačljivi.
5	Na voljo samo kupcem.
6	Prvi trije so na voljo samo kot plačljiva pomoč. Pomoč je odvisna od paketa, ki ga plačamo.
7	Prvi štirje so na voljo samo kot plačljiva pomoč. Pomoč je odvisna od paketa, ki ga plačamo. Neposredni dostop pomeni, da se strokovnjak iz podjetja JasperSoft priključi na naš sistem preko interneta in odpravi težavo.
8	Možno kot lastniško ali prosto.
9	Navedene možnosti so odvisne od programa podpore, ki ga izberemo in tudi plačamo.

Vir: Lastni vir.

7.5 Rezultat analize

Pri oceni tehničnih meril za izbiro programja s področja poslovne inteligence se je pokazala največja razlika med odprto-kodnim in lastniškim programjem.

Tehnična podpora programju po tem, ko smo je že implementirali v poslovanje podjetja, se razlikuje glede na to, ali je licenca za programje odprto-kodna ali lastniška ter glede na vzdrževalni paket, ki ga plačamo pri ponudniku izbrane rešitve. Vzdrževalni paket je potrebno plačati pri vseh ponudnikih, ne glede na to ali smo implementirali odprto-kodno ali lastniško različico programja. Vzdrževalni paket se pri nekaterih ponudnikih imenuje program, paket ali še kakšen podoben izraz, povsod pa je mišljena tehnična podpora programju. Iz tabele je razvidno, da ponudniki lastniškega programja nudijo bistveno večje možnosti podpore oziroma tehnična podpora je bistveno večja pri nakupu lastniškega licence, ne glede na to ali smo jo kupili pri ponudniku odprto-kodnega ali lastniškega programja. Odprto-kodne različice programja imajo navadno le najosnovnejše vrste podpore: forum, tehnična dokumentacija, elektronska pošta in telefon. Slednja sta omejena le na obvezen odgovor s strani ponudnika, ni pa tudi definiran čas, do katerega mora ponudnik odgovoriti.

Možnost lastnega razvoja in spreminjanje izvorne kode je možna le pri odprto-kodnem programju. Vsi analizirani ponudniki odprto-kodnega programja nudijo dostop do svojih rešitev z izvorno kodo preko interneta oziroma foruma. Vendar je pri tem potrebno upoštevati dejstvo, da so glavni razvijalci programja redno zaposleni v podjetju, ki ponuja to programje. V praksi to pomeni, da bi, če bi želeli sami razviti programje in ga popolnoma prilagoditi svojim potrebam, potrebovali zelo močno razvojno ekipo. Ta ekipa pa bi morala paziti tudi na razvoj s strani razvijalcev pri ponudniku, saj bi bilo potrebno vgrajevati vse nadgradnje in

posodobitve. Glede na to, da na tem področju izjemno hitro napreduje Pentaho, je to skoraj nemogoče doseči. Tudi ob dejstvu, da se v osnovno programje neprestano vključujejo nove sestavine. Pentaho tako npr. nudi možnost programiranja na zahtevo. Kar pomeni, da njihovi razvijalci prilagodijo programje po željah kupca. Enako storitev nudijo tudi ponudniki lastniškega programja. Seveda pa je programiranje na zahtevo potrebno tudi ustrezno plačati.

Vsi ponudniki nudijo podporo s strani skupnosti. Pri odprto-kodnih skupnostih je dostop do skupnosti brezplačen, ponudniki lastniškega programja pa nudijo to podporo samo svojim kupcem. Cognos in Business Objects celo omejujeta dostop do svojih skupnosti kupcev. Hyperion pa omogoča dostop do skupnosti, vendar nam sama skupnost nič ne pomaga, če nimamo programske opreme, na kateri bi lahko izvajali prilagoditve. Razen v primeru, ko skušamo dobiti kakšno splošno informacijo ali izkušnjo kupca. Velikosti skupnosti so v tabeli ocenjene glede na Pentaho. Zgornja navedena številka je število članov skupnosti, spodnja navedena številka pa je število dejavnih članov skupnosti. Vrednost NP pa pomeni, da ni mogoče dobiti podatka o številu članov skupnosti. Oceno velikosti skupnosti sem naredil glede na primerjavo števila objav na forumih. Seveda na tistih forumih, do katerih sem imel dostop. Za vse analizirane odprto-kodne skupnosti je značilno, da se glavni razvoj vrši v razvojnih oddelkih podjetja, ki je ponudnik programja. Tako so ti razvijalci istočasno tudi moderatorji na forumih. Drugih neformalnih skupnosti za analizirane rešitve nisem našel. Edino za rešitev Crystal Reports, programje za poročanje, ki spada v sklop programja podjetja Business Objects.

Vsi ponudniki nudijo poosebitev programja. Prilagajanje in poosebitev programja spadata v skupino dodatnih storitev, ki se pri vseh ponudnikih posebej obračunavajo ne glede na vrsto licence, ki jo ima izbrano programje. Pri odprto-kodnem programju lahko tudi sami naredimo poosebitev v smislu prilagoditve grafičnih vmesnikov.

Vsa dokumentacija, tako tehnična kot druga dokumentacija, npr. materiali za šolanje, je pri ponudnikih odprto-kodnega programja prosto dostopna preko spletnih strani. Za ponudnike lastniškega programja oziroma ponudnike odprto-kodnega programja pod lastniško licenco pa načeloma velja, da je dokumentacija na voljo samo kupcem programja. Obstaja še ena bistvena razlika med lastniškim in odprto-kodnim programjem glede tehnične dokumentacije. Tehnična dokumentacija pri odprto-kodnem programju vsebuje poleg navodil za implementacijo in prilagoditev programja tudi samo zasnovo programja. S tem želijo pospešiti razvoj svojega odprto-kodnega programja. Materiali za šolanje so tudi prosto dostopni, razen pri Cognosu, ki jih omejuje le na svoje kupce ter pri Business Objects, kjer takih materialov ne uporabljajo. Pri slednjem ponudniku poskušajo prenos znanja narediti s tečaji preko interneta ali osebno v učilnici. Predpostavljam pa, da za tovrstna izobraževanja uporabljajo nekaj materialov.

Analizirano odprto-kodno programje ima možnost povezljivosti z drugim programjem. Le JasperSoft omejuje povezljivost z namestitvijo lastniške različice programja. Lastniške

različice programja pa so samostojne, in se nato povezujejo na vse mogoče vire podatkov v podjetju.

8 Sklep

Poslovna inteligenca se v podjetjih pojavlja že od samega začetka poslovanja, kljub temu, da v podjetjih ali organizacijah niti ne vedo, da jo uporabljajo ali da sploh obstoja. S tehnološkim napredkom na področju računalniških tehnologij, ki smo mu priča od začetka 90-ih let prejšnjega stoletja, pa je poslovna inteligenca postala prava panoga v računalniški industriji. Sama poslovna inteligenca načeloma nima nobene zveze z računalništvom, le to, da je razvoj tehnologije omogočil mnogo bolj obširno zbiranje in analiziranje informacij kot kdaj koli prej. Torej nam je sodobna tehnologija le pripomoček pri ustvarjanju procesa poslovne inteligence v podjetju.

Najprej moramo poslovno inteligenco v podjetju razumeti in oblikovati kot proces. Nato jo moramo prilagoditi organizaciji podjetja oziroma drugim procesom in dejavnostim. Šele nato se odločamo o izbiri ponudnikov programja s področja poslovne inteligence. Po drugi strani pa je res, da so nam velikokrat v pomoč prav ponudniki programja, saj imajo le-ti izkušnje z implementacijo programja v drugih podjetjih. Vendar je še vedno potrebno ločiti med samim programjem in poslovnim procesom.

Ponudniki programja uporabljajo različne poslovne modele s katerimi nam skušajo prodati svoje rešitve. Običajno se v literaturi ločijo na tradicionalne ali komercialne ponudnike ter ponudnike odprto-kodnega programja in rešitev. Komercialni ponudniki nam ponujajo licenčno programje, ki je plačljivo, kot kupec pa nosimo tudi vse druge stroške, ki nastajajo v zvezi s tem. Beseda odprto-kodno (*ang. open source*) programje nam vzbudi občutek, kot da bomo programje lahko dobili zastonj ter ga tako tudi prilagodili poslovanju svojega podjetja. Vendar je tu potrebna previdnost, saj ne pomeni, da je neko programje, ki je pod odprto-kodno licenco, tudi zastonj. Običajno odprto-kodni ponudniki res ponudijo svoje proizvode tudi v prosto dostopni različici, vendar je to programje funkcionalno precej omejeno in kot tako neuporabno za večino podjetij. Svetovanje in drugi stroški implementacije, razvoja in vzdrževanja programja pa niso brezplačni pri nobenem od ocenjevanih ponudnikov. Zato je pred izbiro ustreznega programja potrebna celovita analiza potreb podjetja in ponudbe ponudnika.

Pazljivi moramo biti tudi na primerjavo med podjetji. Velikokrat se v praksi dogaja, da zaposleni iz nekega podjetja vidijo določeno programje v nekem drugem podjetju, ki operativno deluje brez vsakršnih težav. Rezultati, ki jih daje programje, pa so zelo uporabni. Nato se odločijo, da bodo tudi sami implementirali takšno rešitev, vendar pri tem pozabijo, da nimajo ustrezno prilagojene organizacijske strukture, profil njihovih zaposlenih je lahko drugačen, so pa še druge težave, ki se lahko pojavijo ob takem brezglavem početju. Končni

rezultat so visoki stroški nakupa in implementacije programja, ki ne deluje, ter se zato na koncu tudi ne uporablja več. Poudaril bi rad, da je vsako podjetje zgodba zase, tudi če se primerja s konkurenčnim podjetjem v isti panogi. Zato je potrebno vedno, preden se odločimo za implementacijo določenega programja, narediti analizo koristi, ki nam jih bo to programje prineslo. Če se izkaže, da dodana vrednost ne bo dovolj velika, se za implementacijo ne odločimo.

Nadaljnji razvoj bo prinesel nove rešitve, tako na področju programja kot tudi na področju metodologij implementacije, zato pričakujem, da se bodo v prihodnosti omenjena programja precej lažje implementirala ter nato tudi uspešneje uporabljala. Je pa v veliki meri uspešnost delovanja programja odvisna od zaposlenih v podjetju.

Literatura

1. Barausse Erika, Jermol Mitja, Stevens Richard: Open Source Enterprise Resource Planning and Order Management System for Eastern European Tool and Die Making Workshops. 2006. 87 str.
2. Bitterer Andreas, Hostmann Bill, Schlegel Kurt: Magic Quadrant for Business Intelligence Platforms. Gartner. [URL: <http://mediaproducts.gartner.com/reprints/hyperion/145507.html>], 2007.
3. Cubarrubia Eydie: Open Source 20: Red Herring profiles a score of innovative startups that are championing open source. [URL: <http://www.redherring.com/Article.aspx?a=18126&hed=Open+Source+20>], 2006.
4. DeLong V. James: Interoperability: Open Source vs. Proprietary Software. 11.04.2006, 5 str.
5. Dresner Howard: A short history of Business Intelligence and Where It's Headed. [URL: http://www.hyperion.com/company/overview/thought_leadershi/exec_perspectives/history_bi.cfm], 2007.
6. Eckerson W. Wayne: Strategies and Technologies for deploying BI on an Enterprise Scale. [URL: <http://www.tdwi.org/Publications/WhatWorks/display.aspx?id=7735>], 2005.
7. Everett Dan: Open Source BI. San Mateo : Ventana Research, 2006. 12 str.
8. Ferguson Boucher Renee: Cognos Acquires Celequest to Speed BI. [URL: <http://www.eweek.com/article2/0,1895,2084914,00.asp>], 17. 01. 2007.
9. Grebenc Matic: Analiza koristi sistema poslovne inteligence. Diplomsko delo. Ljubljana : Ekonomska fakulteta, 2006. 45 str.
10. Jerič Franci: Primerjava odprtega in zaprtega programja za poslovne informacijske sisteme. Magistrsko delo. Ljubljana : Ekonomska fakulteta, 2004. 86 str.
11. Kabay E. Michael, Seymour Bosworth: Computer Security Handbook. Fourth Edition. 1228 str. [URL: <http://books.google.com/books?id=rCx5OfSFUPkC&pg=PA2&ots=aGV1f-YLVq&dq=system+security&sig=1axV2rRCE1BRAB92BkuP3bjL2Rk#PPP1,M1>], 2002.
12. Kerner Michael Sean: A New Open Source Approach to Weakness. Developer. South Darien : Jupitermedia, 2006.
13. Kerr M. Ian: Open Source. University of Ottawa: Faculty of Law. [URL: <http://www.cippic.ca/en/faqs-resources/open-source/#content>], 02. 06. 2007.
14. Legan Lucija: Pomen poslovne inteligence pri upravljanju odnosov s strankami. Diplomsko delo. Ljubljana : Ekonomska fakulteta, 2006, 47 str.
15. Loshin David: Business Intelligence. The Savvy Manager's guide. [URL: http://www.amazon.co.uk/gp/reader/1558609164/ref=sib_rdr_cr/202-6415895-7716649?ie=UTF8&p=S005&j=0#reader-page], 2003. 272 str.
16. Ohlhorst J. Frank: Value Judgment: When it comes to open-source vs. commercial software, many solution providers have already taken their stand. New York : CMP Media LLC. 2004, 4 str.

17. Tiemann Michael: Software Industry vs. Software Society: Who wins in 2020?. 2007, 3 str.
18. Torfinn Sven: Giving away secrets: can open source convert the software world?. ICT – no.5. London : Panos media. 2007, 6 str.
19. Wilcox Joe: IBM to spend \$1 billion on Linux in 2001. CNET News. [URL: <http://news.com.com/2100-1001-249750.html>], 12. 12. 2000.
20. Williams Nancy, Williams Steve: The Profit Impact of Business Intelligence. San Francisco : Morgan Kaufmann Publishers, 2007. 218 str.

Viri

1. Data Warehouse Institute. [URL: <http://www.dw-institute.com/>], 2007.
2. Free Software Foundation. [URL: <http://www.fsf.org/>], 25. 08. 2006.
3. Maurer Robert: Prevod GNU GPL Licence. [URL: <http://www.lugos.si/oss/Licence/Splosno%20dovoljenje%20GNU>], 27. 05. 2007.
4. Naude Frank: Glosary of terms. [URL: <http://www.orafaq.com/glossary/faqglosm.htm>], 25. 09. 2006.
5. Odprta koda. [URL: <http://www.odprtakoda.org>], 25. 06. 2007.
6. Odprto kodno orodje za poslovno inteligenco. Pentaho. [URL: <http://www.pentaho.com/products>], 15. 04. 2007.
7. Open Source Initiative. [URL: <http://opensource.org/docs/osd>], 25. 08. 2006.
8. Open Source Software Ecosystem. IDC. [URL: <http://www.idc.com/prodserv/maps/opensource.jsp>], 13. 6. 2007.
9. Opis podjetja. BusinessObjects. [URL: http://www.businessobjects.com/company/default.asp?intcmp=ip_company1], 17. 04. 2007. a.
10. Opis podjetja. Hyperion. [URL: <http://www.hyperion.com/company/overview/>], 17. 04. 2007. b.
11. Opis podjetja. JasperSoft. [URL: http://www.jaspersoft.com/co_overview.html], 15. 04. 2007. c.
12. Opis podjetja. Spago. [URL: http://www.eng.it/chisiamo_e/1_0.htm], 15. 04. 2007. c.
13. Opis produktov. BusinessObjects. [URL: http://www.businessobjects.com/products/default.asp?intcmp=ip_products1], 17. 04. 2007. a.
14. Opis produktov. Cognos. [URL: <http://www.cognos.com/products/cognos8businessintelligence/index.html>], 17. 04. 2007. b.
15. Opis produktov. JasperSoft. [URL: http://www.jaspersoft.com/re_downloadpage.html], 15. 04. 2007. c.

16. Opis produktov. Spago.
[URL: <http://spago.eng.it/ecm/faces/public/guest/home/solutions/spagobi>], 15. 04. 2007.
d.
17. Pentaho Open Source Business Intelligence Platform Technical White Paper. 20 str.
[URL:
http://ovh.dl.sourceforge.net/sourceforge/pentaho/Pentaho_Technical_Whitepaper_1-6.pdf], 2007.
18. Pingo Linux. [URL: <http://www.pingo.org/dok/odprta-koda/vse-o-ok.shtml>], 21. 10. 2004.
19. Seznam kriterijev za izbiro ponudnika informacijskega sistema. Agregata, d. o. o., 30. 04. 2007.
20. Slovar informatike. [URL: http://www.islovar.org/iskanje_enostavno.asp], 04. 05. 2007.
21. Splošne informacije. JasperSoft. [URL: <http://www.jasperforge.org/>], 15. 04. 2007. a.
22. Splošne informacije. Pentaho. [URL: <http://pentaho.org/about>], 15. 04. 2007. b.
23. Why you shouldn't use the Lesser GPL for your next library. Boston: Free Software Foundation, 2007.
24. Wikipedia. [URL: <http://en.wikipedia.org/wiki>], 25. 07. 2007.

Priloga

To je neuradni prevod Splošnega dovoljenja GNU GPL (angl. GNU General Public License) v slovenščino. Ni ga objavila ustanova Free Software Foundation in zakonsko ne določa pogojev za razširjanje programov, ki uporabljajo GNU GPL. Te določa le angleški izvirnik licence GNU GPL ali njegovi uradni prevodi, ki so objavljeni na spletnih straneh Free Software Foundation. Uradni prevod v slovenski jezik zaenkrat še ne obstaja.

GNU GPL licenca

Različica št. 2, junij 1991

Pravice razširjanja © 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Vsakdo sme razmnoževati in razširjati dobesedne kopije tega licenčnega dokumenta, ni pa ga dovoljeno spreminjati

Predgovor

Licenčne pogodbe večine programja so zasnovane tako, da vam preprečujejo njegovo svobodno razdeljevanje in spreminjanje. Za razliko od teh vam namerava Splošno dovoljenje GNU (angl. GNU General Public License, GPL) zajamčiti svobodo pri razdeljevanju in spreminjanju prostega programja ter s tem zagotoviti, da ostane programje prosto za vse njegove uporabnike. Ta GPL se nanaša na večino programske opreme ustanove Free Software Foundation in na vse druge programe, katerih avtorji so se zavezali k njeni uporabi. (Nekatero drugo programje ustanove Free Software Foundation je namesto tega pokrito s Splošnim dovoljenjem GNU za knjižnice, angl. GNU Library General Public License.) Uporabite jo lahko tudi za vaše programe.

Ko govorimo o prostem programju, imamo s tem v mislih svobodo, ne cene. Naša splošna dovoljenja GNU vam zagotavljajo, da imate pravico razširjati kopije prostega programja (in zaračunavati za to storitev, če tako želite); da dobite izvorno kodo ali jo lahko dobite, če tako želite; da lahko spreminjate programje ali uporabljate njegove dele v novih prostih programih; in da veste, da lahko počnete vse te stvari.

Zaradi zavarovanja vaših pravic moramo uvesti omejitve, ki prepovedujejo vsakomur, da bi vam te pravice kratil ali od vas zahteval predajo teh pravic. Te omejitve se preslikajo v določene odgovornosti za vas, če razširjate kopije programja ali ga spreminjate.

Na primer, če razširjate kopije takega programa, bodisi zastonj ali za plačilo, morate dati prejemnikom vse pravice, ki jih imate vi. Prepričati se morate, da bodo tudi oni prejeli ali

imeli dostop do izvorne kode. In morate jim pokazati te pogoje (pravzaprav izvirnik, opomba prevajalca), da bodo poznali svoje pravice.

Vaše pravice varujemo z dvema korakoma: (1) s pravno zaščito programja in (2) ponujamo vam to licenco, ki vam daje pravno dovoljenje za razmnoževanje, razširjanje in/ali spreminjanje programja.

Zaradi zaščite vsakega avtorja in zaradi naše zaščite želimo zagotoviti, da vsakdo razume, da za to prosto programje ni nobenega jamstva. Če je programje spremenil nekdo drug in ga posredoval naprej, želimo, da njegovi prejemniki vedo, da to, kar imajo, ni izvirnik, zato da se problemi, ki jih povzročijo drugi, ne bodo odražali na ugledu izvornega avtorja.

Končno, vsakemu prostemu programu nenehno grozijo programski patenti. Želimo se izogniti nevarnosti, da bi razširjevalci prostega programa posamično dobivali patentne licence in s tem naredili program lastniški (angl. proprietary). Za preprečitev tega jasno zahtevamo da mora biti vsak patent licenciran tako, da ga lahko vsakdo prosto uporablja, ali pa sploh ne sme biti licenciran.

Sledijo natančne določitve in pogoji za razmnoževanje, razširjanje in spreminjanje.

DOLOČITVE IN POGOJI ZA RAZMNOŽEVANJE, RAZŠIRJANJE IN SPREMINJANJE

0. Licenca se nanaša na vsak program ali drugo delo, ki vsebuje obvestilo lastnika avtorskih pravic (angl. copyright holder) z izjavo, da se lahko distribuira pod pogoji Splošnega dovoljenja GNU (angl. General Public License). „Program“ se v nadaljevanju nanaša na vsak tak program ali delo, in „delo, ki temelji na programu“ pomeni bodisi program ali pa katerokoli izvedeno delo po zakonu o avtorskih pravicah (angl. copyright law): se pravi delo, ki vsebuje program ali njegov del, bodisi dobesedno ali s spremembami in/ali prevedeno v drug jezik. (Tukaj in povsod v nadaljevanju je prevod vključen brez omejitev v pojem „spremembe“.) Vsaka licenca je naslovljena na „vas“.

Ta licenca ne pokriva nobenih drugih aktivnosti razen razmnoževanja, razširjanja in sprememb; ostale so izven njenega dometa. Dejanje poganjanja programa ni omejeno in izhod programa je zajet le, če njegova vsebina sestavlja delo, iz katerega je izpeljan program (ne glede na to, da je bil narejen s poganjanjem programa). Ali je to res ali ne, je odvisno od tega, kaj počne program.

1. Razmnožujete in razširjate lahko dobesedne izvode izvorne kode programa v enaki obliki, kot jo dobite, preko kateregakoli medija, če le na vsakem izvodu razločno in primerno objavite obvestilo o pravicah razširjanja in zanikanje jamstva; vsa obvestila, ki se nanašajo na to licenco in odsotnost vsakršnega jamstva pustite nedotaknjena; in daste vsem drugim prejemnikom programa poleg programa še izvod te licence.

Za fizično dejanje prenosa kopije lahko zaračunavate in po vaši presoji lahko ponudite garancijsko zaščito v zameno za plačilo.

2. Spreminjati smete vaš izvod ali izvode programa ali katerikoli njegov del, in tako narediti delo, ki temelji na programu, ter razmnoževati in razširjati takšne spremembe ali dela pod pogoji zgornjega razdelka 1, če zadostite tudi vsem naslednjim pogojem:

a. Zagotoviti morate, da spremenjene datoteke nosijo vidna obvestila o tem, da ste jih spremenili in datum vsake spremembe.

b. Zagotoviti morate, da je vsako delo, ki ga razširjate ali izdajate in ki v celoti ali deloma vsebuje program ali katerikoli njegov del ali pa je iz njega izpeljano, licencirano pod pogoji te licence kot celota brez plačila katerikoli tretji osebi.

c. Če spremenjeni program ob zagonu navadno bere ukaze interaktivno, morate zagotoviti, da se ob najbolj običajnem zagonu za takšno interaktivno uporabo izpiše ali prikaže najava, ki vključuje primerno sporočilo o pravicah razširjanja in sporočilo, da jamstvo ni zagotovljeno (ali pa sporočilo, da ponujate jamstvo) in da lahko uporabniki razširjajo program pod temi pogoji, in pove uporabniku, kako pogledati izvod te licence. (Izjema: če je sam program interaktiven, a navadno ne izpiše takšne najave, tudi za vaše delo, ki temelji na programu, ni nujno, da jo.)

Te zahteve se nanašajo na spremenjeno delo kot celoto. Če kosi tega dela, ki jih je lahko prepoznati, niso izpeljani iz programa in se jih lahko ima za neodvisna in ločena dela sama po sebi, potem ta licenca in njeni pogoji ne veljajo zanje, kadar jih razširjate ločeno. Vendar, kadar te iste kose razširjate kot del celote, ki je delo, ki temelji na programu, mora biti razširjanje celote izvedeno pod pogoji te licence, katere dovoljenja za druge licence se razširjajo na vso celoto in torej na vsak njen del, ne glede na to, kdo ga je napisal.

Torej, namen tega razdelka ni, da bi zanikal ali spodbijal vaše pravice do dela, ki ste ga v celoti napisali sami; namesto tega je namen razširiti pravico do nadzora razširjanja na izpeljana ali zbrana dela, ki temeljijo na programu.

Poleg tega, če gre za zgolj kopičenje drugega dela, ki ne temelji na programu, s programom (ali z delom, ki temelji na programu) na mediju za shranjevanje ali distribucijskem mediju, se licenca na to drugo delo ne nanaša.

3. Program (ali delo, ki temelji na njem, pod razdelkom 2) lahko razmnožujete in razširjate v objektni kodi ali izvedljivi obliki pod pogoji zgornjih razdelkov 1 in 2, če izpolnite tudi kaj od tega:

a. Opremite ga s popolno in ustrezno izvorno kodo v strojno berljivi obliki, ki mora biti razširjana pod pogoji zgornjih razdelkov 1 in 2 na mediju, ki se navadno uporablja za izmenjavo programja; ali,

b. Opremite ga z napisano ponudbo, veljavno vsaj tri leta, da boste katerikoli tretji osebi, za plačilo, ki ne bo presegalo vaših stroškov fizičnega izvajanja izvorne distribucije, dali popoln izvod ustrezne izvorne kode v strojno berljivi obliki, ki bo razširjana pod pogoji zgornjih razdelkov 1 in 2 na mediju, ki se običajno uporablja za izmenjavo programja; ali,

c. Opremite ga z informacijo, ki ste jo dobili vi, kot ponudbo distribucije ustrezne izvorne kode. (Ta alternativa je dovoljena le za nekomercialne distribucije in le, če ste dobili program v obliki izvorne kode ali izvedljivi obliki s takšno ponudbo, glede na podrazdelek b, zgoraj.)

Izvorna koda pri delih pomeni obliko dela, najprimernejšo za izdelavo sprememb. Pri izvedljivem delu pomeni izvorna koda vso izvorno kodo za vse module, ki jih vsebuje, poleg tega pa še morebitne datoteke z definicijami vmesnika, povezane s tem delom in skripte, uporabljane za nadzor prevajanja in namestitvev izvedljive datoteke. Vendar - kot posebna izjema - ni nujno, da razširjana izvorna koda vključuje vse, kar se navadno razširja (v izvorni ali binarni obliki) z večjimi komponentami (prevajalnik, jedro, in tako naprej) operacijskega sistema, na katerem teče izvedljiva datoteka, razen če ta komponenta spremlja izvedljivo datoteko.

Če se razširjanje izvedljive datoteke ali objektne kode izvede s ponujenim dostopom za prepisovanje z za to namenjenega mesta, potem ponujanje ekvivalentnega dostopa za razmnoževanje izvorne kode z istega mesta šteje kot razširjanje izvorne kode, čeprav tretje osebe niso prisiljene razmnoževati izvorne kode poleg objektne kode.

4. Ne smete razmnoževati, spreminjati, podlicencirati ali razširjati programa drugače, kot to izrecno določa pričujoča licenca. Vsak poskus siceršnjega kopiranja, spreminjanja, podlicenciranja ali razširjanja programa je ničen in bo samodejno prekinil vaše pravice pod to licenco. Vendar pa se osebam, ki so svoj izvod ali pravice dobile od vas pod to licenco, licenca ne prekine, dokler se ji popolnoma podrejajo.

5. Ni vam treba sprejeti te licence, saj je niste podpisali. Vendar vam razen nje nič ne dovoljuje spreminjanja ali razširjanja programa ali iz njega izpeljanih del. Če ne sprejmete te licence, ta dejanja prepoveduje zakon. Torej, s spremembo ali razširjanjem programa (ali kateregakoli dela, ki temelji na programu), pokažete svoje strinjanje s to licenco in z vsemi njenimi določitvami in pogoji za razmnoževanje, razširjanje ali spreminjanje programa ali del, ki temeljijo na njem.

6. Vsakič, ko razširjate program (ali katerokoli delo, ki temelji na programu), prejemnik samodejno prejme licenco od izvirnega izdajatelja licence (angl. original licensor) za razmnoževanje, razširjanje ali spreminjanje programa glede na ta določila in pogoje. Ne smete vsiljevati nobenih nadaljnjih omejitev izvajanja prejemnikovih pravic, podeljenih tukaj. Niste odgovorni za vsiljevanje strinjanja tretjih oseb s to licenco.

7. Če so vam, kot posledica presoje sodišča ali suma kršitve patenta ali zaradi kateregakoli drugega razloga (ne omejenega zgolj na patentna vprašanja), vsiljeni pogoji (bodisi z odlokom sodišča, sporazumom ali drugače), ki nasprotujejo pogojem te licence, vas ne odvezujejo pogojev te licence. Če programa ne morete razširjati tako, da hkrati zadostite svojim obvezam pod to licenco in katerimkoli drugim pristojnim obvezam, potem posledično sploh ne smete razširjati programa. Na primer, če patentna licenca ne dovoli razširjanja programa brez plačevanja avtorskega honorarja vseh, ki prejmejo kopije neposredno ali posredno od vas, potem je edina možna pot, da zadostite temu pogoju in tej licenci ta, da se v celoti vzdržite razširjanja programa.

Če se za katerikoli del tega razdelka ugotovi, da je neveljaven ali da se ga ne da izvajati pod kateremkoli določenim pogojem, je mišljeno, da velja usmeritev tega razdelka (angl. balance of the section) in razdelek kot celota velja v drugih primerih.

Namen tega razdelka ni, da bi vas napeljeval h kršitvi patentov ali drugih trditev lastništva pravic ali izpodbijal veljavnost katerihkoli takšnih trditev; edini namen tega razdelka je ščitenje integritete sistema distribucije prostega programja, ki je izveden s prakso javnih licenc. Mnogi ljudje so radodarno prispevali k širokemu naboru programja, razširjanega skozi ta sistem, v upanju na njegovo dosledno izvajanje; od avtorja/dajalca je odvisno, če je pripravljen razširjati programje skozi katerikoli drug sistem, in izdajatelj licence ne more vsiljevati te izbire.

Ta razdelek namerava temeljito pojasniti, kaj so predvidene posledice nadaljevanja licence.

8. Če sta razširjanje in/ali uporaba programa omejena v določenih državah, bodisi zaradi patentov ali vmesnikov s posebno pravico razširjanja (angl. copyrighted interfaces), lahko izvorni lastnik ali lastnica pravic razširjanja, ki postavlja program pod to licenco, doda eksplicitno zemljepisno omejitev razširjanja, ki izključuje te države, tako da je razširjanje dovoljeno le v in med državami, ki niso na tak način izključene. V takem primeru ta licenca vključuje omejitve, kot da so napisane v telesu te licence.

9. Ustanova Free Software Foundation lahko od časa do časa izdaja preurejene in/ali nove različice Splošne javne licence (angl. General Public License). Nove različice bodo pisane v duhu trenutne različice, vendar se lahko razlikujejo v podrobnostih, ki bodo obdelovale nove težave ali poglede.

Vsaki različici je prirejena razločevalna številka različice. Če program določa številko različice te licence, ki se nanaša na njo in „na katerekoli poznejše različice“, imate izbiro upoštevanja pogojev in določil bodisi te različice ali katerekoli poznejše različice, ki jo je izdala ustanova Free Software Foundation. Če program ne določa številke različice te licence, lahko izberete katerokoli različico, ki jo je kdajkoli izdala ustanova Free Software Foundation.

10. Če želite vključiti dele programa v druge proste programe, katerih pogoji razširjanja so drugačni, pišite avtorju in ga prosite za dovoljenje. Za programje, katerega pravice razširjanja ima Free Software Foundation, pišite na Free Software Foundation; včasih naredimo izjemo pri tem. Našo odločitev bosta vodila dva cilja: ohranitev prostega statusa vseh izvedenih del iz našega prostega programja in spodbujanje razdeljevanja in ponovne uporabe programja na splošno.

BREZ JAMSTVA

11. KER JE PROGRAM LICENCIRAN KOT BREZPLAČEN, NI NOBENEGA JAMSTVA ZA PROGRAM DO MEJE, KI JO DOLOČA PRISTOJNI ZAKON. RAZEN, ČE NI DRUGAČE NAPISANO, IMETNIKI PRAVIC RAZŠIRJANJA IN/ALI DRUGE OSEBE PONUJAJO PROGRAM „TAK, KOT JE“, BREZ ZAGOTOVILA KAKRŠNEKOLI VRSTE, NEPOSREDNEGA ALI POSREDNEGA, KAR VKLJUČUJE, A NI OMEJENO NA POSREDNA JAMSTVA CENOVNE VREDNOSTI IN PRIMERNOSTI ZA DOLOČENO UPORABO. CELOTNO TVEGANJE GLEDE KAKOVOSTI IN DELOVANJA PROGRAMA PREVZAMETE SAMI. ČE SE PROGRAM IZKAŽE ZA OKVARJENEGA, SAMI NOSITE STROŠKE VSEH POTREBNIH STORITEV, POPRAVIL ALI POPRAVKOV.

12. V NOBENEM PRIMERU, RAZEN ČE TAKO PRAVI VELJAVNI ZAKON ALI JE PISNO DOGOVORJENO, NE BO LASTNIK PRAVIC RAZŠIRJANJA ALI KATERAKOLI DRUGA OSEBA, KI LAHKO SPREMENI IN/ALI PONOVRNO RAZŠIRJA PROGRAM, KOT JE DOVOLJENO ZGORAJ, PREVZEL ODGOVORNOSTI ZARADI ŠKODE, NAJSI GRE ZA SPLOŠNO, POSEBNO, NENAMERNO ŠKODO ALI ŠKODO, IZHAJAJOČO IZ UPORABE ALI NEZMOŽNOSTI UPORABE PROGRAMA (VKLJUČNO Z, A NE OMEJENO NA, IZGUBO PODATKOV ALI NENATANČNO OBDELAVO PODATKOV ALI IZGUBO, POVZROČENO VAM ALI TRETJIM OSEBAM ALI NEZMOŽNOST PROGRAMA, DA BI DELOVAL S KAKIM DRUGIM PROGRAMOM), ČETUDI JE BIL TAK LASTNIK ALI DRUGA OSEBA OBVEŠČEN O MOŽNOSTI NASTANKA TAKŠNE ŠKODE.

KONEC DOLOČB IN POGOJEV

Kako upoštevati te določbe pri vaših novih programih

Če razvijate nov program in želite, da bi bil kar najbolj uporaben za javnost, je najboljši način, da to dosežete, ta, da ga proglasite za prosto programje, ki ga lahko vsakdo razširja in spreminja pod zgornjimi pogoji.

Za kaj takega vašemu programu pripnite naslednja sporočila (pravzaprav, pripnite izvorna angleška sporočila, tako jih bo razumelo več ljudi). Najvarneje je, da jih pripnete na začetek vsake izvorne datoteke, saj boste tako najbolj učinkovito izrazili odsotnost jamstva; vsaka datoteka pa bi morala vsebovati tudi vrstico „o pravicah razširjanja“ (angl. copyright) in kazalec, kje najti polno sporočilo.

Vrstica, v kateri podate ime programa in kratek opis, kaj počne.

Copyright (C) 1991/2000 ime avtorja

Ta program spada med prosto programje; lahko ga razširjate in/ali spreminjate pod pogoji splošnega dovoljenja GNU (GNU General Public License), kot ga je objavila ustanova Free Software Foundation.; bodisi različice 2 ali (po vaši izbiri) katerekoli poznejše različice.

Ta program se razširja v upanju da bo uporaben, vendar BREZ VSAKRŠNEGA JAMSTVA; tudi brez posredne zagotovitve CENOVNE VREDNOSTI ali PRIMERNOSTI ZA DOLOČEN NAMEN. Za podrobnosti glejte besedilo GNU General Public License.

Skupaj s tem programom bi morali prejeti izvod splošnega dovoljenja GNU (GNU General Public License); če ga niste, pišite na Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

Dodajte tudi informacije o tem, kako stopiti v stik z vami po elektronski ali papirni pošti.

Če je program interaktiven, poskrbite, da bo izpisal kratko obvestilo o tem, ko se zažene v interaktivnem načinu:

Gnomovizija različica 69, Copyright (C) 1991/2000 ime avtorja

Gnomovizija je BREZ VSAKEGA JAMSTVA; za podrobnosti napišite „prikaži j“.

To je prosto programje in brez zadržkov ga lahko razširjate pod določenimi pogoji; napišite „prikaži p“ za podrobnosti.

Hipotetična ukaza „prikaži j“ in „prikaži p“ bi morala prikazati ustrezne dele Splošnega dovoljenja GNU. Seveda bodo ukazi, ki jih boste uporabljali, morda poimenovani drugače kot „prikaži j“ in „prikaži p“; morda bodo celo kliki z miško na menujske postavke - kar je pač primerno za vaš program.

Pridobiti bi morali tudi vašega delodajalca (če delate kot programer) ali vašo šolo, če jo obiskujete, da napiše „odrekanje pravicam razširjanja“ za program, če je to potrebno. Tukaj je primer, imena prilagodite:

Jojodin, d. o. o., se s tem dokumentom odreka vsakemu plačilu od pravic razširjanja za program `Gnomovizija' (ki namiguje prevajalnikom), ki ga je napisal Janez Hekeršek.

podpis Tineta Bogataja, 1. april 1999

Tine Bogataj, predsednik vic

Splošno dovoljenje GNU ne dovoljuje vključevanja vašega programa v lastniške programe. Če je vaš program knjižnica podprogramov, se vam bo morda zdelo bolj uporabno, da dovolite povezovanje lastniških aplikacij s knjižnico. Če je to tisto, kar želite storiti, uporabite Splošno dovoljenje GNU za knjižnice (angl. GNU Library General Public License) namesto te licence.

Slovar izrazov

Tuj izraz	Kratica	Slovenski prevod
Balanced scorecard	BSC	Uravnoteženi sistem kazalnikov
Business intelligence	BI	Poslovna inteligenca
Business logic		Poslovna logika
Customer Relationship Management	CRM	Menedžment odnosov s kupci
Dashboard		Digitalna delovna tabla
Data manipulation language	DML	Jezik za manipulacijo s podatki
Data warehouse		Podatkovno skladišče
Data Warehouse Institute	DWI	
Download		Snemanje
Enterprise Information Integration	EII	Integracija informacij v podjetju
Enterprise Information Management	EIM	Menedžment informacij v podjetju
Enterprise Performance Management	EPM	Menedžment učinkovitosti delovanja podjetja
Enterprise Resource Planning	ERP	Planiranje virov podjetja
Extensible Markup Language	XML	Razširljivi označevalni jezik
Extract Transform Load	ETL	Izveči, preoblikuj, naloži
Executive information systems	EIS	Direktorski informacijski sistem
Free software	FS	Prosto programje
Free Software Foundation	FSF	
Freeware		Zastonjsko programje
General Public License	GPL	
Gnu's Not Unix	GNU	
Key process indicator	KPI	Ključni procesni indikator
Library or "Lesser" General Public License	LGPL	
Megabyte	MB	Megabit
Metadata		Meta podatki
Online Analytical Processing	OLAP	Sprotna analitična obdelava podatkov
Open source	OS	Odprta koda
Open Source Initiative	OSI	Odprto-kodna iniciativa

Tuj izraz	Kratica	Slovenski prevod
Open source software	OSS	Odprto-kodno programje
Response time		Čas vračila informacije
Query by example	QBE	Poizvedba po primeru
Server		Strežnik
Shareware		Preizkusno programje
Terabyte	TB	Terabit
Total cost of ownership	TCO	Celotni stroški lastništva
Workflow		Procesni tok
World trade organization	WTO	Svetovna trgovinska organizacija