

**UNIVERZA V LJUBLJANI
FAKULTETA ZA UPRAVO**

**Diplomsko delo
univerzitetnega programa**

**ODPRTO KODNE PROGRAMSKE REŠITVE IN
NJIHOVO LICENCIRANJE**

Sašo Zupanec

Ljubljana, september 2009

**UNIVERZA V LJUBLJANI
FAKULTETA ZA UPRAVO**

**Diplomsko delo
univerzitetnega programa**

**ODPRTO KODNE PROGRAMSKE REŠITVE IN NJIHOVO
LICENCIRANJE**

Kandidat: Šašo Zupanec
 Številka indeksa: 04034351

Mentor: doc. dr. Ljupčo Todorovski

Ljubljana, september 2009

POVZETEK

Odrpta koda je razvojna metodologija, ki omogoča vpogled v izvorno kodo programske opreme ali nekega drugega produkta. V današnjem času se je odrpta koda razvile do te stopnje, da služi kot podlaga večjim resničnim sistemom. Problem nastane v deljenju pridobljenega znanja na tem področju, ki bi omogočil še hitrejši razvoj odrpte kode, saj znanje in izkušnje iz področja niso prosto deljene in avtorji pogosto naletijo na težave pod katero licenco objaviti publikacije in izvorne kode.

Namen te naloge je torej najprej predstaviti koncept odrpte kode. Predstavitev vključuje pregled zgodovinskega razvoja koncepta odrpte kode od začetka GNU projekta pa vse do danes, prednosti in slabosti odrpte kode in nekaterih vrste odrpto kodne programske opreme. Drugi del diplomskega dela bo pa posvečen predstavitvi odrpto kodne skupnosti in podrobnejšemu pregledu licenc, ki omejujejo avtorje in uporabnike. V zaključnem delu bom raziskal razlike med samimi licencami in naredil primerjave med bolj pogosto uporabljenimi licencami.

Ključne besede: odrpta koda, odrpto kodna programska oprema, izvorna koda, licenca, odrpto kodne licence.

SUMMARY

Open Source is a development methodology that provides access to the source code of a software (or other type of) product. Recently open source products have developed to a level, which enables its use as a base for large-scale systems. A problem of open source, which is in focus of this thesis, is in the process of sharing the knowledge through open source solutions, which is crucial for faster development of open source. This is due to the fact that knowledge and experience from the areas of science are not always shared openly and authors are commonly faced with the issue of choosing the license for publishing a certain solution and its source code.

The purpose of this diploma is first to present the concept of open source. The presentation includes historical development of the open source concept from the beginnings, rooted within the GNU project, up to present time, advantages and disadvantages of open source solutions, and some commonly used categories of open source software. Second part of the thesis is focused on presenting open source community, while the last, most significant contributions, presents a variety of commonly used open source licenses that limit authors and users and provides a comparative analysis thereof.

Key words: open source, open source software, source code, license, open source license

KAZALO VSEBINE

POVZETEK.....	ii
SUMMARY.....	iii
1 UVOD.....	1
1.1 OPREDELITEV PROBLEMA	1
1.2 NAMEN IN CILJI DIPLOMSKE NALOGE	2
2 RAZVOJ ODPRTE KODE.....	3
2.1 PROSTO PROGRAMJE IN ODPRTA KODA.....	3
2.1.1 Prosto programje	3
2.1.2 Odprta koda.....	5
2.2 ZGODOVINA IN RAZVOJ ODPRTE KODE.....	9
2.2.1 UNIX	9
2.2.2 GNU	10
2.2.3 Linux.....	11
2.2.4 Apache	12
2.2.5 Odprta koda danes.....	12
2.2.6 Odprta koda v Sloveniji.....	13
3 ZNAČILNOSTI ODPRTE KODE.....	15
3.1 PREDNOSTI ODPRTE KODE	15
3.1.1 Zanesljivost.....	15
3.1.2 Stabilnost	16
3.1.3 Preglednost.....	16
3.1.4 Stroški	16
3.1.5 Prilagodljivost in svoboda.....	17
3.1.6 Podpora in odgovornost.....	17
3.2 SLABOSTI ODPRTO KODNE PROGRAMSKE OPREME.....	18
3.3 ODPRTA KODA IN ZNANOST	20
3.3.1 Prost dostop	21
3.3.2 Vpliv odprte kode na strojno učenje	22
3.4 ODPRTO KODNE SKUPNOSTI	24
3.4.1 Trg.....	25

3.4.2	Značilnosti pripadnikov skupnosti	26
3.4.3	Motivacija	27
4	LICENCIRANJE ODPRTE KODE	29
4.1	AVTORSKE PRAVICE IN IZPELJANA DELA	29
4.2	ODPRTO KODNE LICENCE	30
4.3	KATEGORIJE ODPRTO KODNIH LICENC	32
4.3.1	Licence kategorije A	33
4.3.2	Licence kategorije B	34
4.3.3	Licence kategorije C	35
4.4	PREGLED ODPRTO KODNIH LICENC	36
4.4.1	Gnu general public license (GPL)	36
4.4.2	Lgpl lesser general public license (LGPL).....	37
4.4.3	Barkeley software distribution (BSD)	37
4.4.4	Mozilla public license (MPL).....	38
4.4.5	EUPL v1.0 (European Union Public Licence)	38
4.5	DRUGE ODPRTO KODNE LICENCE	39
4.6	PRIMERJAVA LICENC	41
4.6.1	Skupne lastnosti licenc:	41
4.6.2	Recipročna obveznost (Reciprocal Obligation)	42
4.6.3	Izbor licence	43
4.6.4	Težave pri odprto kodnem licenciranju.....	45
4.6.4.1	Spor zaradi licenc (SCO vs. IBM).....	46
5	ZAKLJUČEK	48
	LITERATURA.....	50
	IZJAVA O AVTORSTVU IN NAVEDBA LEKTORJA	53

1 UVOD

1.1 OPREDELITEV PROBLEMA

Odprto kodna programska oprema sega v same začetke razvoja računalništva in programske opreme in je danes vse bolj konkurenčna licenčni programski opremi. Kar loči odprto kodne programske opremo od licenčne programske opreme je to da je vpogled v izvorno kodo dostopen in možen vsem. Za uporabo ali spremembo te kode ni potrebno plačevati posebnih licenc ali raznih drugih avtorskih pravic. Kljub temu pa še vedno obstajajo licenčna pravila uporabe odprte kode.

Odprta koda je pristop, ki ponuja praktično dostopnost do izvorne kode programske opreme ali drugega produkta. Odprto kodo nekateri dojemajo kot enega izmed mnogih možnih pristopov načrtovanja, drugi pa jo smatrajo kot kritičen strateški element svojega delovanja. Izraz odprta koda je postal priljubljen z vzponom interneta, ki je omogočil dostop do različnih modelov produkcije, komunikacijskih poti in interaktivnih skupnosti. Odprto kodni model delovanja in sprejemanja odločitev omogoča hkraten vnos dela, pristopov in prioritet ter se razlikuje od bolj zaprtih, centraliziranih modelov razvoja. Principi in prakse se po navadi nanašajo na razvijanje izvorne kode programov, ki so razpoložljivi za javno sodelovanje.

Začetek sodelovanja pri razvoju odprto kodne programske opreme sega v čas nastanka interneta; v današnjem času sta internet in odprta koda tesno povezani med seboj. Internet na začetku ni bil dostopen za komercialno rabo in šele leta 1991 se je odprl tudi uporabnikom s profitnimi motivi. To je lahko tudi opravičilo za ponudnike lastniške programske opreme, zakaj niso igrali nobene vloge pri njegovem nastajanju. Vendar že petnajst let izdatno izkoriščajo to skupno dobro, bistvo interneta pa še vedno tvori pet projektov, ki vsi sodijo v odprto kodno programske opremo: Linux, BIND, Sendmail, Perl in Apache (Pivec, 2006).

Odprta koda ima lahko prav tako velik vpliv tudi v različnih znanstvenih disciplinah. Prav prevzem odprto kodne miselnosti in modela prostega deljenja izvornih podatkov lahko znanstvenikom omogoči prost dostop do predhodnih raziskav in poskusov na njihovem področju in jim lahko močno pomaga pri razvoju novih rešitev. Znanstvenikom in avtorjem lahko odprto kodni model olajša delo predvsem s tem, da jim ni treba ponovno izvajati prejšnjih poskusov in ponovno implementirati že znane postopke.

1.2 NAMEN IN CILJI DIPLOMSKE NALOGE

Namen mojega diplomskega dela je torej predstaviti odprto kodno programsko opremo, predstaviti odprto kodne programske rešitve in si boljše ogledati avtorske pravice in licenciranje odprto kodne programske opreme. Cilj moje diplomske naloge je s pomočjo raziskave organizacij in licenc, ki se ukvarjajo z odprto kodo postaviti odprto kodne rešitve v nek pravni okvir in preučiti namen nekaterih večjih licenc ter narediti primerjavo med njimi.

Diplomsko delo bi bilo torej sestavljeno iz treh vsebinskih poglavij. Drugo je sestavljeno torej iz opredelitve kaj odprta koda sploh je, kakšen je bil razvoj in zgodovina odprte kode in kakšne vrste odprto kodnih programskih rešitev poznamo. V tretjem poglavju se bom poglobil v opisovanje razlik med licenčno programsko opremo in odprto kodno programsko opremo ter prednosti ene ali druge. V tem poglavju diplomskega dela bo pa boljše predstavljena odprto kodna skupnost in pregledal faktorje, ki jo ženejo naprej in motivirajo. V četrtem poglavju boljše analiziram odprto kodne licence (kot so splošna GNU licenca in podobne), ki urejajo pravila uporabe in nadaljnega razvoja odprto kodnih rešitev ter primerjam poglavne vidike posameznih licenc.

2 RAZVOJ ODPRTE KODE

2.1 PROSTO PROGRAMJE IN ODPRTA KODA

Izraz prosto programje je treba razlikovati od izraza odprta koda. Vendar imata s strani izvirne kode oba skupno to da mora biti izvorna koda za oba izraza razpoložljiva na vpogled, spreminjanje in nadaljnjo distribucijo. Slika 1 prikazuje različne kategorije programja, ki se pogosto omenjajo v razpravah o prostem programju. Slika kaže katere kategorije se prekrivajo ali so del drugih kategorij.

2.1.1 Prosto programje

Prosto programje je programska oprema, ki se lahko uporablja, pregleduje in spreminja brez omejitev in ki se lahko kopira in distribuira v spremenjeni ali nespremenjeni obliki brez omejitev ali z minimalnimi omejitvami. Te omejitve zagotavljajo da nadaljnji prejemniki lahko prav tako izvajajo prej omenjene postopke ter da proizvajalci strojne opreme omogočijo prilagoditev njihove strojne opreme uporabnikom. Prosto programje je v večini primerov dosegljivo brez plačila. To v praksi pomeni, da če nekdo želi distribuirati neko programsko opremo kot prosto programje mora izdati tudi človeku jasno obliko programa tako imenovano izvorno kodo vključno z klavzulo z prej omenjenimi omejitvami. Taka klavzula je lahko licenca ali obvestilo da je izvorna koda objavljena v javni domeni.

Ko govorimo o prostem programju izraza ne smemo mešati s brezplačnim programjem ali brezplačno programsko opremo, saj kljub temu da za oba velja da sta brezplačna, brezplačno programje ne obvezuje objava izvirne kode in ga varuje *Copyright*, ki ne dovoljuje vpogleda v izvorno kodo, spreminjanje in distribuiranje. Organizacija FSF (Free Software Foundation) zagovarja stališče da je odprto programje stvar prostosti ne cene, nanaša se pa predvsem na prostost pregledovanja spreminjanja in distribuiranja programske opreme. To se natančneje nanaša na štiri vrste prostosti za uporabnika, Organizacija FSF našteva kot štiri osnovne vrste prostosti programske svobode (What is Free Software, 2009) :

- *Prostost uporabljati program v katerikoli namen (prostost 0);*
- *Prostost preučevati program in ga spremeniti po svoji želji. Dostop do izvirne kode je predpogoj za to (prostost 1);*
- *Prosto posredovanje kopij (prostost 2);*
- *Prostost izboljšanja programa in objava izboljšav (in prilagojenih verzij) javnosti v korist skupnosti (prostost 3) dostop do izvirne kode je predpogoj za to.*

Torej program je prost program le ko imajo uporabniki vse štiri zgoraj naštetih svobode in ima uporabnik s tem možnost spreminjanja in posredovanja kopij brez potrebe po raznih dovoljenjih in plačil licencah Prva prostost proste uporabe

programa pomeni svobodo vsakemu človeku ali organizaciji da uporabi program na kakršnem koli računalniškem sistemu za katerikoli namen ali nalogo, brez da bi o tem moral govoriti z razvijalcem opreme ali katerikoli drugim. V tej prostosti je poudarek na uporabnikovem namenu ne razvijalčevem in mi kot uporabniki smo svobodni uporabljati program v svoj namen in ga posredovati nekomu drugemu in ta ga tudi lahko prosto uporablja, nismo pa dolžni vsiljevati naše namene. Prostost posredovanja kopij programa mora vključevati izvršljivo binarno kodo ali izvorno kodo tako originalnih verzij kot tudi spremenjenih. V nekaterih primerih je to nemogoče predvsem zaradi jezikovnih ovir a vseeno mora vsakdo imeti svobodo posredovanja tudi takih oblik v kolikor najde ali razvije način izdelave. Predpogoj vsakega prostega programja je torej dostop do izvorne kode.

Pomemben način spreminjanja programa je združevanje že obstoječih rešitev. Če pa licenca programa tega ne dovoljuje in obvezuje uporabnika, da je lastnik avtorskih pravic za spreminjanje je licenca preveč restriktivna in se program ne šteje kot prosta programska oprema. Da bi zgoraj naštetje svobode kaj veljale morajo biti stalne in nepreklicne vsaj dokler posameznik ne stori kaj narobe. Če ima razvijalec pravico spremeniti pogoje licence ali licenco preklicati brez da bi uporabnik storil kaj narobe se program ne šteje za prosto programska opremo. Vendar so določena pravila v zvezi z načini posredovanja sprejemljiva, dokler ne nasprotujejo štirim osrednjim prostostim. Primer je Copyleft¹ pravilo, ki obvezuje da ko posreduješ program ne smeš postavljati omejitev, ki bi znikale porabnikom glavne prostosti. To pravilo ne omejuje prostosti ampak jih celo ščiti. Prosto programje pa še ne pomeni da so programi nekomercialni. Prav nasprotno prosti program mora biti na razpolago komercialni rabi, razvoju in posredovanju.

Slika 1: Različne kategorije programske opreme



Vir: What is Free Software, Free Software Organization

Komercialni razvoj prostega programja ni več nenavaden, saj je komercialna prosta programska oprema postala zelo pomembna. Ne glede na to ali smo dobili kopije proti plačilu ali brezplačno imamo vedno pravico svoje kopije svobodno spreminjati

¹ Copyleft je oblika licenciranja avtorskih del, obratno načelu Copyright s tem da odpravlja vse pravice, ki se tičejo avtorjev. Odpravlja vse omejitve glede razmnoževanja kopij ali modificiranih verzij ter zahteva, da bodo novi izdelki ali modificirane verzije prav tako prosto dostopne. Več o Copyleft v poglavju 4.

in kopirati. Kljub temu so pravila, kako predstaviti neko spremenjeno verzijo, sprejemljiva v kolikor ne omejujejo posameznikovo svobodo objavljanja prilagojenih verzij programa ali uporabljanje le tega. Prav tako so pod istim pogojem sprejemljiva pravila da objaviš verzijo na tak ali drugačen način in pravila, ki te ob objavi nekega programa obvezujejo z objavo izvorne kode ter pravila, ki te obvezujejo da spremenjeno kopijo z izvorno kodo pošlješ razvijalcu oziroma določiš katere spremembe so bile narejene. Znotraj GNU projekta se uporablja Copyleft za pravno zaščito teh pravic posameznikov, vendar poznamo tudi prosto programsko opremo, ki ni zaščiten s Copyleft. Copyleft je splošna metoda s katero zagotovimo da je nek program ali delo brezplačno in da so vse spremenjene verzije programa prav tako brezplačne (What is Free Software, 2009).

2.1.2 Odprta koda

Odprta koda je pristop do razvoja, oblikovanja in distribucije programske opreme, ki ponuja praktičen dostop do izvorne kode programske opreme. Nekateri avtorji in razvijalci sprejemajo odprto kodo kot enega mnogih različnih pristopov, med tem ko drugi gledajo na odprto kodo kot ključen element njihovega delovanja. Preden je odprta koda postala splošno priznana so avtorji in razvijalci uporabljali različne izraze za opis tega koncepta. Izraz odprta koda je postala priljubljena z vzponom interneta, ki omogoča dostop do različnih ustvarjalnih modelov, komunikacijskih poti in interaktivnih skupnosti. Primeri teh prostih vsebin vsebujejo znanstvene prispevke, film, glasbo in umetnost. V odprto kodni kulturi je določeno da so dela s pravico do zaščite z avtorskimi pravicami, splošno dostopne. Pripadniki kulture lahko spreminjajo te produkte in jih ponovno razdelijo nazaj skupnosti ali drugim organizacijam.

Porast odprto kodne kulture v 20. stoletju je posledica vse večje napetosti med ustvarjalnimi praksami, ki vsebujejo prilagojeno vsebino in zato zahtevajo dostop do vsebin, zaščiteneh z avtorskimi pravicami ter vedno bolj omejujočimi zakoni in politikami za intelektualno lastnino, ki upravljajo z dostopom do avtorsko zaščitene vsebin. Ti zakoni so v 20. stoletju postali bolj omejujoči na dva načina in sicer z razširjanjem izraza avtorska pravica (še posebej v ZDA) in kaznimi, ki so sledile za poskuse preprečevanja proti-piratskih tehnologij. Čeprav je umetniško prilagajanje pogosto dovoljeno z doktrinami pravične uporabe, kompleksnost in nejasnost teh doktrin ustvarjata negotovost med pripadniki kulture. Prav tako pa zaščitna dejanja lastnikov avtorskih pravic povzročajo tako imenovani »*chilling effect*«² med pripadniki kulture(Wikipedia, 2009) .

Na koncu 20. stoletja so pripadniki kulture vključno z Creative Commons³ začeli privzemati tehnike intelektualno-lastniških dovoljenj prostega in odprtega programja, da bi lahko njihova dela svobodneje krožila med uporabniki. Ideja odprto kodne

² Stanje, ko je govor ali vedenje zatrto s strahom kazni zaradi interesov drugih ljudi ali skupin

³ Creative Commons (kratica CC) je neprofitna organizacija, ki si prizadeva za razširitev obsega avtorskih del, ki bi bile na volju drugim, ki bi tako lahko legalno uporabili to delo za razširitev in delitev. Projekt tako ponuja vnaprej pripravljene licence, ki natančno opredelijo pogoje uporabe dotičnih del.(<http://creativecommons.org/>)

kulture je vzporedna »prosti kulturi«, v osnovi pa se razlikujeta. Izraz prosta kultura (*free culture*) je izpeljanka iz *free software movement* (gibanje za prosto programje). V nasprotju s to kulturno miselnostjo pa pripadniki odprto kodne kulture podpirajo idejo, da nekateri zakoni intelektualne lastnine morajo obstajati zaradi zaščite avtorjev in ustvarjalcev. Zakona intelektualne lastnine ne dojemajo kot izraz instrumentalnih pravil za vzdrževanje naravnih pravic in zaželenih rezultatov, ampak so za njih pomembne različne dobrine in doseganje ciljev, kot so omogočanje širšega dostopa do kulturnih vsebin.

Kot je predvideval Moorov zakon⁴ (Moore, 1965) so stroški digitalnih medijev strmo padali konec 20. stoletja. Najnižje cene digitalne reprodukcije vsega, kar se je dalo prenašati preko digitalnih medijev so padle na zelo nizko raven. Če to povežemo s porastom uporabe osebnih računalnikov in lastništvom informacijsko-komunikacijskih tehnologij, je rezultat povečanje dostopa širše populacije do digitalnih medijev.

Ta fenomen je pospešil rast odprto kodne kulture, saj je omogočil hitro in poceni reprodukcijo in distribucijo programske opreme. Dostop do večine kulturnih vsebin ustvarjenih pred pojavom digitalnih medijev je bil omejen z ostalimi ovirami lastniških in potencialno »odprtih« medijev. Digitalni mediji so bili tehnologija z zmožnostjo, da povečajo dostop do kulturnih vsebin in produktov. Umetniki in uporabniki, ki se odločijo za distribucijo svojega dela, se ne soočajo s fizičnimi omejitvami kot tradicionalni kulturni izvajalci. Izraz odprta koda je postal priljubljen z vzponom interneta, ki je omogočil dostop do različnih modelov produkcije, komunikacijskih poti in interaktivnih skupnosti. Odprto kodni model delovanja in sprejemanja odločitev omogoča hkraten vnos dela, pristopov in prioritet ter se razlikuje od bolj zaprtih, centraliziranih modelov razvoja (Wikipedia, 2009).

Če smo se pri prostem programju osredotočili na definicijo po FSF (Free Software Foundation) velja za odprto kodo vse kar je izdano pod OSI (Open Source Initiative) licenco. Open Source Initiative (OSI) je javna neprofitna združba, ki skrbi za definicijo odprte kode in je priznana organizacija za preverjanje in odobrevanje licenc na področju odprte kode. Organizacije je aktivno vključena v izgrajevanje odprto kodne skupnosti in poučevanje pomembnosti ne-lastniške programske opreme. Temelj odprte kode so torej možnosti vpogleda v izvorno kodo in prosto spreminjanje in posredovanje programske opreme. Da nek program doseže standard definicije po OSI mora zadovoljevati 10 kriterijev ali pogojev po definiciji odprte kode. V nadaljevanju so naštet in opisani kriteriji po najnovejši verziji Open Source Definition objavljeni na strani <http://www.opensource.org>:

1. Prosta distribucija:

Licenca ne sme omejevati kateregakoli uporabnika pri prodaji ali deljenju programske opreme kot komponente večje skupine programske opreme in zahtevati plačila ali globe za tako prodajo.

2. Izvorna koda:

Program mora vključevati izvorno kodo in mora dovoljevati distribucijo tako odprte kode kot izvedene oblike. V kolikor kakšna vrsta produkta ne

⁴ Moorov zakon pravi, da se približno na vsako leto in pol podvoji vsakih zmogljivost visoko zmogljivih integriranih vezij (procesorjev in namenskih procesorjev).

vsebuje odprte kode mora obstajati objavljen način pridobitve izvorne kode po možnosti dosegljiv na internetu brez plačila.

3. Izpeljana dela:

Licenca mora dovoljevati spremembe in izpeljana dela in mora dovoljevati distribucijo pod enakimi pogoji kot licenca originalne programske opreme

4. Integriteta avtorjeve izvorne kode:

Licenca lahko omejuje izvorno kodo pri posredovanju v spremenjeni obliki samo če licenca dovoljuje distribucijo popravkov, ki vključujejo izvorno kodo z namenom spreminjanja programa v času izgradnje. Licenca lahko zahteva da se izpeljana dela objavijo pod drugačnim imenom ali verzijo kot je bila objavljena originalna verzija.

5. Prepoved diskriminiranja ljudi ali skupin:

Licenca ne sme diskriminirati kogarkoli ali katerekoli skupine. Namen tega kriterija je vključitev čim več ljudi v proces oblikovanja odprte kode.

6. Prepoved diskriminiranja področja znanosti:

Licenca ne sme omejevati nobenega pri uporabi programa na drugem področju znanosti. Licenca tako ne sme nasprotovati uporabi programa v poslovnem svetu ali medicinskem področju.

7. Distribucija licence:

Pravice določene programu veljajo za vse katerim je program distribuiran brez potrebe da bi sami izdajali še dodatne licence

8. Licenca ne sme biti specifična produktu:

Pravice, vezane na program, ne smejo biti odvisne od tega, ali je program del določene programske distribucije. Če je program odstranjen iz distribucije in uporabljen ali distribuiran naprej pod pogoji licence programa, morajo vse stranke, katerim je bil program distribuiran, imeti enake pravice kot tiste, ki so jim bile dodeljene pravice za originalno programsko distribucijo.

9. Licenca ne sme omejevati druge programske opreme:

Licenca ne sme postavljati omejitev nad programsko opremo, ki je distribuirana skupaj z licencirano programsko opremo. Na primer, licenca ne sme zahtevati, da mora biti vsa programska oprema, ki je distribuirana na istem mediju prav tako odprto kodna programska oprema.

10. Licenca mora biti nevtralna do tehnologije:

Prevzemanje licenc ne sme biti omejeno na posamezne tehnološke rešitve ali vmesnike.

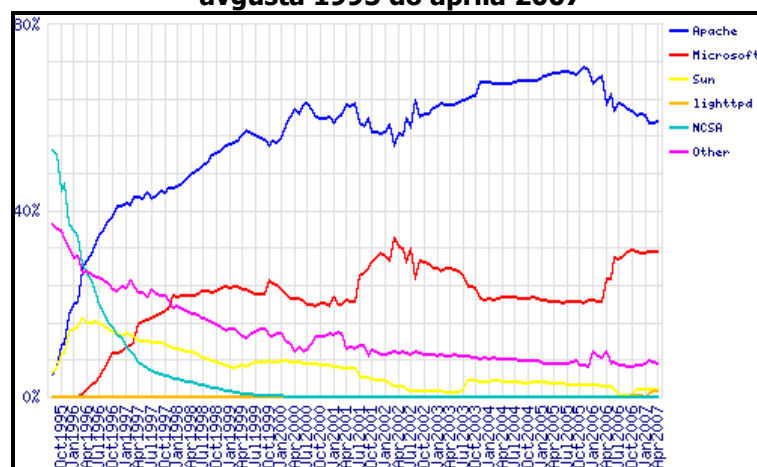
Na začetku se je izraz odprta koda nanašal samo na izvorno kodo programja, danes pa se nanaša na ogromno drugih področij, kot npr. odprto kodna ekologija, gibanje, ki se zavzema za decentralizacijo tehnologij tako, da bi jih lahko uporabljal kdorkoli. Nasprotniki širjenja izraza »odprta koda«, tudi Richard Stallman⁵, trdijo, da zahteve in omejitve zagotavljajo trajnost dosežka ter preprečujejo poskuse za preoblikovanje izraza. Stallman pravi tudi, da je večina podpornikov odprte kode v bistvu

⁵ Stallman je osrednja osebnost gibanja prostega programja, ustanovitelj GNU-ja in Free Software Foundation. V podporo tega gibanja je leta 1984 uvedel zamisel pojma brez avtorskih pravic (angleško copyleft) in ga vključil v razširjeno Splošno dovoljenje GNU. Je tudi ugleden programer in je med drugim avtor urejevalnika besedil Emacs (1976), C-jevskega prevajalnika GCC (1987) in standardnega razhroščevalnika GDB, ki so vsi del Projekta GNU.

podpornikov veliko bolj pristranskih dogovorov in da podpirajo reintegracijo doseženega dela (Stallman, 1985).

Večina, ki ni pripravljena izdati svojega dela drugim, ki bi ga lahko razvijali naprej, skrrijejo podatke ali pa za njih zahtevajo licence in/ali omejijo uporabo vsem ostalim uporabnikom. Načela odprte kode so bila uporabljena za ogromno drugih oblik uporabniško proizvedenih vsebin. Podporniki gibanja odprte vsebine uporabljajo nekaj omejitev uporabe, zahtevajo, da se spremembe delijo in da se pripiše druge avtorje proizvoda. Ta kultura zagovarja, da se načela nanašajo bolj splošno na pospeševanje hkratnih vložkov poslovanja, pristopov in prioritet, v nasprotju z bolj centraliziranimi modeli razvoja, kot so v komercialnih podjetjih.

Slika 2: Vrste spletnih strežnikov, ki prevladujejo na trgu in njihova uporaba v obdobju od avgusta 1995 do aprila 2007



Vir: David A. Weeler, 16.6.2009

Uporaba odprto kodnih rešitev je vse večja in vsak dan se pojavlja več odprto kodnih projektov. Kakršne koli skrbi o težavah v odprto kodni skupnosti so bile odveč saj je po meritvah Netcraft-a iz leta 2007 skoraj 70% vseh spletnih strežnikov Apache strežnikov, prav tako skoraj tri četrtine spletnih strani temelji na LAMP⁶. Rezultati analize uporabe spletnih strežnikov prikazuje slika 2.

Na prvi pogled se lahko razlika med odprto kodo in prostim programjem zdi minimalna, ampak sprejem tržnih pravil s strani razvijalcev odprte kode, je vodilo do tega da so se pripadniki prostega gibanja ločili od njih. Copyleft programska oprema ohranja prost dostop tako razvijalcem kot uporabnikom med tem ko odprto kodna programska oprema dopušča izdelavo plačljivih verzij. Razkol med stranema povečuje še viralna klavzula GPL licence in GPL licencirane kode.

⁶ Kombinacija operacijskega sistema Linux ter programskih paketov Apache, MySql in Perl.

2.2 ZGODOVINA IN RAZVOJ ODPRTE KODE

Kljub splošnemu prepričanju da se izraz odprta koda nanaša samo na računalnike se je izraz uporabljal že veliko preden so ljudje sploh razmišljali o računalnikih. Odprta koda se nanaša na vsa področja človekovega delovanja od odprte politike, odprtega izobraževanja do odprte glasbe in odprtega filma. Na začetku digitalne tehnologije ni bilo jasne meje med strojno in programsko opremo. Vsak konstruktor ali uporabnik nekega stroja je bil prisiljen napisati lastno programsko opremo za uporabljanje teh strojev.

Take ad hoc⁷ rešitve za uporabo strojne opreme, tipično niso bile uporabne pri drugih tipih strojne opreme. Zato se telekomunikacijska podjetja Western Electric in ATT v ZDA zaceli z razvojem splošno uporabne programske opreme (oz. operacijskega sistema), ki bi pomagala uporabnikom razvijati univerzalno kodo uporabno na različnih tipih strojne opreme. Leta 1954 je pravosodna oblast ZDA v sklopu proti monopolne zakonodaje odločila, da se Western Electric in ATT kot javni službi ne smeta več angažirati v proizvodnji, ki ni neposredno povezana s telekomunikacijami. Zato so si pri ATT izmislili licenciranje programske opreme in drugih komunikacijskih tehnologij z nominalno pristojbino in niso jih mogli več obtoževati, da proizvajajo programsko opremo (ne-telefonsko robo) za profit. Ta strategija je omogočila Bell Labs, da so še naprej razvijali in distribuirali programsko opremo za druge uporabnike. Leta 1964 so raziskovalci na MIT, v sodelovanju z Bell Labs in General Electric razvili sistem "time sharing", ki so ga poimenovali MULTICS. Programske sestavine projekta je bilo zelo težko implementirati in leta 1969 so v Bell Labs vse skupaj opustili (Pivec, 2006).

2.2.1 UNIX

Leta 1969 sta se dva raziskovalca Ken Thompson in Dennis Richie odločila da bosta nadaljevala delo Bell Labs in sta razvila jedro operacijskega sistema, ki sta ga prvotno poimenovala UNICS kasneje pa poenostavila ime v UNIX. UNIX je postal tako danes najbolj uporabljen operacijski sistem v raziskovalnih skupnostih, vojaških in poslovnih sistemih ter internetu, kasneje so pa uporabniki sami začeli izpopolnjevati projekte in programe. UNIX je teklen na vsakem stroju, ki je imel prevajalnik za C in ne samo na DEC, na katerih je bil sicer razvit. UNIX je hitro postal najbolj uporabljeno učno orodje, ki ga je bilo mogoče hitro razumeti in hitro spremeniti ali prilagoditi potrebam saj je bila izvorna koda na razpolago vsem. UNIX je tako postal zelo popularen zelo hitro in to v času ko niso poznali interneta, zato so potekale razprave o UNIXu na raznih konferencah, seminarjih in drugih srečanjih avtorjev in razvijalcev, ki so se na podlagi tega povezali v močno mednarodno skupnost.

Računalniški oddelek na univerzi v Berkeley je bil med prvimi ki je sprejel sistem UNIX. Istega leta se je Kenu Thompsonu na univerzi pridružila nova generacija

⁷ Ad hoc je fraza latinskega izvora in pomeni »za ta namen«. Normalno označuje rešitev ki je specifična za to nalogo ali problem, se je ne da posplošiti in je neuporabna za druge namene. Znani primeri so organizacije, komiteji in komisije postavljene na nacionalni ali internacionalni ravni za specifično nalogo

študentov med katerimi sta bila Bill Joy in Chuck Haley, ki sta mu leta 1976 v sistemu Pascal pomagala izpopolniti jedro UNIXa. Joy je kasneje zbral komplet vseh uporabljenih orodij in pripomočkov in jih poimenoval Berkeley Software Distribution (BSD) in v taki obliki je zadeva tudi prišla med uporabnike. Odtlej se je BSD izpopolnil v številnih verzijah, kar je zasluga študentov na Berkeleyu, drugih razvijalcev iz različnih koncev sveta. Zaradi velikega povpraševanja po UNIXu so začeli pri ATT razmišljati, kako bi vzpostavili nadzor nad produktom, zato so omejili brezplačno uporabo njihove licence zgolj na Berkeley in samo za študijske in raziskovalne namene.

Kot reakcija na namero Bell Labs, da bi komercializirali uporabo UNIXa, je skupina programerjev na MIT-jevem Artificial Intelligence Lab pod vodstvom Richarda Stallmana leta 1984 zasnovala Free Software Foundation z načrtom, da bodo zgradili operacijski sistem, ki bi sicer temeljil na tradiciji UNIXa, a ne bi bil UNIX in bi ga lahko prosto distribuiral in uporabljali. Poimenovali so ga GNU.

2.2.2 GNU

Marca, 1985 je Richard Stallman je napisal in objavil GNU Manifesto, kjer je naštel že prej omenjene svobode (*GNU Manifesto, 1985*):

- *svobodo uporabe programa za kakršen koli namen;*
- *svobodo proučevanja programa in njegovega preoblikovanja glede na potrebe;*
- *svobodo razširjanja kopij, bodisi za plačilo ali brezplačno;*
- *svobodo spreminjanja programa in njegovega ponovnega razširjanja v javnosti na način, da lahko vsakdo naredi isto.*

Pogoj za tako svobodno delovanje je, da ima uporabnik popolnoma prost dostop do izvirne kode programov. Stallman je posledično zasnoval pravno orodje za varovanje teh štirih prostosti in sicer General Public Licence (GPL). Z zasnovanjem te licence je Stallman nadomestil Copyright s Copyleft. Tako ni bilo več mogoče programske opreme licencirane z GPL licenco privatizirati in kodo pod GPL ni bilo več mogoče uporabljati v katerikoli kombinaciji z lastniško programsko opremo, če celotna kombinacija ni bila objavljena pod GPL. Free Software Foundation je prispevala nekaj izredno kakovostne programske opreme, v katero gotovo sodita prevajalnik za programski jezik C (GCC) in urejevalnik besedil GNU Emacs. Seveda pa zaradi zaščite omejenih virov pod GPL nikoli ni mogel nastati popoln operacijski sistem. K temu je prispevalo tudi dejstvo, da je Stallman delal s precej ozko skupino programerjev, pod močnim intelektualnim nadzorom in se ni posluževal potenciala sodelovanja preko interneta.

Zato je bil GPL/GNU konec osemdesetih precej omejeno gibanje, v katerem je sodelovalo skromno število programerjev. V tem času pa je BSD doživel številne komercialne aplikacije, med katerimi so bile najopaznejše rešitve in aplikacije podjetja Sun Microsystems⁸ v ZDA. V poslovnem svetu so največji uporabniki UNIXa

⁸ <http://www.sun.com/>

(kot so IBM, HP in Siemens) ustanovilo Open Software Foundation kot opozicijo ATT-emu vztrajanju pri nadzorovanju UNIXa, vendar je kriza računalniške industrije v začetku devetdesetih povzročila njeno ukinitve. Vse je kazalo, da se bo ideja odprte kode omejila na subkulturo programerjev, brez potrebnih orodij za uresničitev njihovih sanj. Vznemirljivo je bilo, da je Microsoft tako zlahka dosegel monopol pri operacijskem sistemu in kazalo je, da bo tudi UNIX z derivati vred krenil v smer popolne privatizacije.

2.2.3 Linux

Linux je nastal iz potrebe Linusa Torvaldsa, 21-letnega študenta računalništva na helsinški univerzi, ki je leta 1991 pripravljaj svoj magisterij. Želel je uporabiti UNIX, vendar ni imel dovolj zmogljivega stroja, pač pa se je spustil v poizkus, da bi UNIX priredil za svoj nov PC386. Začel je z zelo primitivnim jedrom, jeseni 1991 pa je že objavil verzijo 0,02, ki ji je dal ime Freeix, njegov sistemski administrator pa ga je samovoljno preimenoval v Linux. Torvalds je postavil svoje jedro operacijskega sistema Linux na internet ter pozval na pomoč pri njegovem razvoju. Seveda je razkril tudi izvorno kodo in eksplicitno nadziral njene spremembe ob pričakovanju, da bodo tudi te spremembe objavljene na način, da jih bo mreža zainteresiranih programerjev lahko izkoristila pri svojem delu (Peeling, Satchell, 2001).

Januarja 1992 je Torvalds objavil že novo verzijo (Linux 0.12), ki jo je opremil z GPL, vključno z "*viralno klavzulo*⁹", kar je pomenilo, da so vsi prispevki k razvoju Linuxa odprti in prosti. Od samega začetka se je Linux razvijal v mreži sodelovanja, ki se je odvijalo odprto na internetu in kljub ostrim razpravam nikoli ni zapadlo v ideološke razprtije. In čeprav se je rodil kot prost, Linux nikoli ni bil obtožen s strani korporacijskih pravnikov (le 2002 so se začeli manevri Microsofta, s katerimi so poskušali dokazati, da Linux nepooblaščno vključuje lastniško programsko opremo drugih podjetij, česar pa po dveh letih pravedanja niso mogli dokazati).

Do marca 1994 je bil Linux prvič izdan pod verzijo 1 in skupnost uporabnikov je začela močno rasti. Ena bolj opaznih lastnosti Linuxa so njegove nizke zahteve po zmogljivosti strojne opreme, kar omogoča uporabo na starejših strojnih konfiguracijah. V razvoj Linuxa se je naglo vključila mreža tisočih sodelavcev z jedrom nekaj sto programerjev in s približno 21 milijoni uporabnikov že v letu 2002. Klub takšnemu obsegu mreže in naraščajoči kompleksnosti programa, se je Linux izognil "razkolu", ko bi del skupnosti razvil alternativno verzijo kode, neskladno z drugimi, kar bi razdvojilo sodelavce. K temu je največ prispevala Torvaldsova odprtost, s katero je sprejel obsežne modifikacije kode in dopuščal drugim ljudem, da predlagajo in nadzirajo ključne sestavine kode. To je še posebej očitno pri razvoju operacijskega sistema za internet mimo uporabe kode, ki bi bila kakorkoli povezana s spori med ATT in BSD na Berkeleyu. Linus Torvalds je glede tega prisluhnil angleškemu razvojniku Alanu Coxu (Pivec, 2006) .

⁹ Kode z GPL ni mogoče uporabljati v kombinaciji z katerokoli programsko opremo če nastala kombinacija ni kasneje objavljena pod GPL

Linux so postopoma sprejele vse velike korporacije, vlade in institucije. Vse večje število podjetij se ukvarja z razvojem Linuxovih aplikacij in servisov. Vse to se je dogajalo, ne da bi se odrekli značaju "odprto kodnega programa", brez zasebnega prisvajanja programske opreme in brez dobičkov, ki bi si jih prisvajal Linus Torvalds ali kdo drug od glavnih razvijalcev v mreži. Linux se še naprej razvija in vzdržuje v globalni skupnosti z interakcijami preko interneta, brez formalne hierarhije in brez zapovedane delitve dela.

2.2.4 Apache

Pomembna stopnja v zgodovini odprte kode je razvoj programske opreme spletnega strežnika Apache. Gre za programsko opremo, ki odgovarja na zahteve za pošiljanje spletne strani računalniku, ki povprašuje po njej. Prvi spletni strežnik je bil Daemon, ki ga je razvil Rob McCool v National Center for Supercomputing Application na Univerzi Illinois. (Peeling, Satchell, 2001).

Je pa delo nadaljeval Brian Behlendorf, študent na Berkeleyu, ki je sodeloval pri spletni publikaciji Wired Magazine. Ker je bila NCSA pod režimom odprte kode, jo je Behlendorf lahko uporabil. K projektu je pritegnil še osem drugih razvijalcev in po treh mesecih so imeli na poštni listi projekta že 150 računalnikov. Decembra 1995 so objavili Apache 1.0. Ker je bilo mnogo razvijalcev vključenih tudi v razvojne aktivnosti komercialnih spletnih strani, so iz previdnosti kodo objavili pod licenco BSD in ne pod GPL. Ključna razlika je, da BSD zahteva objavo izvirne kode le za tiste dele kode, ki izvirajo iz odprto kodne osnove. Ne preprečuje pa kombiniranja z lastniško programsko opremo za komercialno rabo, pod pogojem ohranjanja odprtega dostopa do informacij, ki so bile izvirno objavljene v odprtem formatu. Še naprej se razvija kot dosledno odprto-kodni projekt (Pivec, 2006).

2.2.5 Odprta koda danes

Za nas uporabnike so najbolj pomembne tri točke v razvoju odprte kode. Prvi je začetek razvojnega projekta KDE leta 1996, katerega nosilec je bil Matthias Ettricht. Rezultat projekta je bil tako imenovani Gnome, ki je poenotil zgled namizja na računalniškem zaslonu. Drug mejnik je bila odločitev Sun Microsystems, da izda StarOffice pod GNU LGPL licenco. Posledica tega je bil razvoj OpenOffice, ki je kompleten prosto dostopen in odprto koden pisarniški komplet. Tretji pomemben projekt pa je Mozilla. Mozilla je nastala iz leta 1998 sproščene kode Netscape Communicator-ja. Mozillin najbolj znan produkt je spletni brskalnik Firefox. V zadnjih letih je začela odprta koda dobivati zagon. Veliko k temu prispeva pojav uporabnikom prijaznih distribucij Linux-a. Ena takih je Ubuntu ki je ena najprijaznejših. Velik pomen pri popularnosti odprte kode ima tudi sprejem odprtih standardov in vedno hujša borba proti piratstvu.

2.2.6 Odprta koda v Sloveniji

Ministrstvo za šolstvo, znanost in šport Republike Slovenije je pričelo v letu 2003 izvajati dejavnosti namenjene uvajanju odprto kodne in proste programske opreme v vzgojno-izobraževalne zavode. Te dejavnosti je Ministrstvo združilo v projekt OKO. Snovalci projekta OKO so kot poglavitni cilj projekta opredelili hitro in učinkovito uvajanje odprto kodne in proste programske opreme v vzgojno-izobraževalne zavode. Ena večjih težav pri uporabi odprto kodne programske opreme je pomanjkanje znanja v vzgojno-izobraževalnih zavodih. Tako je bila temeljna naloga projekta OKO v prvi vrsti izobraziti učitelje, da bodo znali in želeli uporabljati odprto kodne programe in na tem temelječe didaktične aplikacije (Projekt OKO, 2004).

Veliko odprto kodne programske opreme najdemo na svetovnem spletu. Med njimi tudi didaktične odprto kodne programe, ki jih je potrebno prevesti, dograditi in uporabiti pri poučevanju in učenju. Ti izdelki in polizdelki so priložnost za naše razvijalce, učitelje, dijake, študente. Projektna skupina OKO je bila sestavljena iz študentov, dijakov profesorjev, strokovnjaki iz podjetij in samostojni raziskovalci. Z Ministrstvom za šolstvo, znanost in šport, ki je omogočilo začetne dejavnosti skupine OKO so sodelovali vzgojno-izobraževalnimi zavodi, Zavod Republike Slovenije za šolstvo, Univerze v Ljubljani, Mariboru in Kopru, Center Vlade RS za informatiko, Ministrstvo za informacijsko družbo, podjetje Agenda, društvo Lugos in samostojni strokovni sodelavci.

Med rezultati polletnega dela skupine je bila slovenska distribucija operacijskega sistema Linux OKO/Pingo 2.0 z grafičnim namizjem, spletnim brskalnikom, elektronsko pošto in pisarniškim paketom. Vsa programska oprema je v slovenskem jeziku in je na voljo na zgoščenkah. Narejena je bila tudi obširna strokovna literatura v slovenskem jeziku, ki je dostopna v elektronski in tiskani obliki.

Posebno priznanje je distribuciji operacijskega sistema Linux OKO/Pingo 2.0 podelilo Ministrstvo za informacijsko družbo, ki mu je podelilo prvo nagrado na Natečaju za odprto kodne projekte. Vsa programska oprema na zgoščenkah se lahko uporablja v skladu z licenco, ki je združljiva z določili OSI (<http://www.opensource.org/>) in jo lahko v Sloveniji prosto razširjamo.

Nato je leta 2007 je Ministrstvo za visoko šolstvo, znanost in tehnologijo prvič objavilo razpis za izbor izvajalca za razvojno-podporni center odprte kode Center Odprte Kode Slovenije¹⁰ (COKS). Na razpisu je zmagal konzorcij, ki ga sestavljajo gospodarske družbe in neprofitne organizacije, pod vodstvom podjetja Agenda d.o.o. Center odprte kode se pojavlja prvič v takšni obliki v Sloveniji in bo uporabnikom nudil storitve pomoči in podpore ter bo zagotavljal rešitve za potrebe javnega in zasebnega sektorja. Vse storitve, ki jih izvaja COKS, izvajajo organizacije ustanoviteljice konzorcija, oziroma njihovi zaposleni ali pogodbeni sodelavci.

Glavni cilji ustanovitve centra so bili vzpodbuditi razvoj, širjenje in uporabo programske opreme in rešitev problemov temelječih na odprti kodi. Zaradi tega so v

¹⁰ Več o COKS na strani <http://www.coks.si/>

okviru ustanovitve COKS-a zagotovili sistemsko podporo, klicni center in razvoj aktualnih odprto kodnih rešitev. Med glavne naloge centra sodijo nudenje pomoči in svetovanje uporabnikom pri rešitvah, temelječih na odprti kodi, ter svetovanje javnemu sektorju pri uvajanju, implementaciji in uporabi odprto kodnih rešitev. Center nastopa tudi v vlogi nacionalnega koordinatorja na področju strategij odprte kode ter spodbujevalca sodelovanja med različnimi neprofitnimi organizacijami, gospodarskimi subjekti ter posamezniki (OKO-Poročilo o ustanovitvi in pričetku dela, 2007).

3 ZNAČILNOSTI ODPRTE KODE

3.1 PREDNOSTI ODPRTE KODE

Zagovorniki odprte kode pogosto trdijo, da ta v primerjavi z komercialnimi produkti nudi številne prednosti. Komercialni izdelki pogosto dajo prednost vizualnim učinkom pred težje merljivimi kvalitetami kot so stabilnost, varnost in podobnimi manj vidnimi značilnostmi. Razvijalci so očitno motivirani zaradi številnih faktorjev, ampak dajanje prednosti zmožnostim preko kvalitete ni eden od teh. Za mnoge razvijalce je priznanje sodelavcev in pohvala pomembna, zato bodo verjetneje gradili programsko opremo, ki jo bodo strokovnjaki občudovali. Zelo pomembni dejavniki so za njih torej preprosta zasnova, zanesljivost in vzdržljivost z upoštevanjem znanja in vrednot skupnosti in v nadaljevanju si bomo ogledali ali odprto kodne rešitve res ustrezajo tem značilnostim (Smeh, 2005).

3.1.1 Zanesljivost

Zanesljivost je prost izraz in na grobo ga lahko opredelimo kot pomanjkljivost defektov ali napak, ki bi povzročila izgubo podatkov ali nenadnih sesutij. Natančneje, napaka bi tudi pomenila nedoseganje standardov, toda glede na to da večina odprto kodnih projektov odvrta stroge definicije kaj je posebnost ali kaj napaka, je težko natančno definirati napako. Očitna nesposobnost izvedbe ali neizpolnjevanje predpisanih funkcij bi lahko opisali kot napako. Varnostne nepravilnosti so tudi napake. Take in drugačne napake so popravljajo s popravki tako imenovanimi "patch-i" in zagovorniki odprte kode zagovarjajo hitro odzivnost za popravilo teh napak v programih (Benefits of using open source, 2009).

Hujše napake so ponavadi popravljene v roku nekaj ur, postopek, ki mu močno pripomore dosegljivost odprte kode. Sposobni ustvarjalci bodo napake odkrili jih popravili o njih obvestili vzdrževalce in pogosto tudi objavili popravljeno verzijo pod svojim imenom. Uporabniki se pa nato odločijo ali bojo uporabili neuradno popravljeno verzijo ali uradno popravljeno verzijo. Uradna različica pomeni originalno kopijo programa izdano s strani distributerjev odprto kodnih paketov. Če prej omenjeno primerjamo z zaprtimi ali licenčnimi programi ponavadi pri pojavu napake pri teh uporabnik najprej izpolni nek obrazec ali poročilo o napaki in ponavadi preteče nekaj časa preden prodajalec določi kdaj bo izšla popravljena verzija. Tako so uporabniki veliko bolj odvisni od notranjih procesov prodajalcev kot pri odprto kodnih rešitvah.

3.1.2 Stabilnost

V poslovnem okolju je programska oprema prej kot ne neko nujno zlo, orodje za opravljanje dela. Razen če se delo spremeni ali se odkrijejo bolj pomembni procesi je redko pritisk da se razvije nova programska oprema. To je ravno nasprotno željam ali motivaciji prodajalcev programske opreme in jih postavi u neugoden položaj zagotavljanja dobrine ki ne zastara. Prodajalci morajo imeti stabilno prodajo če želijo imeti nek reden priliv medtem ko uporabniki nimajo ne želje ne potrebe po novih sistemih dokler le ti zadovoljujejo njihove potrebe. V kolikor lahko prodajalci vzpostavijo nek virtualen monopol nad neko programsko opremo tako lahko vsiljujejo nadgradnje te opreme (Benefits of using open source, 2009)

Odrpta koda tako še ne pomeni nekega univerzalnega zdravila v svetu hitro spreminjajoče programske opreme, vendar lahko omili posledice komercialne prodaje. Odrpto kodni produkti se pogosto prilagodijo standardom, saj se standardi spreminjajo počasi in izmenjalni formati so dokaj stabilni. Rezultat je, da so nezdružljivi vrst datotek manjši problem.

V praksi imajo odrpto kodne rešitve dokaj dobra poročila in ko so uporabljeni neznani ali nezdružljivi formati je ponavadi priložen pretvornik kot je Perl, ki bo pretvoril podatke v nov format. V realnosti ni nič statičnega in programske spremembe morajo slediti trendom. Izbira uporabe odrpte kode je kot proti utež posodabljanju opreme po želji komercialnega ponudnika.

3.1.3 Preglednost

Pogosto spregledana prednost programske opreme je pogosto njena preglednost. Zaprto kodna programska oprema ali licenčna sili uporabnike da zaupajo proizvajalčevim trditvam o varnosti, značilnostih, kompatibilnosti standardom in fleksibilnosti v prihodnosti. V kolikor izvorna koda ni znana te trditve res ostanejo samo trditve.

Z objavljanjem odrpte kode, avtorji zagotavljajo uporabnikom teh programov da imajo zaupanje da obstaja podlaga za trditve proizvajalcev. Ne glede na to ali gre samo za navaden pregled ali bolj strogo pregledovanje je jasno da brez objave izvirne kode zunanji pregled ni mogoč. Pregledi se trenutno še ne zahtevajo a se ob vse večji popularnosti odrpto kodnih sistemov lahko pojavljajo v vse večjem številu.

3.1.4 Stroški

Večina odrpto kodnih projektov je dostopna brezplačno. Zagovorniki odrpto kodnih programskih rešitev pogosto zagovarjajo svobodo pred stroški, kljub temu so pomembnejši odrpto kodni projekti prosti v obeh pogledih. S strani poslovnega sveta nakupni strošek ni najpomembnejši dejavnik ampak le eden izmed mnogih. Celotni stroški lastninjenja ali TCO (Total Cost of Ownership) je tisto kar je najpomembnejše. Argumenti, ki govorijo u prid tem argumentom so (Peeling, Satchell, 2001):

- *Možnost ničelne nakupne cene*
- *Brez stroškov v primeru potrebe več kopij*
- *Zmanjšana potreba po rednih posodobitvah*
- *Daljše delovanje opreme in zmanjšana potreba po dragih administratorjih*
- *Skoraj sto odstotna varnost proti virusom zmanjšanje potrebe po preverjanju, izgubi podatkov in nedelovanju*
- *Podaljšanje delovne dobe starejše programske opreme brez zmanjšane zmogljivosti*

Na žalost so na tem področju številne trditve in proti trditve. Zanesljive stroške lastninjenja je praktično nemogoče ugotoviti. Tako bodo različne organizacije morale same ugotoviti kje bi lahko odprto kodne rešitve prinesle prednosti v stroških pred licenčno programsko opremo.

3.1.5 Prilagodljivost in svoboda

Pri prilagodljivosti gre predvsem za možnost izbiranja rešitev primerne za uporabnike. Številna licenčna oprema trdi da ima že vgrajeno prilagodljivost in mnogi jo dejansko tudi imajo. Z izbiro odprto kodnih programskih rešitev se lahko organizacije in posamezniki izognejo izključevanju, ki ga nudi komercialna programska oprema, saj ponudniki komercialne opreme želijo doseči, da bi uporabniki izključno uporabljali njihovo programsko opremo in si s tem zagotovili neko odvisnost uporabnikov na njihovo programsko opremo.

Pri odprto kodni programski opremi za tako izključevanje ni visoke motivacije in je upoštevanje "de iure" ali "de facto"¹¹ standardov. Kjer teh standardov ni še vedno ostaja dejstvo da je izvorna koda objavljena in to preprečuje izključevanje. To se kaže predvsem na uspehu odprte kode na omrežnem področju. Mnogi komercialni ponudniki so poskušali ustvariti strežnike, ki bi bili konkurenčni Apache strežnika še nobenemu ni uspelo uspešno prilastiti večji delež internetnega brskalnika in strežnika.

3.1.6 Podpora in odgovornost

Načini za dobivanje podpore in odgovornosti so pri odprto kodni so očitno drugačni, toda izid uporabe te opreme je v večini boljši od licenčnih razmerij med uporabnikom in ponudnikom.

Eden najbolj pogosto uporabljenih argumentov pri uporabi odprto kodnih programskih rešitev je "koga tožim", torej kdo bo odgovoren za napake pri delovanju programa. Argument se na prvi pogled zdi pravilen, toda v realnosti je resničen v le majhnem številu primerov. Pregled običajnih licenc programske opreme

¹¹ de iure standardi nastajajo po zakonu pod pokroviteljstvom avtoriziranih agencij za standardizacijo. Nastajanje takšnega standarda je zaprto v toge strokovno administrativne postopke, ki imajo za posledico veliko počasnost pri sprejemanju standardov. de facto standardi nastajajo neodvisno od mednarodno priznanih organizacij za standardizacijo: VMS, DOS, UNIX, PC. Rojevajo se veliko bolj praktično in se potrjujejo s svojo komercialno in uporabniško kvaliteto.

pove da se vse odvezujejo odgovornosti in prelagajo večino odgovornosti na uporabnike ne na ponudnika. Vse te licence so namenjene varovanju ponudnika. Odprto kodne licence se ne razlikujejo od tega, vendar uporabnike odprto kodnih rešitev ne zanima na koga se bodo obrnili za odgovornost, ampak cenijo uporabnost, zanesljivost in varnost same opreme. Nasprotniki trdijo da odprto kodna programska oprema nima tehnične podpore ponudnika, vendar tudi mnogo licenčne programske opreme nima tehnične podpore in prepušča uporabnike same sebi.

3.2 SLABOSTI ODPRTO KODNE PROGRAMSKE OPREME

Poleg vseh omenjenih prednosti odprto kodnih programskih sistemov seveda obstajajo slabosti odprto kodnih programskih rešitev in tako kot na drugih področjih v življenju tudi tukaj velja rek da nič ni zastonj. Toda večina slabosti se navezuje samo na tiste, ki niso spretni z izvorno kodo in se niso pripravljene potruditi za neko ustrezno programsko rešitev. V nadaljevanju bom naštel nekaj primerov slabosti odprto kodnih programskih rešitev(Smeh, 2005):

1. **Projekti lahko zamrejo:** Kot pri komercialnih programih nekateri odprto kodni programi enostavno ostanejo na mestu se ne razvijajo naprej in zamrejo. Programerji enostavno izgubijo zanimanje za razvijanju programa in le ta zamre. V tem primeru uporabniki ostanejo s programsko opremo, ki je polna napak in nevarnosti vdorov in brez posameznika, ki bi jih popravil in če jih ne morejo popraviti sami morajo za to nekoga plačati.
2. **Podpora:** Če uporabljamo licenčno programsko opremo je dobavitelj obvezan zagotoviti podporo v nekem normalnem časovnem okviru. V odprti kodo lahko najdemo napako, ki za katero bomo težko našli nekoga, ki bi jo odpravil ali bi za to celo morali plačati. Čeprav ima odprta kodna skupnost številne forume, ki služijo kot neki centri za podporo a številna vprašanja še vedno ostanejo neodgovorjena. Tako bi na primer uporaba odprtega programa povzročila enake stroške, kot uporaba licenčnega programa, če bi morali iskati strokovnjake za odpravo napak. Torej v kolikor potrebujemo podporo pri uporabi programa preko telefona ali elektronski pošti odprta koda ni za nas.
3. **Nove zmožnosti:** Četudi mislimo da bi bila neka nova zmožnost dobra za nek program, ni nujno da se s tem strinja tudi skupnost in mogoče bomo za nove možnosti primorani plačevati da bi se novost uvedla na naši programski opremi.

Kako se torej odločiti za uporabe odprte kode in katere dejavnike upoštevati pri izbiri uporabe odprte kode. Z velikim številom odprto kodne programske opreme bi lahko rekli celo da nam ni treba kupovati programske opreme. Toda na koncu bomo plačali največ prav s časom. Čas je potreben za postavitve sistema njegovo prilagajanje in odpravljanje težav. Če pa že ni časa za te stvari je pa potrebno razmisliti ali smo pripravljeni najeti strokovnjake za to in ali obstajajo posamezniki, ki bi bili zmožni tak sistem vzdrževati (Smeh, 2005).

Če se odločimo za izbiro odprto kodne programske opreme moramo upoštevati naslednje dejavnike:

- *Ali je programska oprema stabilna ali se še razvija*
- *Ali so redne posodobitve, popravki in novosti*
- *Ali obstaja forum ali drug način podpore uporabniku*
- *Ali si uporabniki programa pomagajo med sabo*
- *Ali katera uveljavljena organizacija že uporablja tak sistem*
- *Ali bodo stroški z vzdrževanjem sistema in strokovnjaki ki bi sistem vzdrževali*

3.3 ODPRTA KODA IN ZNANOST

Ko govorimo o odprti kodi in odprto kodnem modelu ne mislimo vedno na programsko opremo in računalništvo. Odprta koda lahko pripomore pri različnih znanstvenih disciplinah in danes poznamo več načinov uporabe odprtih modelov pri različnih disciplinah. Med najbolj znanimi uporabami odprto kodnega modela so odprt dostop ali "Open Acces", odprti podatki ali "Open Data" in veliko drugih kot so odprti repozitoriji in odprte raziskave. Vsem je skupno prosto kroženje podatkov brez licenčnih omejitev ali plačevanja avtorskih pravic. Prednost teh metod je predvsem kroženje znanja in znanstvenikom in raziskovalcem ni treba ponavljati metod in poskusov od začetka. V nadaljevanju tega poglavja si bomo podrobneje pogledali enega pomembnejših načinov odprtega deljenja podatkov, to je odprt dostop in primer kako lahko prevzem odprto kodnega modela pozitivno vpliva na znanstveno disciplino.

Zadnjih nekaj stoletij so bili temelj ustaljenega znanstvenega procesa odprte publikacije novih znanstvenih rezultatov, dopolnjeni z posledičnimi kritikami in citati in postopno uveljavitvijo najpomembnejših del in rezultatov v neko skupno znanje. Ta proces je bil okrepljen z vzponom preglednih dnevnikov. Prvotno so bili taki dnevniki objavljeni s strani znanstvenih društev, kot storitev namenjena svojim članom, a je zaradi nagle rasti števila teh dnevnikov in poročil proces objavljanja v drugi polovici dvajsetega stoletja padel v roke predvsem komercialnim založnikom.

Zaradi naglega naraščanja števila znanstvene literature je nastalo omrežje knjižnic, ki bi pomagale znanstvenikom pri iskanju zanimivih stvari z kazali in posojilnimi postopki. To je delovalo dobro vse do srednjih devetdesetih oziroma do razvoja interneta. Hitri vzpon interneta, kjer so bili akademiki v ospredju kot uporabniki je korenito spremenil položaj. Za znanstvenike je bil prost dostop do podatkov na spletu velika prednost in veliko preprosteje je bilo naložiti prosto dostopno literaturo z interneta kot iskanje po knjižnicah in plačevanje članarin. Toda kljub prednostim, ki jih prinaša prosto objavljane literature preko spleta še vedno obstajajo debato o tem kaj je dejansko ceneje. Založniki trdijo da je objavljanje literature na spletu enako drago kot navadno objavljane, kar je bilo tudi z več študijami jasno zavrnjeno in tudi če gledamo s strani celotnega postopka nastajanja literature in metod stroški objavljanja ne predstavljajo večji del stroškov (Björk, str 14).

Toda znanstvene publikacije niso tako poceni in dostopne, kot bi si posameznik predstavljal. Pomembnejši znanstveni dnevnik niso prosto dostopni širši javnosti in so pogosto omejeni na nekaj naročnikov. Malo znanstvenih blogov, kot so blog strojnega učenja, umetne inteligence ali javna knjižnica znanstvenih blogov, je prosto dostopnih pod odprto kodnim modelom.

Odprto kodna literatura je v električni obliki dostopna na internetu brez omejitev in avtorskim pravic med tem ko je bilo to nemogoče v času tiska. Tako bi odprti dostop do raziskav drugih znanstvenikov premostil vse cenovne ovire in večinoma drugih licenčnih ovir (Journal of Machine Learning Research 8, str 2445).

3.3.1 Prost dostop

Prost dostop oz. "Open Access" (OA v nadaljevanju) je prost, neposreden, stalen, s polnim besedilom dostopen material za kateregakoli uporabnika, razširjen na spletu, za digitalni znanstveni in šolski material. Primarno so bili znanstveno raziskovalni članki objavljeni v znanstvenih revijah. Prosto dostopni članek ima omejene avtorske pravice in licenčne omejitve, kar pomeni, da si lahko kdorkoli, kjerkoli, z dostopom do interneta lahko bere, naloži, kopira ter posreduje ta izdelek.

Prvo večja mednarodna izjava glede prostega dostopa je bila "*Budapest Open Access Initiative*"¹² v februarju 2002. Zagotavljala je definicijo prostega dostopa, in vsebovala naraščajočo listo podpisnic. Naslednje izjave, ki so sledile so bile "*The Bethesda Statement on Open Access Publishing*" v juniju 2003 in "*Berlin Declaration on Open Access to Knowledge in the Science and Humanities*" v oktobru 2003.

Prvi prosti znanstveni arhiv je bil arXiv.org, ustanovljen leta 1991. Na začetku je bil pred tisk za fizike, kot so Paul Ginsparg. Samo-arhiviranje je kasneje postalo standardno v fiziki, tako kot v energijski fiziki, ki je imela 100 odstotno raven samo-arhiviranja. Danes arXiv arhiv vsebuje dela različnih disciplin, kot so računalništvo in matematika, toda pred odprtimi arhivi, kot so arXiv so znanstveniki raje samo-arhivirali na svoje osebne spletne strani in so to počeli celo dlje kot pa sami fiziki. Izumitelji interneta in računalniške znanosti so samo-arhivirali na lastno FTP spletno stran. V letu 1997 je Nacionalna knjižnica medicine (NLM) ustvarila »Medline«, najbolj obširno kazalo za medicinsko literaturo na planetu, prosto dostopno v obliki »PubMed«. Do povečanja uporabe te podatkovne baze je prišlo s tem, ko je postala prosta, vzrok za slabo dostopnost prej je bila pa predvsem omejitve na začetku. Medtem ko razna kazala niso najpomembnejša za prosto dostopno gibanje, je »Medline« pomemben z vidika, da je prost in da je povzročil čisto novo obliko uporabe znanstvene literature s strani javnosti, ne samo s strani profesionalcev.

V letu 1998 se je pojavil Ameriški znanstveni prosto dostopni forum. Leta 1998 pa se je pojavila ena izmed prvih prosto dostopnih revij, ki je imela prvo objavljeno temo leta 1999. To je bila revija medicinskih internetnih raziskovanj (JMIR). V letu 1999 je Harold Varmus iz NIH predlagal revijo, z imenom E-Biomed, zasnovano kot prosto dostopno elektronsko publikacijo, ki bi bila kombinacija med pred tiskom in za nazaj objavljenimi članki. Revija E-biomed je kasneje uzrla luč v prenovljeni obliki kot PubMed Central. V letu 2000 je bil BioMed Central, prosto dostopni založnik objavljen s strani trenutne znanstvene skupine. BioMed Central je bil v nekaterih točkah podoben Harold Varmus-ovemu originalnemu E-biomed-u. BioMed Central dandanes objavlja več kot 170 revij.

V letu 2001 je 34.000 učenjakov po svetu podpisalo »Odprto pismo znanstvenim založnikom«, ki so pozvali za ustanovitev spletne javne knjižnice, ki bi bila opremljena z vsebinami objavljenih zapisov, raziskav, predavanj medicine, katere bi bile prosto dostopne z možnostjo polnega iskanja. Znanstveniki so tudi zagotovili, da ne bodo objavljali v ne-prosto dostopnih revijah. To je pripeljalo do ustanovitve

¹² Več o iniciativi <http://www.soros.org/openaccess/read.shtml>

Javne knjižnice znanosti. Večina znanstvenikov je nadaljevalo objavo v teh revijah, oz. objavo osnutkov za ne prosto dostopne revije. Ideja obvezovanja samo-arhiviranja za samo-arhiviranje je bila sporna že od začetka leta 1998. Od leta 2003 dalje je bil trud osredotočen na prosto dostopni mandat vlagateljev raziskav: vlada, vlagalne raziskovalne agencije, univerze. Ti ideje so bile v nasprotju z založniško industrijo. Kakorkoli že, veliko držav, vlagateljev, univerz in drugih organizacij so sedaj ali naredile obveze za prosti dostop, ali pa so v procesu pregledovanja politike in procedur, z namenom, da bi odprli dostop do rezultatov raziskav, za katere so odgovorni.

OA je od takrat postal predmet razprav med znanstvenimi raziskovalci, akademiki, knjižničarji, univerzitetnimi administratorji, skladi, agencijami, vladnimi uslužbenci, komercialnimi in družbenimi založbami. Kljub osnovnemu sporazumu glede koncepta OA, še vedno obstajajo pomembne debate in diskusije o ekonomiji investiranja v državne obveznice revij s prosto dostopnim objavljanjem ter o zanesljivosti in ekonomskih posledicah samega arhiviranja (Budapest Open Access Initiative, 2002)

3.3.2 Vpliv odprte kode na strojno učenje

Strojno učenje je znanstvena disciplina ki se ukvarja z oblikovanjem in razvojem algoritmov, ki omogočajo računalnikom da se učijo na podlagi podatkov kot so podatki zbrani s senzorji ali podatki iz podatkovnih baz. Velik del raziskovalnega področja strojnega učenja se osredotoča na samodejno učenje prepoznavanja kompleksnih vzorcev in njihovo pametno izbiro glede na dane podatke. Torej je področje strojnega učenja tesno povezano s področjem statistike, verjetnostne teorije, podatkovnim rudarjenjem, prepoznavanjem vzorcev, umetno inteligenco in teoretično računalniško znanostjo.

Aplikacije strojnega učenja vključujejo strojno dojetanje, računalniško vizijo, naravno obdelovanje jezikov, prepoznavanje umetnih vzorcev, brskalnike, zdravstvene diagnoze, bio-informatike, kemijske informatike, prepoznavanje prevar z bančnimi karticami, raziskave borznega trga, razvrščanje DNA zaporedij, prepoznavanje govora in pisave, prepoznavanje objektov in inženiringu programske opreme.

Obstaja veliko prednosti v deljenju podatkov in virov in v tem delu bom podrobneje opisal nekatere prednosti odprto kodnega razmišljanja in kaj vse lahko prestop na odprto kodno programsko opremo lahko doprinese znanstveni disciplini, ki združuje tako potrebe kot zahteve znanstvenega prizadevanja in proizvajalca in porabnika programske opreme. Največje prednosti so kategorizirane v (Journal of Machine Learning Research 8, str 2449):

- *Ponovljivost znanstvenih rezultatov in primerna primerjava algoritmov*
- *Odkrivanje problemov*
- *Grajenje na obstoječih virih*
- *Dostop do znanstvenih orodij*
- *Kombiniranje prednosti*

- *Hitrejše prilagajanje metod v različnih znanstvenih disciplinah in industriji*
- *Skupno razvijanje standardov*

Ponovljivost znanstvenih rezultatov in primerna primerjava algoritmov

Ponavljjanje rezultatov je temelj vsake znanosti. Na mnogih področjih znanosti je poskus ali eksperiment veljaven samo ko ga neodvisno ponovi neka druga skupina ali organizacija. Pogosto je težko ponoviti rezultate in raziskovalci vložijo ogromno truda, da bi to dosegli. Na področju strojnega učenja bi bilo rezultate preprosto ponoviti z deljenjem polne kode uporabljene za eksperimente. Na področju strojnega učenja so numerične simulacije pogosto uporabljene, da priskrbijo potrditev in primerjavo metod. Vendar je v strojnem učenju možno zaradi več razlogov nemogoče teoretično analizirati algoritem strojnega učenja. Tako na področju strojnega učenja poskusi igrajo drugačno vlogo kot v drugih naravoslovnih znanostih.

Vseeno so rezultati poskusov enako pomembni kot pri drugih disciplinah. Pri skupnosti strojnega učenja prihaja do veliko površnosti v praksi objavljanja dokumentov, saj v večini primerov niso dovolj podrobni da bi dovoljevali ponovitev rezultata. V preteklosti, pred dobo interneta je bilo to razumljivo, saj bi podrobne opisovanje algoritmov na papirju zahtevalo veliko truda toda danes teh ovir ne bi smelo biti saj bi se celoten material objavil na spletu. Uporaba odprto kodnega pristopa bi bila torej idealna za reševanje tega izziva (Journal of Machine Learning Research 8, str 2449).

Hitrejše odkrivanje in popravljanje napak

Pomembna lastnost, ki je prispevala k uspehu odprte kode je, da pri njeni lahki dostopnosti veliko lažje odkrivanje napak in njihovo popravljanje. Medtem ko vsi ne bi bili naklonjeni ali sposobni popravljanja napak, dostopnost odprte kode omogoča vpogled vsakemu v izvorno kodo programa, odkriti rešitev in priložiti popravek razvijalcem programa ali projekta. Vsaka programska oprema tako licenčna kot odprto kodna vsebuje napake edino kar ju loči je zmožnost popravljanja napak in tu ima vpogled v izvorno kodo velik pomen.

Hitrejši razvoj temelječ na obstoječih metodah

Znanstveni razvoj vedno temelji na raziskovalnem delu predhodnikov in področje strojnega učenja ni nič drugačno. Toda ponovno implementiranje že obstoječih metod je veliko breme za raziskovalce in deloma prihaja do tega zaradi nepopolne dokumentacije predhodnih projektov. Tako je delo na takih metodah omejeno za zaprto skupino ljudi, ki imajo dostop do začetnih implementacij, novinci na področju pa morajo ponoviti delo predhodnikov. Uporaba odprto kodnega modela lahko ogromno pripomore k hitrejšemu razvoju na tem področju.

Dolgoročna dostopnost in podpora

Za posamezne razvijalce to pomeni zagotovilo da bo lahko razvijal nek projekt tudi po menjavi delodajalca, saj imajo mnoge organizacije razne omejitve in zakasnitve na uporabo kode, ki jo je razvijalec razvijal na organizaciji. Za raziskovalca to pomeni zastoj pri delu in sprostitev kode pod odprto kodo bi močno povečalo stalno dostopnost do izvorne kode.

Kombiniranje napredkov

Do znanstvenega napredka ponavadi prihaja v majhnih korakih in ne v velikih preskokih. Do napredka pride z majhnimi izboljšavami že obstoječih metod. Velikokrat se zgodi več majhnih napredkov naenkrat ko neka tema ali problem pride v javnost. Kako združiti te majhne korake v neko skupno nadgradnjo.

Hitrejše prilagajanje metod v različnih znanstvenih disciplinah in industriji

Dostopnost visoko kakovostnih odprto kodnih implementacij lahko olajša prevzemanje teh v strojnem učenju in drugih industrijah predvsem zaradi dveh razlogov in sicer odprto kodna programska oprema ne zahteva učenja in če je neka metoda uspešna in je za njo podana izvorna koda se lahko metodo uporabi v reševanju resničnih situacijah in na drugih področjih industrije.

Skupno razvijanje standardov

Razvejanost na področju strojnega učenja ne dopušča uporabe enotne programske opreme, ki bi zadoščala vsem potrebam tudi na področjih kjer bi v teoriji to bilo mogoče, a je oprema preveč raznolika in ne deluje skladno z drugo opremo. Zaželeno bi bilo postavitve standardov, ki bi omogočali pretok podatkov znotraj skupnosti raziskovalcev na področju strojnega učenja in omogočali sodelovanje različne programske opreme.

3.4 ODPRTO KODNE SKUPNOSTI

Odprto kodne skupnosti (*v nadaljevanju OKS*) so ene najbolj uspešnih in hkrati najslabše razumljenih primerov visoko kakovostnega sodelovanja in grajenja skupnosti preko interneta. Ostale skupnosti bi se lahko ogromno naučile iz razumevanja delovanja odprto kodnih skupnosti. Zadnjih nekaj let je vpliv različnih odprto kodnih programskih rešitev na trg programske opreme očaral veliko ljudi. Najbolj opazna med temi programi sta operacijski sistem Linux in Apache spletni strežnik, ki sta postala resna tekmeča Microsoftu in jih je Microsoft priznal kot resna konkurenta (Pivec, 2006). KS je skupina razvijalcev in uporabnikov odprto kodne programske opreme, ki delijo skupne interese v projektih in ki redno sodelujejo z izmenjavo znanja ter skupaj iščejo rešitve za odprte probleme. (Ye/Nakakoji, 2005). Skupnost deluje po načelih doslednosti, preglednosti ter enakopravnosti.

Člani OKS imajo različne vloge:

- *projektni vodje,*
- *člani jedra (ustvarjalci, komunikatorji, drugi sodelavci),*
- *aktivni uporabniki,*
- *pasivni uporabniki.*

Marsikdo najprej pomisli na računalniške "geeke", ki z debelimi očali zatopljeni in asocialno buljijo v monitor, živeč v svojem svetu in ne meneč se za navadne ljudi.... Ta stereotip že dolgo ne drži. Odprto kodo ustvarjajo uporabniki sami. Posamezniki in podjetja sodelujejo v skupnostih, pomagajo pri iskanju hroščev, prevajanju, pisanju dokumentacije, pomoči uporabnikom na forumih, programiranju, ipd. Ustvarjalci sodelujejo med seboj pri razvoju in si izmenjujejo izkušnje. Odprto kodni programi so

rezultat sodelovanja tudi po nekaj sto ali nekaj tisoč uporabnikov. Še pomembnejše je to, da odprto kodo razvijajo ponavadi *ad-hoc* skupnosti, ki so ponavadi slabo organizirane in jih sestavljajo ljudje ki se še nikoli niso srečali v živo. Nekako tej veliki zmešnjavi ljudi uspe ustvariti izjemno težko nalogo in sicer izgradnja visoko kakovostne programske opreme.

3.4.1 Trg

Odprto kodno skupnost predstavlja večja skupina ljudi, ki izdeluje programje pod standardi ki jih nalagajo določene licence. V teh skupnostih se programerji z različnih delov sveta učijo na že obstoječih aplikacijah. Na internetu so tri bolj obširne takšne skupnosti

- *FSF/UNESCO Free Software Directory*
- *Freshmeat.net*
- *SourceForge.net*

FSF/UNESCO ima označenih 5364 paketov programske opreme in večina programske opreme je pod GPL licenco. Freshmeat.net ima 209.192 programov pri tem odprtih še 42.967 novih projektov ter prijavljenih 384.717 uporabnikov. Sourceforge.net je po številu uporabnikov največja saj jih je prijavljenih kar 1.596.878, ter imajo registriranih 149,391 novih projektov s tem da imajo že 197.361 programov.

Tabela 1: Število paketov programske opreme v posameznem repozitoriju

Repozitorij	Števil paketov programske opreme
SourceForge	197.361
Freshmeat	209.129
GNU Free Software Repository	5.364

Vir: (Kim, 2003).

Od teh treh omenjenih strani Freshmeat¹³ pokriva največji spekter programske opreme in vključuje pakete tako iz Scourgenet¹⁴ kot GNU Free Software. Tudi vsi paketi naštetih na Freshmeat niso odprto kodni programski paketi, in majhno število naštetega je samo dokumentacija in ne programska oprema. Od 20 najbolj priljubljenih projektov jih je okoli 15 namenjenih končnim uporabnikom ostalih 5 pa avtorjem in razvijalcem. Večina jih je ustvarjena samo za uporabo na UNIX platformah medtem ko jih je nekaj namenjenih tudi uporabi na Windows operacijskih sistemih (Kim, 2003).

¹³ <http://freshmeat.net/>

¹⁴ <http://sourceforge.net>

Tabela 2: Primeri bolj znanih in priljubljenih programov s posameznega področja

Pisarna / urejevalniki besedil	Internet	Audio/Video
AbiWord EZ Word GNU TeXmacs Groff KOffice KWord LyX OpenOffice.org Open Blog R4U(Panacea) Ted WordPress	FileZilla, odjemalec za FTP Gaim, izmenjava sporočil Mercury, messenger Miranda IM , izmenjava sporočil Mozilla Firefox Mozilla Thunderbird Nvu, oblikovanje spletnih strani XChat , odjemalec za IRC	Amarok, predvajalnik zvočnih datotek Audacity, snemalnik Beep Media Player CDex, stiskalec glasbe, v slovenščini Media Player Classic VLC media player
P2P in programi Torrent	Protivirusni programi	Grafična orodja
aMule, v slovenščini Azureus, DC++, v slovenščini eMule, v slovenščini Shareaza uTorrent	ClamAV za Linux ClamWin za Windows OpenAntivirus Avast! Antivirus nod32mta	Blender, 3D modeliranje GIMP, urejanje rastrskih slik Inkscape, vektorska grafika Karbon14, vektorska grafika Scribus, namizno založništvo
Operacijski sistemi	Grafična namizja	Stiskanje datotek
Linux, poslovenjen Darwin, derivacija OS X SliTaz, GNU Linux distribucija	GNOME KDE	7-Zip 7-zip, v slovenščini 7-Zip 7-zip, v angleščini

V tabeli 2 sem predstavil za posamezna področja informacijske tehnologije najbolj poznane in uporabljene programe. Gotovo so med bolj znanimi tisti, ki jih uporabljamo sami in to so Mozilla spletni brskalnik razni programi namenjeni predvajanju zabavnih vsebin in programi namenjeni deljenju datotek preko interneta, kot so eMule, Azureus in mnogi drugi.

3.4.2 Značilnosti pripadnikov skupnosti

V preteklih letih je nastalo kar nekaj anket in raziskav na temo kdo razvija odprto kodo in zakaj. Tri take raziskave izstopajo

- *Who Is Doing It (WIDI) Survey (2001)*
- *The Boston Consulting Group/OSDN (BCG/OSDN) Hacker Survey (2002)*
- *Free/Libre and Open Source Software: Survey and Study (FLOSS) (2002)*

Rezultati treh študij so razumno podobni in kažejo na to kdo je tipičen razvijalec odprto kodne programske opreme (Free/Libre and Open Source Software: Survey and Study, 2002).

Med raziskovalci v veliki meri prevladujejo moški, saj je bilo kar 98 odstotkov vprašanih moških (FLOOS, str 8). Preko 70 odstotkov anketiranih je bilo starih med

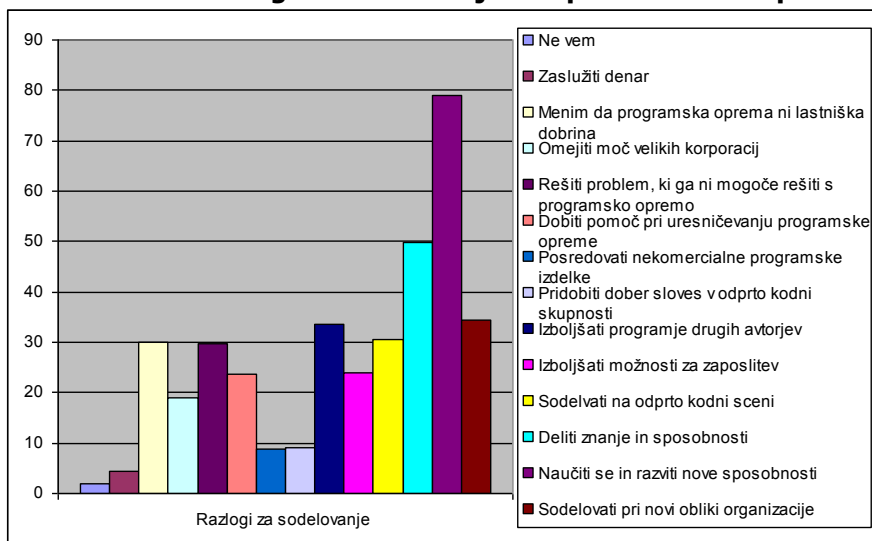
22 in 37 let znotraj teh let pa večina med leti 27 in 30 (FLOSS, str 9). Raziskave so prav tako pokazale da velik del razvijalcev prihaja iz Združenih držav in Evrope, saj je kar 80 odstotkov anketirancev prihajalo iz prej omenjenih področij (FLOSS, str 16-17). Raziskava je tudi pokazala da je večinoma razvijalcev univerzitetno ali srednješolsko izobraženih s kar 30 odstotki anketiranih z magistrsko izobrazbo. Prav tako so bili anketiranci v večini strokovnjaki na področju informacijskih tehnologij in so se z razvijanjem odprte kode ukvarjali le občasno (FLOSS, str12,13,21).

3.4.3 Motivacija

Med temeljnimi gibal človeškega razvoja so predvsem različne potrebe. Tako ločimo primarne potrebe (fiziološke in socialne), ki omogočajo preživetje oziroma ohranitev človeške vrste, od sekundarnih potreb, ki povečujejo zadovoljstvo človeka, če so zadovoljene in ne ogrožajo preživetja, če niso zadovoljene. Ljudje se razlikujemo med seboj po neskončnem številu značilnosti, zato so tudi naše potrebe različne, ne le pri posameznikih, ampak tudi pri isti osebi v različnih časovnih obdobjih in celo trenutkih. (Brejc, str 55).

Nekateri ločijo te potrebe na zunanje in notranje in notranje zadovoljen človek ne išče motivacije v zunanjih. Spodnji graf prikazuje glavne faktorje, ki motivirajo avtorje odprte kodne skupnosti. Vidi se da največjo prednost dajejo učenju in razvijanju novih sposobnosti. Najmanj pozornosti pa namenijo zaslužku.

Graf 1: Glavni razlogi za sodelovanje v odprto kodnih skupnostih



Vir: Free/Libre and Open Source Software: Survey and Study

Po FLOSS raziskavi razvijalce odprto kodne programske opreme žene predvsem želja po uresničitvi notranjih ali sekundarnih potreb. FLOSS raziskava kaže na to da skoraj polovica anketirancev ne zasluži nič denarja, tako direktno kot indirektno za njihovo delo na področju odprte kode 16 odstotkov je plačanih za razvoj odprte kode, 18 odstotkov je plačanih za nameščanje opreme in 12 odstotkov je plačanih da vzdržuje odprto kodne sisteme.

Nadalje, 26 odstotkov razvijalcev je bilo plačanih indirektno in 18 odstotkom je razvijanje odprte kode pomagalo do zaposlitve. Vse raziskave tudi kažejo, da je glavni razlog za razvijanje odprte kode ne želja po zaslužku ampak širiti in združevati znanje. 93 odstotkov anketiranih je trdilo da je širitev znanja eden pomembnih faktorjev in za skoraj polovico je bil to najpomembnejši faktor (Free/Libre and Open Source Software: Survey and Study, 2002)

Poleg prej omenjenih motivov obstajajo še trije glavni motivi, ki so pogosto omenjeni v razlagah o tem zakaj nekdo začne odprto kodni projekt:

- *Potreba po določeni rešitvi v programski opremi*
- *Zabava/igra*
- *Družba talentov ali družbeni ugled*

Motivacija za vsakega posameznega programerja za začetek odprto kodnega projekta je seveda mešanica prej omenjenih razlogov ali motivov. Začetniki ponavadi začnejo projekt, ker niso zadovoljni z obstoječimi rešitvami ali ker določene rešitve sploh ne obstajajo. Linus Torvalds je potreboval UNIX za svoj osebni računalnik in je razvil Linux, Eric Allman je potreboval bol učinkovit strežnik za elektronsko pošto, iz česar je sledil Sendmail¹⁵, Larry Wall je potreboval orodje za avtomatsko ustvarjanje spletnih strani in je razvil Perl, podobno je Don Knuth potreboval priročno orodje za znakovno določene dokumente in nastal je TEX (Knuth, 1979).

¹⁵ Več o Sendmail na <http://www.sendmail.org/>

4 LICENCIRANJE ODPRTE KODE

4.1 AVTORSKE PRAVICE IN IZPELJANA DELA

Da bi spodbudili avtorje k ustvarjanju nove vsebine, različne države odobrijo avtorjem začasen monopol za nadzor posredovanja njihovih del. Temu rečemo avtorske pravice. Če vzamemo primer knjige, ki je zaščiten z avtorski pravici za določen čas. Knjigo lahko kupimo ali beremo ne smemo jo pa kopirati ali posredovati našim prijateljem. Zanimivo je da ima avtor malo nadzora nad samo uporabo vsebine. Če ostanemo pri primeru knjige, ki nam jo je prodal avtor bi ta avtor težko nadziral samo uporabo knjige ali nam vsilil da jo beremo samo v kopalnici ali samo v spalnici. Pravniki tako trdijo da z avtorski pravici nikoli ni bilo mišljeno nadzirati uporabe vsebine le njeno posredovanje naprej.

Če mi sedaj vzamemo to knjigo in dodamo neko poglavje smo ustvarili izpeljano delo originalne knjige. Tako kot je veljalo za originalno knjigo sedaj velja za našo in sicer sami lahko izbiramo kaj bomo počeli z njo in kako jo bomo distribuirali. Lahko tudi odkupimo avtorske pravice originalnega dela in jo posredujemo kot izpeljano delo. Težave večinoma prihajajo pri natančni definiciji izpeljanega dela. Če prepisemo celotno knjigo in dodamo eno poglavje potem je to izpeljano delo, kaj se pa zgodi če nekdo naredi povsem novo knjigo, ki je podobna neki knjigi a ne uporablja enakih besed. To se v praksi določi za vsak primer posamično (Understanding Open-Source Licensing, 2009).

Pomen za odprto kodo

Izvorna koda je podobna vsebina kot knjiga, napisana s strani posameznika ali skupine ljudi. Izvorna koda je zaščiten s strani avtorjev ne glede na to ali prijavijo pravice ali ne. Avtor se lahko odloči kako bo distribuiral izvorno kodo, če to želi. To posredovanje ne da prejemnikom kode nobene druge pravice kot to, da uporabijo to kodo. Nekateri razvijalci tako preprosto uporabijo to metodo. Uporabniki lahko prosto naložijo programsko opremo, toda ne smejo napraviti izpeljana dela in jih posredovati naprej. Tako lahko drugi razvijalci izpeljana dela lahko objavljajo kot popravke, kar je podobno prej omenjenim posameznim poglavjem knjige brez objavljanja originalnega dela. In končni uporabnik je zadolžen da sestavi posamezne dele v neko smiselno celoto.

Ključno je torej da ta nova poglavja (ali v primeru izvorne kode popravki) zaščiten s strani avtorjev teh dodatkov originalnim delom. Tako avtor originalnega dela ne more objaviti popravkov z originalnim delom in avtorji popravkov ali dodatkov ne smejo objaviti njihovega dela z originalnim delom. Splošno gledano so avtorske pravice velika ovira pri ustvarjanju skupnih del, ker vsak avtor lasti pravice svojih koščkov dela in nadzira distribucijo le teh. Če vsi avtorji popolnoma sodelujejo in se strinjajo na končnih pogojih licence je vse v redu, dokler ne pa pride do zmešnjave. Odprto kodna licenca je tako niz redistribucijskih pravil na katere lahko pristane več avtorjev preden začnejo ustvarjati nov skupni projekt. Vsak avtor se mora strinjati s pravili licence preden prispeva k projektu in da bi spremenili posamezna pravila licence same bi se morali strinjati vsi avtorji.

4.2 ODPRTO KODNE LICENCE

Licenca je javni dokument, ki določa kdo, kako in v katere namene se lahko uporablja programska oprema, ki "sodi pod to licenco", kar pomeni, da je kršenje licenčnih pogojev prekršek, za katerega se lahko plača denarna kazen ali nosi druge posledice. V licenci lahko piše, da lahko določen program brezplačno prenesemo z interneta z določene spletne strani in ga brezplačno uporabljamo, ne smemo pa ga ponujati za prenos na svoji spletni strani. Pogosta licenca za dokumentacijo odprto kodne programske opreme npr. pravi, da lahko dokumente prenesemo z interneta, jih prevedemo in priredimo po svoje ter jih ponudimo drugim uporabnikom, vendar mora biti jasno zapisan izvirni avtor dokumenta.

Glavni namen odprto kodnih licenc je preprečiti komurkoli pravico do izključnega izkoriščanja nekega dela ali vsebine. Ponavadi avtorji, če hočejo živeti ali zaslužiti kaj od ustvarjanja del predajo vse pravice nekemu založniku in jim s tem omogočijo izključno izkoriščanje dela. Posledica tega so založniki, ki močno branijo to pravico do avtorskih pravic. Pri programski opremi je drugače saj je programska oprema tako funkcionalna kot dinamična; funkcionalna s tem da program dela in dinamična s tem da ga je možno uporabiti za reševanje drugih problemov. Ko uporabnik kupi programsko opremo dobi kopijo programa in pravico uporabe programa, kot mu naroča licenca in s tem ko uporabnik odpre škatlo izgubi pravico do posredovanja dela, do spreminjanja programa ter kopiranja programa. Odstranitev teh treh omejitev je bistvo odprto kodnega licenciranja (O'Reilly, str. 5).

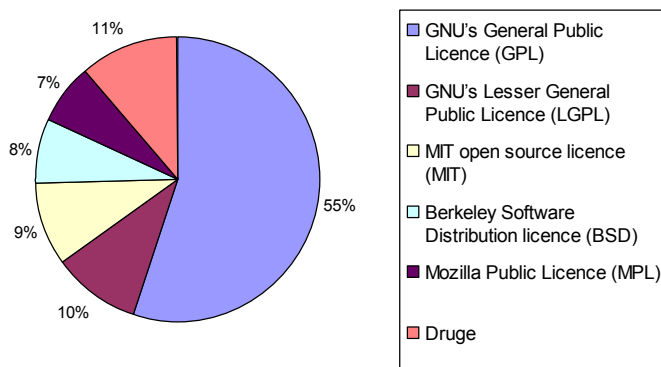
Pravice odprto kodnih licenc:

Dopuščanje uporabnikom dostop do izvorne kode še ne pojasnjuje kakšne pravice ima posameznik do uporabe te kode ali kakšne pravice ima uporabnik do uporabe, spremembe ali distribucije te programske opreme na splošno. Te pravice so definirane v samih licenčnih pogojih. V kontekstu odprto kodnega licenciranja so pravice, ki se razlikujejo od običajnega lastniškega licenciranja spadajo v naslednje kategorije:

- Pravice dostopa do izvorne kode;
- Pravice uporabe izvorne kode: te pravice so lahko omejene z dodatnimi pogoji, ki zagotavljajo združljivost z drugimi programi, večjo varnost, vzdrževanje, izboljšave in spremembe izpeljanih del in možnosti vključevanja v druge programe;
- Pravice kopiranja so lahko neomejene, omejene na licencirano entiteto, ali omejene za določen namen;
- Pravice distribucije so lahko neomejene, omejene na povezane družbe ali sodelavce, omejene glede na obliko in pogoje uporabe;
- Pravice, ki se nanašajo na lastnosti izdelkov (garancij) in odnos do popravkov napak (Ponavadi poskušajo biti odprto kodni izdelki v "as is" ali "kot je" obliki, brez garancije za kompatibilnost nekim specifikacijam ali garancij glede varnosti.

Licence, pod katerim je na voljo odprta koda se precej razlikujejo. Med najbolj poznanimi so GPL licenca, LGPL licenca, BSD licenca IN MPL licenca

Graf 2: Najbolj pogosto uporabljene licence



Vir: Basset, Fitzgerald, 2003

Zaradi prevladovanja GNU standardov v odprto kodni skupnosti prevladujeta predvsem splošna GPL licenca in Lesser GPL licenca, katere se uporablja pri skoraj 70 odstotkih vseh projektov. Sledijo MIT, DSB in MPL, ki so med drugimi najbolj uporabljenimi. Ostale licence so skupno uporabljene v le 10 odstotkih ostalih projektov. Podatki so prikazani v Grafu 2 (Basset, Fitzgerald, 2003).

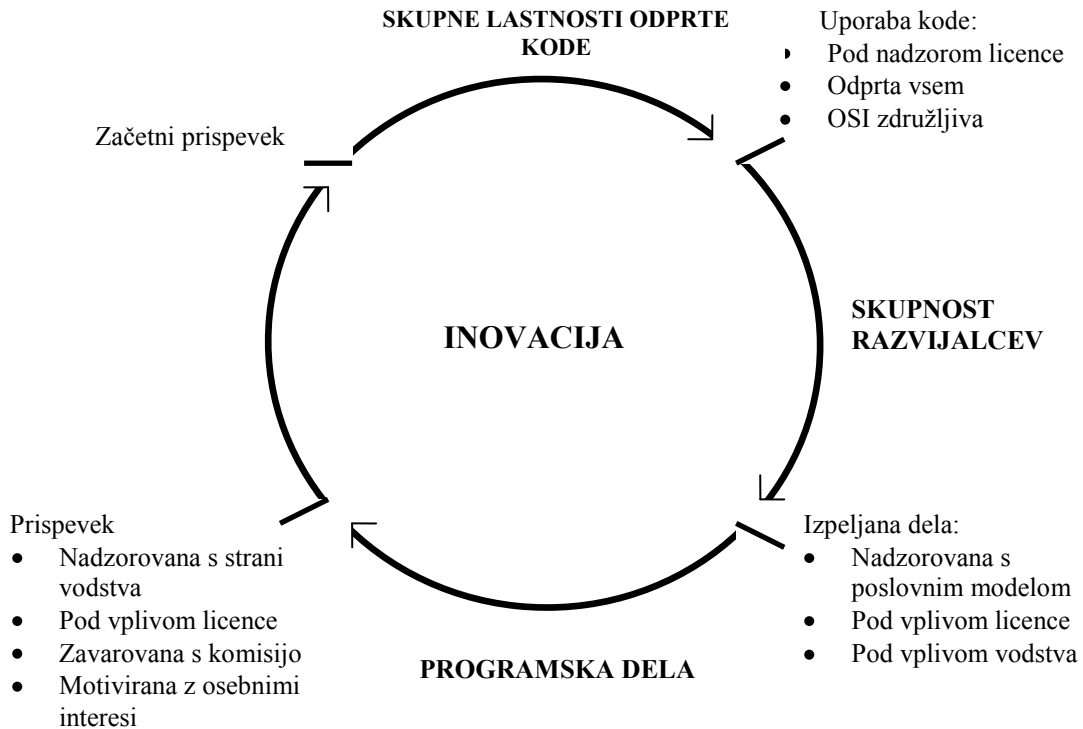
Znotraj odprto kodne skupnosti so se v današnjem času licence razmahnile do točke, kjer v času pisanja tega teksta obstajalo že preko 60 licenc priznanih s strani mednarodne odprto kodne organizacije ali OSI. Veliko število licenc se lahko kategorizira v tri glavne kategorije, glede na attribute ali značilnosti posameznih licenc. Obstoječe licenc lahko predstavlja problem za nekoga, ki ni podrobno seznanjen z odprto kodo, kar mu onemogoča razumevanje posameznih licenc in razlik med samimi licencami še manj pa dovoljuje izbiro za pravo licenco med velikim številom licenc.

Tako je skoraj nemogoče odgovoriti na vprašanja, kot so (Sun Microsystems, *str. 1*):

- *Kako se organizacija odloči glede prostega in odprtega programja*
- *Do kakšne mere vplivajo določeni projekti na odločitve organizacije*
- *Kako na izbire licenciranja vplivajo pogledi skupnosti vezane na posamezne projekte*

Razumevanje teh vprašanj in odgovorov nanje je pomembno pri spoznavanju kako deluje sama odprto kodna skupnost in kako vplivajo odločitve organizacij za posamezne licence na skupnost. Kategorizacija posameznih licenc omogoča lažje opisovanje razlik med samimi licencami, kar omogoča lažje raziskovanje principov in vodil, ki vodijo organizacije k izbiri posameznih licenc. Potrebno se je zavedati, da je izbira odprto kodne licence včasih lahko zelo zapleten proces, tudi zaradi posameznikovih specifičnih potreb in da se je pametno lotiti procesa s pravno pomočjo in ostati v okviru skupnih ciljev posamezne skupnosti.

Slika 3: Neskončni krog uporabe in prispevanja razvoju odprte kode



Vir: (Sun Microsystems, Free and Open Source Licensing)

Do odprtega licenciranja lahko pristopimo s prisposodbo navideznega kroga ali neskončnega kroga (slika 3) v katerem razvijalci delijo skupne imenovalce izvorne kode, ustvarjalnih izpeljanih del dane kode in potem sami prispevajo k samim skupnim imenovalcem v obliki inovacije. Znotraj tega kroga so spremenljivke okoli tega kako avtorji strukturirajo kodo, kako so izpeljana dela ustvarjena iz skupne izvorne kode postopoma licencirana. Ključ odločanju o odprto kodnih licencah za razvoj programske opreme je razumevanje kakšna vrsta skupnosti je zaželena in nato izbrati tip licence, ki je najbolj naklonjena tej skupnosti in njenim skupnim ciljem (Sun Microsystems, str. 2).

4.3 KATEGORIJE ODPRTO KODNIH LICENC

Tri kategorije za prosto in odprto kodno licenciranje temeljijo na prejšnjem delu večjih raziskovalcev vključno z delom organizacije FSF (Free Software Foundation) in OSI (Open Source Initiative), ki sta vodilni na področju določanja lastnosti licenc, pomembnih za nadaljnji razvoj programske opreme. Odprto kodna definicija (Open Source Definition) s strani OSI ima še vedno zelo velik vpliv na posamezne značilnosti različnih licenc in definicija je tako uspešna da je bila obsežna paleta licenc, ki jih poznamo danes podana na ogled in odobravanje OSI komisiji (*Sun Microsystems, str. 3*).

4.3.1 Licence kategorije A

Licence kategorije A ne dajejo nikakršnih obvez, da bi izpeljana ali spremenjena dela morala uporabljati enako licenco, kot je bila uporabljena pri izdaji originalnega dela. Take licence ne spadajo pod Copyleft. Licence so popolnoma neomejene v pogledu licenciranja izpeljanih del. Avtor ali razvijalec delujoč pod licenco A kategorije ima popolno svobodo uporabe izvorne kode in ustvarjanja kakršnih koli izpeljanih ali spremenjenih del na kakršen koli način in ima svobodo objave dela pod katerokoli licenco si avtor to želi. Avtor lahko tako izvorno kodo pod licenco kategorije A spremeni v neko spremenjeno ali izpeljano delo in če želi lahko to delo celo objavi kot komercialno zaprto kodno delo (zasciteno s Copyright licenco).

Seveda je nemogoče trditi da so licence kategorije A popolnoma brez omejitev ali pogojev na avtorje delujoče pod licenco kategorije A. Licence te kategorije še vedno lahko vsili nekatere pogoje, kot so priznanje avtorja originalnega dela ali originalne izvorne kode. Lahko tudi zahtevajo, da razvijalec spremenjenega ali izpeljanega dela odstopa od pravnega pregona originalnega avtorja pod določenimi pogoji. Na primer Apache licenca, ki jo bom podrobneje opisal v naslednjem poglavju, zahteva da je v dokumentaciji izpeljanega dela končnega uporabnika zapisano ali priznано, da izdelek vključuje programsko opremo razvito s strani organizacije Apache (Sun Microsystems, str. 4).

Toda splošno gledano, pogoji licenc kategorije A ne posegajo v obvezovanje avtorjev pod kašno licenco objavijo novo delo. Licence te kategorije se lahko uporabijo za ustvarjanje okolja, kjer postane tehnologija široko uporabljena s strani več končnih uporabnikov in se lahko smatrajo za licence, ki ustvarjajo nove trge. Če povzamemo lastnosti licenc kategorije A imajo te licence naslednje omejitvena nova ali spremenjena dela:

- *Neomejen razvoj novih del*
- *Popolnoma neomejena uporaba licenc za nova ali izpeljana dela*
- *Kateri koli pogoj uporabe so izven licenčnih obveznosti*

Tipi licenc kategorije A ponavadi predstavljajo akademski tip licenciranja, katerega glavni predstavnik je Berkeley Software Distribution (BSD) licenca. Originalna BSD licenca je bila razvita s strani Univerze Berkeley v Kaliforniji za posredovanje njene programske opreme. Danes je verjetno najbolj znana licenca znotraj te kategorije Apache licenca v26. Take licence lahko uporabi kdorkoli za ustvarjanje kateregakoli dela za katerikoli namen, brez omejitev ali obvez prispevanja nazaj originalnim ustvarjalcem.

4.3.2 Licence kategorije B

Tako kot velja za licence kategorije A, velja za licence te kategorije da dovolijo avtorjem uporabo izvornega dela za nastanek kateregakoli izdelka na kakršen koli način. Ampak te licence predvidevajo da bodo nova dela izpeljana iz originalne kode, na koncu postala enotna binarna datoteka in zahtevajo da bodo kakršna koli novo nastala dela objavljena pod enako licenco kot skupna izvorna koda. Toda to velja samo za izpeljana dela ki vsebujejo izvorno kodo originalnega dela in za novo nastala dela, ki ne vsebujejo datotek starega dela velja da so lahko licencirana po želji avtorja ali razvijalca. Take licence so torej že Copyleft licence.

Na licence kategorije B je možno gledati, kot datotečno temelječe licence, saj so zahteve uporabiti isto licenco kot jo je izvorno delo določene z vrsto datoteke pod katero je objavljeno novo delo. Če ta datoteka uporablja enako ali spremenjeno izvorno kodo potem mora uporabiti licenco izvornega dela, če ne potem ni delo obvezano na nobeno drugo licenco. Drugače povedano licence B kategorija je potrebno uporabljati samo za datoteke, ki vsebujejo izvorno ali spremenjeno izvorno kodo in ne pa za lastne ali novo razvite datoteke. S tem se lahko pojavi vprašanje kdaj je neko delo novo ali izpeljano ali samo spremenjeno in ali je s tem podvrženo licencam kategorije B in v tem primeru pride v poštev pride pravna pomoč strokovnjakov. Če torej strnimo značilnosti licenc kategorij B glede na omejitve danih na izpeljana dela je za njih značilno (Sun Microsystems, str. 5):

- *Neomejen razvoj novih ali izpeljanih del*
- *Datotečno licenciranje*
- *Nove datoteke izpeljane iz izvorne kode morajo biti izdane pod enako licenco kot originalno delo*
- *Datoteke, ki ne izhajajo iz originalne izvorne kode so lahko izdane pod katerokoli licenco*

Licence kategorija B vključujejo veliko večino od mnogih prosto in odprto kodo licence, ki obstajajo danes. To so licence, ki so bile ustvarjene na podlagi Mozilla Public License (MPL), ki izhaja iz razvoja originalnega brskalnika Mozilla v poznih 1990-ih. MPL je bila proizvedena na željo, da se zagotovi, da so možne izboljšave našle svojo pot v skupno programsko opremo, medtem ko še vedno odprte možnosti za posameznike za raziskovanje drugih možnosti, povezanih s programsko opremo. SUN je ustanovila licenco v slogu Mozilla licence imenovane Common Development and Distribution License (CDDL), ki se trenutno uporabljajo kot platformo za izdajo licenc za OpenSolaris projekt in Glassfish projekt.

4.3.3 Licence kategorije C

Licence kategorije C tako kot licence kategorije A in kategorije B pušča razvijalcem svobodo da uporabijo izvorno kodo in ustvarijo karkoli na katerikoli način, prav tako kot licence kategorije B zahtevajo da je delo, nastalo iz druge izvorne kode izdano pod isto licenco, dodatno pa zahteva da izpeljano ali novo delo, ne glede na izvorno kodo, ampak je povezano z neko programsko opremo v kakršnih koli drugih okoliščinah mora biti izdano z enako licenco kot delo s katero si deli podobnosti. Cilj tega je napraviti čim več prosto programske opreme dostopno javno in praktično (Sun Microsystems, str. 5).

Medtem ko smo licence kategorije B označili za datotečno temelječe, lahko označimo licence kategorije C temelječe na projektih, ker je licenciranje vseh datotek v projektu pod eno licenco učinek licenc te kategorije. Ta značilnost daje licencam te kategorije močan učinek Copylefta. Če je ena datoteka izpeljana iz enega projekta zaščitene s licenco kategorije C je treba celotno delo objaviti pod enako licenco kot originalno delo. Če povzamemo značilnosti licenc te kategorije na nova ali izpeljana dela je za njih značilno:

- *Neomejeno razvijanje novih del*
- *Licenciranje temelječe na projektih*
- *Dela izhajajoča iz projektov pod licenco kategorijo C morajo biti izdana pod enako licenco kategorije C*

Vsaka od teh kategorij licenc predstavlja možen niz izbir, ki predstavljajo različne potrebe za odprto kodni razvoj in licenciranje. V nadaljevanju si bom pogledal nekaj licenc značilnih za vsako od kategorij in njihov vpliv na skupnost. Najbolj poznana licenca licenc kategorije C ali licenc v slogu GNU je GNU GPL (General Public License). GPL licenca je bila osnova za večina prosto dosegljive programske opreme na svetu, s čimer je postala objekt hvale in skrbi, odvisno iz strani pogleda. GPL licenca je služila kot ščit v vzdrževanju stalnega prostega kroženja programske opreme, čeprav nekateri strokovnjaki trdijo da je GPL licenca prestroga za njihove specifične komercialne motive, saj prepoveduje določene lastniške modele.

Tabela 3: Primerjava posameznih karakteristik vsake kategorije

	Kategorija A	Kategorija B	Kategorija C
Omejitve	Brez omejitev	Datotečno usmerjene	Projektno usmerjene
Izpeljana dela	Izdelava česarkoli	Izpeljana dela morajo biti pod isto licenco	Vse datoteke morajo biti objavljene pod isto licenco če vsebujejo katerekoli datoteke iz izvorne kode
Licenciranje	Brez omejitev licenciranja	Dodane datoteke so lahko pod drugo licenco	Dodana koda mora biti objavljena pod isto licenco
Namen	Tržna ustvarjalnost	Podpiranje skupnosti	Zaščita skupnosti

Vir: Sun Microsystems, Free and Open Source Licensing

4.4 PREGLED ODPRTO KODNIH LICENC

4.4.1 Gnu General Public License (GPL)

Večina licenc programske opreme prepoveduje distribucijo in spreminjanje programja. Nasprotno GPL zagotavlja svobodo distribuiranja in spreminjanja prostega programja in poskrbi za to, da je programje prosto za vse njegove uporabnike. GPL je najbolj pomembna licenca prostega programja, saj je večina prostega programja distribuirana pod njenimi pogoji. Glavni razlog njene popularnosti je njen virusni efekt, to je koncept Copyleft. V preambuli GPL so jasno določene in zaključene njeni trije glavni nameni in sicer obdržati programsko opremo prosto, da je programska oprema taka kot je, brez garancij in da je programska oprema pod to licenco prosta brez omejitev raznih patentov (O'Reilly, str 37).

Glavne značilnosti GPL licence so torej):

- *uporablja se za večino Free Software Foundation programske opreme,*
- *programsko opremo se lahko spreminja ali uporablja njene posamezne dele v drugih odprto kodnih programih (nujna je navedba katere datoteke so spremenjene in datum),*
- *upoštevanje moralnih pravic avtorjev in jasno navedbo le teh ob uporabi programa oz. njegovega dela ali po programsko dokumentacijo (Lugos,2004).*
- *Dovoljuje zaračunavanje stroškov distribucije in garancije*
- *Pravica izdelovanja izpeljanih del za lastne namene*
- *Pravica distribuiranja izpeljanih del ob upoštevanju naslednjih pogojev:*
 - *Označiti delo kot spremenjeno*
 - *Da je licenca pod GPL licenco*
 - *Interaktivno posredovanje licenčnih podatkov če je delo interaktivno*

Če objavimo kodo pod GPL licenco lahko nekdo drug vzame to kodo in lahko spremeni po želji ampak je dolžan objaviti kodo pod isto licenco. Če torej hočemo ustvariti neko skupnost kjer bo več ljudi prispevali k naši kodi, potem mora biti vse pod GPL licenco. Če pa že želimo narediti kaj izven GPL licence moramo biti lastnik originalnih avtorskih pravic, saj ima lastnik originalne kode pravico z njo narediti kar hoče (Fitzgerald, Basset, str. 61).

GPL licenca izraža Copyleft obveznost kot, delo, ki smo ga objavili ali posredovali in vsebuje dele drugega programa ali izhaja iz drugega dela moramo objaviti kot celoto pod isto licenco kot delo iz katerega novo delo izhaja. Te obveznosti se nanašajo na spremenjeno delo kot celoto. Prepoznani kosi tega dela, ki niso izpeljani iz istega programa in ki so lahko prepoznani kot neodvisna in ločena dela od originalnega, potem licenca in njeni pogoji ne veljajo za prej omenjeno delo, če jih posredujemo kot ločena dela. Toda ko posredujemo iste odseke kot del celote, ki je narejeno na osnovi drugega programa, potem mora biti posredovanje celote pod pogoji te licence, katere dovoljenja za druge licence se navezujejo na celoto in s tem na vsak posamezen del ne glede na to kdo ga je napisal (General Public License, 1991).

4.4.2 Lgpl Lesser General Public License (LGPL)

Ta licenca se razlikuje od GPL v enem pomembnem vidiku: dovoljuje povezovanje z ne-prostimi moduli. Prvotno je bila oblikovana za standardne knjižnice, z namenom pospeševanja prisvajanja prostega programja, saj tako licencirane knjižnice nudijo priložnost lastniškemu programom, da se izvajajo v sistemu prostega programja. FSF meni, da omogočanje knjižnic pod LGPL škoduje namenu in načelom prostega programja, zato je ne podpirajo. Veliko postavk LGPL licence je skoraj identičnih GPL licenci in se razlikuje samo v nekaj točkah. Glavne značilnosti LGPL so (Lesser General Public License, 1999):

- programsko opremo se lahko spreminja ali uporablja njene posamezne dele v drugih odprtokodnih programih (nujna je navedba katere datoteke so spremenjene in datum),
- upoštevanje moralnih pravic avtorjev in jasno navedbo le teh ob uporabi programa oz. njegovega dela - nikakor ni možno zmanjševanje pravic (pravice, ki jih da LGPL moraš tudi ti posredovati prejemniku "tvoje" rešitve. Vsaka kopija mora jasno opredeliti avtorske pravice in vsebovati opozorilo o varnosti).

4.4.3 Berkeley Software Distribution (BSD)

Ta licenca dovoljuje, da se s programsko opremo naredi skoraj vse. Najbolj pomembno dovoljenje, ki ga v GPL ni, je da se lahko BSD-licencirane spremembe naredijo zasebne. Drugače povedano, lahko se spreminja pridobljeno izvorno kodo BSD-licenciranega programa in se ga potem prodaja v binarni obliki, ne da bi se poleg distribuirala tudi spremenjena izvorna koda. To je še vedno odprta koda, saj definicija odprte kode ne zahteva, da imajo izpeljana dela originalno licenco (BSD License, 2009). Pomembno pri tej licenci je:

- redistribucija mora obdržati izjavo o avtorstvu, listo pogojev in pogoje uporabe (*disclaimer*¹⁶, ki ga mora vsebovati je jasno naveden),
- Imeni organizacije ali posameznih avtorjev ne smeta biti uporabljeni za promoviranje izdelkov, ki izhajajo iz te programske opreme brez poprejšnjega pisnega soglasja.

¹⁶ Disclaimer je ponavadi katerakoli izjava namenjena opredelitvi obsega pravic ali obveznosti, ki so v veljavi in vsiljena s strani strank v pravnem odnosu ali postopku

4.4.4 Mozilla Public License (MPL)

MPL licenco je razvila združba Netscape Corporation, ko je objavila svoj brskalnik Netscape Navigator v odprti kodi. Posledice truda sta bili NPL in MPL licenci, ki sta si bili v veliki meri zelo podobni. MPL licenca je kot hibrid med načeli GPL in BSD licencami omenjenimi prej (O'Reilly, str.61). Licenca omogoča, da postanejo določene modifikacije programa zasebne. Definicija modifikacije programa v MPL je, da je to vsaka sprememba v datoteki, ki je del originalnega programa, in vsaka nova datoteka, v katero je bil prepisan določen del izvorne kode originalnega programa. Iz tega sledi, da lahko ostanejo zasebne samo datoteke, ki v celoti vsebujejo povsem novo izvorno kodo (ni bila vzeta iz originalnega programa) (Mozilla Public License, 2009). Veliko podjetij je sprejelo variacijo MPL za svoje programe. Tako so nastale licence: Netscape Public Licence, Interbase Licence, Nokia Open Source Licence itd. Za MPL je značilno:

- dopušča tudi: uporabo, reprodukcijo, razstavljanje, uprizarjanje, podlicenciranje,
- odvisno od vrste kode se zahteva, da vsebuje izjavo o avtorstvu in pogoje uporabe,
- ne sme se spreminjati licence izvorne kode; pri zaprti in obsežnejši kodi pa se lahko poleg MPL za izvorno kodo doda še svoja "licenca" (jasna navedba, da so določila drugačna od licenčnih določena posebej), vendar ne sme biti v nasprotju z MPL ali jo omejevati,
- MPL ločuje izvorno kodo in zaprto kodo (nujno vsebuje originalno kodo, ki pa ni nujno ista) ter obsežnejšo kodo; za obe skupini velja isto,
- ni potrebno plačati licenčnine (*royalty-free*)

4.4.5 EUPL v1.0 (European Union Public Licence)

Javna licenca Evropske unije („European Union Public Licence - EUPL“) je bila oblikovana v okviru IDABC (**I**nteroperable **D**elivery of European eGovernment Services to public **A**dministrations, **B**usinesses and **C**itizens) programa Evropske skupnosti, s ciljem spodbuditi interoperabilno zagotavljanje evropskih e-vladnih storitev javnim upravam, podjetjem in državljanom. IDABC je nadaljevanje in nadgradnja prejšnjega IDA („Izmenjava podatkov med upravami“) programa. Programske aplikacije, kot je CIRCA, orodje za delo v skupini, namenjeno izmenjavi podatkov znotraj zaprtih uporabniških skupin, IPM, orodje, ki pomaga upravam, da zapolnijo vrzel med njimi in zainteresiranimi stranmi z zagotavljanjem močnega in hkrati enostavno uporabnega orodja za neposredno posvetovanje preko interneta, ali eLink, orodje, ki vključuje ugotavljanje oddaljenih storitev ter zagotavljanje zanesljivih in varnih storitev pošiljanja sporočil preko omrežne infrastrukture, so bile oblikovane znotraj okvira programov IDA in IDABC (EUPL v.1.0 Preambula, 2007).

Evropska skupnost je na podlagi pogodb, ki so omogočile izdelavo takšne programske opreme, lastnik pravic intelektualne lastnine ter posledično izvorne kode in vseh izvršljivih kod. Takšna orodja, razvita na podlagi IDA ali IDABC, uporabljajo javne uprave zunaj Evropskih institucij na podlagi licence, ki jo izda Evropska komisija, institucija, ki deluje v imenu Evropske skupnosti, ker ima slednja avtorske pravice za ta orodja. Interesi za objavo izvorne kode programske opreme na podlagi licence, ki ne bi omejila dostopa in sprememb izvorne kode, so se že pred časom povečali. Prvotna licenca EUPL za takšno programsko opremo je bila oblikovana v skladu s cilji IDABC. Licenca je napisana na splošno in jo je zato mogoče uporabiti za izpeljana dela, za druga dela in uporabljajo jo lahko drugi dajalci licence. Korist, ki jo ta licenca prinaša, je okrepitev pravnega razumevanja s sprejetjem skupnega okvira za združevanje programske opreme javnega sektorja.

4.5 DRUGE ODPRTO KODNE LICENCE

MIT (Massachusetts Institute of Technology) Licenca

- dopušča tudi: spajanje programske opreme, objavo, podlicenciranje, in/ali prodajo licenc
- vse, kar licenca dopušča, mora nujno vsebovati izjavo o avtorstvu in dovoljenje za uporabo
- brezplačna. (MIT License, 2009)

Python Software Foundation License 2.1.1 (PYTHON)

- dopušča tudi: reprodukcijo, analizo, testiranje, javno uprizorjanje, pripravljati izvedena dela,
- kadar izvedeno delo temelji na ali ima pripojen/vključen Python 2.1.1 ali njegov del je potrebno vključiti povzetek sprememb narejenih na Python 2.1.1. PSF ni odgovoren za poškodbe, ki nastanejo kot rezultat modifikacije, distribucije, ali kake druge uporabe Python 2.1.1,
- ni potrebno plačati licenčnine (royalty-free). (Python Software Foundation License, 2009)

The Artistic License (ARTISTIC)

- vsebovati mora original copyright notice in povezane pogoje uporabe,
- ob določenih vrstah modifikacij in distribucij postavlja katere pogoje morajo take spremembe zadostiti,
- dopušča razumno nadomestilo ("reasonable" fee). (Artistic License, 2009)

Apache HTTP Server Version 2.1 (APACHE)

- redistribucija mora obdržati izjavo o avtorstvu, listo pogojev in pogoje uporabe (disclaimer, ki ga mora vsebovati je jasno naveden). Imeni "Apache" in "Apache Software Foundation" ne smeta biti uporabljeni za promoviranje izdelkov, ki izhajajo iz te programske opreme brez poprejšnjega soglasja (Apache License 2.0,).

Common Public License (CPL)

- dopušča tudi: reprodukcijo, pripravljati izvedena dela, javno zastavljanje in uprizarjanje del, pod licenciranje,
- distribucija je omogočena, če se upošteva pogoje licence,
- ni potrebno plačati licenčnine (royalty-free). (Common Public License, 2009)

Zlib License (ZLIB)

- zelo "svobodna" licenca, ki prepoveduje edino zlorabe,
- v nobenem primeru avtorji ne odgovarjajo za poškodbe, ki izhajajo iz uporabe te opreme(OKO, 2004).¹⁷

¹⁷ Več o licencah in seznamu vseh licenc na strani <http://www.opensource.org/licenses>

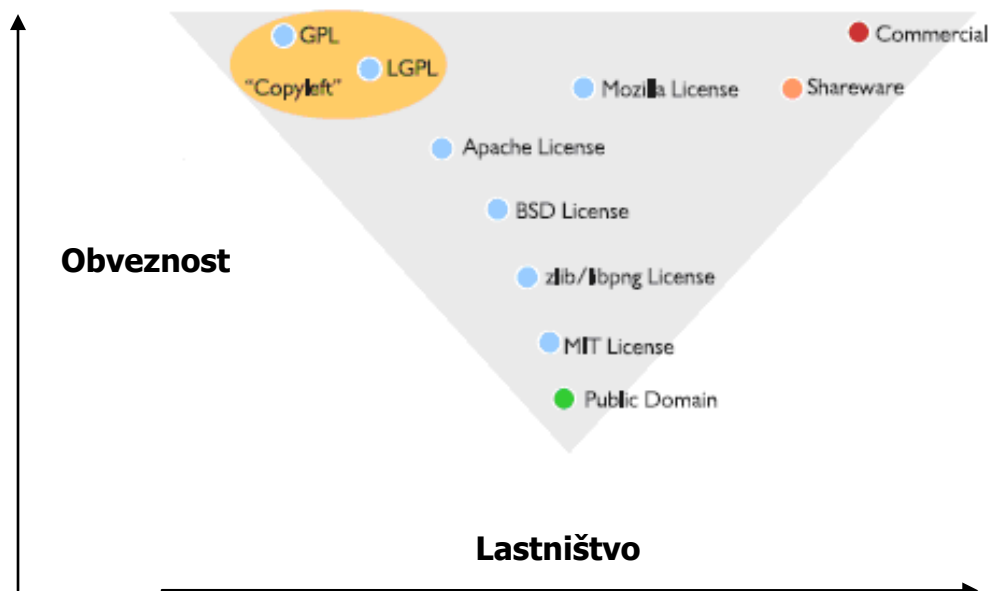
4.6 PRIMERJAVA LICENC

V tem poglavju si bom ogledal kaj je skupno prej opisanim licencam in kašne so podobnosti in razlike med posameznimi, kaj je treba upoštevati pri izbiri različnih licenc in kje lahko prihaja do težav pri licenciranju odprte kode. Osredotočili se bomo tudi na težave do katerih lahko prihaja pri odprto kodnem licenciranju in težav, ki nastajajo pri izbiri odprto kodne licence pod katero želimo objaviti določeno delo.

4.6.1 Skupne lastnosti licenc:

- Svoboda kopiranja, deljenja t.j. distribuiranja in spreminja proste programske opreme;
- Vse zahtevajo, da se navede avtor kopiranega dela, kakor tudi avtor modificiranega dela - s tem se zagotovi, da posledic, ki bi lahko nastale zaradi spreminjanja izvirne kode, ne nosi avtor originala.
- Vse našteje licence so komercialne v smislu, da se lahko svobodno uporabljajo, če upoštevajo navedene pogoje.
- "Prosto" v smislu svobode pa ne pomeni, da je licenca nujno brezplačna (licence uporabljajo različne termine): v petih primerih plačljivosti ne določajo; MIT licenca določa, da je brezplačna; ARTISTIC licenca dopušča sprejemljivo ceno; tri pa določajo, da za uporabo ni treba plačati licenčnine.

Slika 4: Različni vidiki odprto kodnih licenc



(Vir.: *Open for Business: Executive Summary, 2004*)

Slika 4, ki služi kot nek pregled odprto kodnih licenc, predstavlja tri kote kot tri primarne tipe ali kategorije licenc, in sicer GPL licence, komercialne licence in javna domena. Bolj ko se od kotov premikamo proti notranjosti vidimo različne kombinacije licenc. GPL in komercialne nosijo največ licenčnih obveznosti a sta na nasprotnih

koncih trikotnika zaradi njunih pogledov in pravil na izvorno kodo, lastništvo in dostopnost. V javni domeni pa sploh ne poznamo licenc (Open for Business: Executive Summary, 2004)

4.6.2 Recipročna obveznost (Reciprocal Obligation)

Recipročna obveznost pomeni obojestransko ali sodelujočo izmenjavo uslug ali pravic. Po tem ko je nekaj izpeljano ali občuteno z obeh strani ima povratno obveznost. Tako ima GPL licenca recipročno obveznost saj jo občutita obe strani, tako licenciranec in tisti, ki je delo izdal pod neko licenco. Ta obveznost je v GPL licenci napisana kot Copyleft obveznost. Obojestranska obveznost je značilna v odprto kodnih licencah in GPL licenca je le prva in najbolj vplivna v uveljavljanju tega elementa (Rosen, str 105). Vsaka licenca izraža to obveznost drugače in na različne načine in v nadaljevanju si bom ogledal razlike v tej obveznosti med še nekaterimi drugimi licencami kot so LGPL, MPL in CPL licencami.

LGPL licenca uvaja izraz izpeljana dela, ki lahko vključuje kombinacijo med LGPL licencirano vsebino in vsebino, ki ni pod LGPL licenco. LGPL licenca tako zahteva, da so taka kombinirana dela, ki vključujejo LGPL licencirano izvorno kodo ob distribuciji opremljena z informacijami o izvorni kodi. To je lahko pomembna ovira pri uporabnosti LGPL licencirane vsebine s kombiniranjem z licenčno vsebino in to lahko predstavlja težave pri nadaljnjem razvoju programske opreme.

MPL licenca ne uporablja izraza izpeljana dela ampak samo spremembe originalni kodi kot dela nekega izpeljanega dela. To naredi MPL licenco bolj dojemljivo pravnim strokovnjakom in s tem bolj zanesljivo s tega vidika. Toda to tudi omeji povratno obveznost MPL licence. MPL dovoli da je izpeljana koda ali predelana koda distribuirana znotraj večjih del brez omejitve licenciranja celotne kode tudi nove kode pod MPL licenco. Ta večja prostost MPL licence daje prednost pred GPL in LGPL licencama za razvijalce in znanstvenike, ki želijo združiti MPL licencirano kodo s svojimi lastniškimi programi.

CPL licenca kot MPL licenca prav tako govori o spremembah k originalnemu delu, toda CPL licenca ima še ožje omejitve glede recipročne obveznosti saj dodatno preprečuje vpliv povratne obveznosti na ločena dela. Povedano drugače CPL obveznost velja samo za spremembe ali dodatke k originalni kodi, ki so izpeljana dela ne pa ločenim modelom programske opreme (Journal of Machine Learning Research št 8, str 2461).

Torej lahko vidimo da ko neko delo licenciramo pod neko licenco, ki ima recipročno obveznost, to delo ne moremo uporabiti v izpeljanem delu če delo ne bo objavljeno pod isto licenco z enako obratno obveznostjo in prav to je glavni princip povratne obveznosti licenc in je predstavljen v spodnji tabeli (Rosen, str 244).

Tabela 4: Različne kombinacije recipročnosti licenc

	Licenca	Izpeljano delo			
		GPL	MPL	CPL	OSL
Prispevek	GPL	+	-	-	-
	MPL	-	+	-	-
	CPL	-	-	+	-
	OSL	-	-	-	+

Vir: (Rosen, str 244).

Tabela 4 prikazuje da ko enkrat začnemo delati na nekem prispevku pod GPL, MPL ali CPL licenco moramo končni produkt objaviti pod enako licenco. Seveda ne velja za vse licence enako in pri nekaterih prihaja do odstopanj med posameznimi licencami. Nekatero licence kot so MPL in CPL otežujeta to razlago s osredotočanjem na prispevke in ne na izpeljana dela. Taki prispevki niso nujno ločeno licencirani in je biti treba previden pri izbiri posamezne licence za posamezen prispevek (Rosen, str 244).

4.6.3 Izbor licence

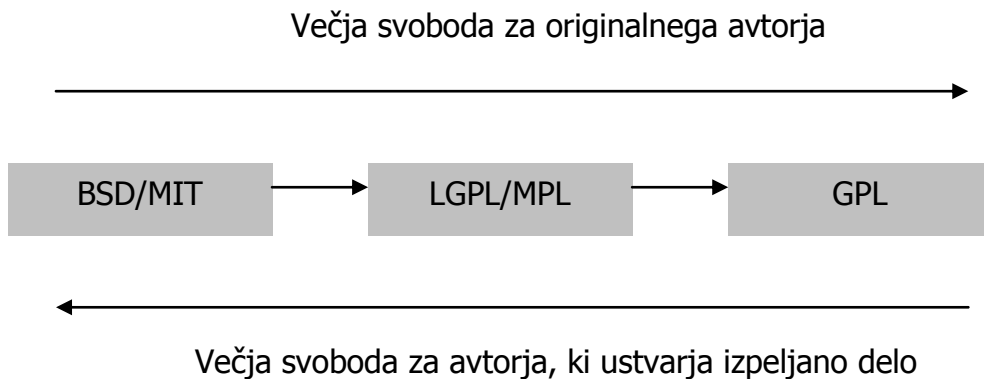
Največ težav imajo avtorji pri izbiri licence, ko prihaja do distribucije izpeljanih ali spremenjenih del. Z razvojem odprto kodne programske opreme je prišlo do razvoja več licenc, ki lahko zmedejo avtorje, ki želijo dati program v javnost. Kljub temu, da se izbira licence zdi trivialna težava, kateri ni vredno posvetiti veliko pozornosti ima lahko izbira prave ali napačne licence velik pomen za vpliv nove programske opreme. Pri izbiri odprto kodne licence se organizacijo pogosto osredotočijo na dva pomembna faktorja (Rosen, str 229):

1. *Kako lahko organizacija zasluži z distribucijo odprto kodnih projektov oziroma katera licenca nam bo omogočila prodajo prostega programja.*
2. *Kako lahko drugim preprečiti neupravičen zaslužek iz naše lastne kode. Stanje, ko so uporabniki upravičeni do vseh ugodnosti brez obranih obveznosti*

Lastnik avtorskih pravic je vedno avtor samega dela in on narekuje kako in pod katero licenco bo delo objavljeno. Največ težav pride v primeru, ko gre za prepletanje odprte in licenčne programske opreme. Pri odprto kodnih licencah poznamo licence, ki skrbijo da odprta koda ostane odprta in licence, ki ne omejujejo avtorjev izvornih del in jim dovoljujejo objavo dela tudi pod zaprto licenco. Licence, ki obvezujejo da odprta koda ostane odprta koda so licence pod tako imenovanim Copyleftom. Najbolj znana med njimi je tako splošna GPL licenca.

Obstajajo tako bolj in manj stroge licence. Najbolj stroga je tako GPL licenca, ki obvezuje avtorja da bo delo tudi ostalo odprto manj strožje so pa BSD/MIT licenci, ki ne obvezujejo avtorjem spremenjenih ali izpeljanih del pri objavi pod isto licenco, kot je originalno delo. S strani avtorja ali razvijalca je stvar obrnjena. S tem, ko je GPL strožja, zagotavlja avtorju da bo prejemal obvestila in popravke o programski opremi in da bo izvorna koda ostala odprta med tem ko manj strožje tega ne zagotavljajo (Journal of Machine Learning Research, št. 8, str 2459).

Slika 5: Prikaz svobode originalnega avtorja in avtorja izpeljanega dela



Vir: Journal of Machine Learning Research št. 8, str 2459

Med obema koncema lestvice poznamo licence, ki so nekje vmes, kot so licence LGPL (Lesser GPL), CPL (Common Public License) in MPL (Mozilla Public License), ki zahtevajo od avtorjev samo objavo sprememb ali popravkov. Tako imajo avtorji dostop do nadaljnjih sprememb.

Vsekakor velja da je najpomembnejša izbira pri odločanju pod katero licenco bomo objavili neko delo odločitev med GPL licencami in manj obvezujočimi licencami. GPL temelji na prepričanju, da se je treba izogibati lastniške programske opreme in spodbujanju ustvarjanja novih odprto kodnih projektov, ki bi spodbudili uporabo odprtih modelov in odvrčali od odvisnosti na projekte, ki niso bili razviti pod odprto kodo. (O'Reilly, str. 176).

Prednost drugih bolj odprtih licenc, kot so MPL, BSD, MIT in Apache licenc je v tem, da odprt model razvoja programske opreme ni neenakovreden razvoju v drugih modelih, tudi lastniškem. Dejstvo je da tudi če je program razvit pod lastniško licenco, ni njegov pomen za odprto kodo nič manjši. Več kot je udeležencev in razvijalcev na neki kodi boljše je za razvoj te kode, čeprav je tudi nekaj razvoja pod zaprtimi ali licenčnimi licencami.

4.6.4 Težave pri odprto kodnem licenciranju

Več težav nastaja pri odprto kodnem licenciranju in v nadaljevanju bom omenil nekaj pomembnejših (Basset, Fitzgerald, 2003).

Vloga vodje odprto kodnega projekta

Ponavadi ni zelo jasen pravni položaj organizacije, ki se ukvarja in koordinira trud programerjev in usklajuje skupne dosežke neke skupnosti na odprto kodnem področju.

Uveljavitev avtorskih pravic in licenc

Ko obstaja skupna programska oprema s stotimi avtorji nastajajo težave okoli obstoja skupnih avtorskih pravic teh avtorjev v rokah vodje projekta in kdo bo uveljavljal avtorske pravice tega skupnega truda. Obstajajo tudi težave pri postopkih v primeru kršitve teh pravic in kako bi bilo veliko število avtorjev združenih v nekem pravnem postopku. Obstaja tudi veliko težav okoli problema, ko je več ljudi prispevalo k skupnem delu projektu različne velikosti, kdo je prispeval več.

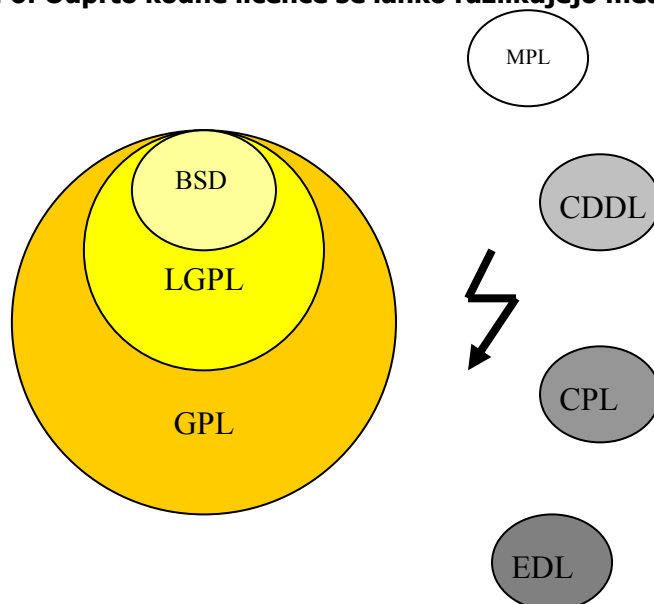
Napraviti licence obvezujoče

Licence programske opreme v odprto kodnih projektih so redko podpisani z licenco. Sama programska oprema je posredovana preko interneta z obvestilom pogojev licence pod katero je objavljena. Način s katerim je to narejeno je v vsakem primeru drugačno in nujno vedno da uporabnik lahko pride do programske opreme le po sprejetju pravil licence skozi formalen "click trough" postopek. To dviguje dvom okoli tega ali bodo pogoji licence sploh obvezujoče ali ne.

Odprto kodne licence si lahko nasprotujejo

Ko objavimo nek program kot odprto kodne še ne pomeni da je postal "odprt" in da je brez omejitev. Obstaja možnost da je program še vedno pod licenco, ki je v nasprotju z drugimi licencami. Licence imajo lahko izključujoče interese glede pristojnosti in imajo lahko različne klavzule. OSI trenutno priznava 60 licenc in posledica tega razvejanja licenc pomeni da ne moremo vseh licenc združiti kot kaže slika 4 za licence BSD, LGPL in GPL licence. Licenci MPL in CDL sta v konfliktu z najbolj razširjeno uporabljenimi licencami kot sta GPL in LGPL. Med tem ko se to lahko uporabi namenoma za ustvarjanje konfliktov, lahko to močno omejuje avtorje in upočasnjuje napredek na področju celotnega področja odprte kode. Splošno velja pravilo da avtorji uporabijo licence združljive z GPL ali drugimi licencami s podporo močne skupnosti (Journal of Machine Learning Research št. 8, str 2459).

Slika 6: Odprto kodne licence se lahko razlikujejo med sabo



Vir: (Journal of Machine Learning Research št. 8, str 2459)

Neuporaba licence kompatibilne z GPL lahko povzroči težave pri podpori od drugih ustvarjalcev pri odprto kodnih projektih. Tudi če nam ni vseč GPL licenca je velika verjetnost da je drugim razvijalcem in projekt zna biti bolj uspešen če sodelujemo z njimi, saj če drugi avtorji ne bodo sodelovali z nami obstaja verjetnost da bodo začeli razvijati konkurenčno programsko opremo.

4.6.4.1 Spor zaradi licenc (SCO vs. IBM)

Marca 2003 je skupina SCO (prej Caldera Systems Inc) začela pravni postopek proti IBM na zveznem sodišču v Združenih državah. SCO domneva da je naslednik v naslovu pravic in interesov v UNIXu, ki izhaja od AT&T skozi serijo poslovnih pridobitev in s tem nadzira pravice vseh UNIX prodajalcev vključno z IBM za uporabo in distribucijo UNIXa. Razlogi skupne SCO za postopek proti IBM izhaja iz njenih domnev da je IBM nepravilno uporabil kodo in znanje razvito s strani SCO v razvoju nekaterih pogledov jedra operacijskega sistema Linux. V svoji je SCO zahtevalo 3 milijarde dolarjev odškodnine (kasneje celo 5 milijard), domnevajoč da je IBM kršil pravila in pogoje več dogovorov o programski opremi v povezavi z izvorno kodo različice UNIXa »Unix System V« oziroma s kopiranjem in prevzemanjem te kode v Linux jedro ter da se je IBM spustil v nepravično konkurenco s pomočjo razvoja Linuxa. SCO tudi domneva, da si je IBM polastil poslovne skrivnosti skupine SCO, še posebej znanja in modela razvitega za pogon sistema temelječega na Unixu na Intelove procesorje (Basset, Fitzgerald, 2003)

IBM je nasprotoval obtožbam z domnevo in argumentacijo, da je skupina SCO kršila pogoje dogovora programske opreme s tem ko so trdili da bodo ukinili IBM-ove trajne in nepreklicne pravice in da je SCO javno napačno predstavil legitimnost IBM Linux povezanih izdelkov in storitev, kar je v nasprotju z Lanham aktom in da je SCO kršil 4 IBM-ove patente programske opreme. IBM je trdil, da je SCO z distribuiranjem

Linuxovih produktov kršil pogoje GPL licence pod katero je obvezan ne vsiljevanja lastniških pravic preko linux izvirne kode.

Obtožbe obeh strani imata resne posledice na odprto kodno skupnost. SCO je objavil uveljavitve plačila licenc za komercialne uporabnike GNU/Linux sistemov. Če bi bilo to sprejeto, da ima SCO moč zaračunavati licence za obstoječe uporabnike, lahko opazimo še večjo negotovost in upočasnitev prevzemanja programske opreme na trgu, zaradi organizacij, ki se ne želijo izpostaviti lastniškim obveznostim, ki niso jasno postavljene že na začetku. Eben Moglen, glavni svetovalec FSF (Free Software Foundation) opazuje, da je nemogoče presoditi težo neobjavljenih dokazov. Trdi da se postavlja veliko vprašanj glede pravnih trditev skupine SCO in da vidi veliko razlogov za zavrnitev pravnih obtožb skupine SCO.

Najpomembnejše pri tej tožbi je da obtožbe obeh strani lahko pripeljejo do dolgo pričakovane pravne razlage GPL licence, vključno z razpravami o njeni klasifikaciji (ali je licenca ali pogodba), njeni nepreklicnosti, uveljavljanju in odgovornosti tretjih strank (Basset, Fitzgerald, 2003).

Po drugi strani pa sodni spor kaže tudi na to, da nekaterim ponudnikom Unixa počasi popušča potrpljenje in nameravajo svoje pravice ter morda tudi zaslužek, izgubljen zaradi Linuxa, povrniti tudi preko pravne poti. Pri tem morda preseneča, da je prvi pravno možnost izrabil prav SCO Group, ki je med drugim eden od ponudnikov Linuxa in partner v združenju UnitedLinux. Toda razumeti je treba, da ima podjetje za sabo kar nekaj slabih četrletij in jim zaradi izgub morda celo grozi konec.

5 ZAKLJUČEK

Prevzemanje odprte kode tako v poslovnem svetu, kot tudi znanstveni sferi postaja vse pogostejše in predstavlja veliko priložnost za podjetnike in znanstvenike. Da pa lahko uporabniki prevzamejo in začnejo uporabljati odprto kodne rešitve se morajo najprej zavedati, kaj dejansko lahko počnejo z dano programsko opremo ali vsebino, na kakšne načine jo lahko uporabijo in kako jo lahko posredujejo naprej.

Odprto koda se je izkazala da ima veliko prednosti pri njeni uporabi. Njene prednosti so predvsem večja stabilnost zanesljivost in hitrejše popraviljanje napak, saj ima veliko uporabnikov dostop do izvorne kode in tako lahko kar sami popravijo nastale napake dokaj hitro. Prevzemanje odprto kodnega modela se je izkazalo tudi za pozitivno v znanosti in v nalogi sem podal primer strojnega učenja in pozitivne učinke ki jih ima lahko prevzemanje odprte kode. V znanstveni sferi je trenutno trend objavljanja literature in vsebin pod odprto kodnim modelom, saj ta olajšuje dostop literature znanstvenikom in močno lajša njihovo delo. Tako vidimo vse večje število prosto dostopnih blogov in objav na internetu.

Da bi razumeli kaj pravzaprav lahko počnemo s tako odprto vsebino ali kodo moramo poznati licence in pravice in dolžnosti, ki izhajajo iz licenc pripetih na samo odprto vsebino. V tej diplomski nalogi sem tako kategoriziral licence v tri kategorije od najbolj strogih do najbolj prostih. Najbolj obvezujoče so splošne GPL licence, manj obvezujoče pa BSD in MPL licence, ki dopuščajo več svobode pri uporabi in posredovanju odprte vsebine. Za katero licenco se posameznik odloči je na konci vedno odvisno od njega in kaj si želi da nastane od same vsebine. V kolikor želi da ostane pod oznako odprta bo uporabil bolj obvezujočo licenco v nasprotnem primeru pa manj obvezujoče. Odgovora na vprašanje katera licenca je prava ali katera je najboljša še do danes ni in verjetno dokler bo obstajalo veliko različnih motivov posameznikov bo to vprašanje ostalo neodgovorjeno.

Druge pomembnejše težave so že v tem da je na izbiro množice licenc, ki so si pogosto podobne ali se prekrivajo. Na uradni strani OSI je naštetih kar 60 licenc, priznanih iz strani te organizacije in to povzroča veliko zmešnjavo pri izbiri licence. Težave, ki nastajajo pri licenciranju odprtih vsebin so še v tem da ponavadi ni jasnega vodje odprto kodnega projekta, da ni nekega uveljavljanja pravic odprto kodnih licenc in da pogosto ni nobenega, ki bi napravil licence obvezujoče. Menim, da bi bilo v korist tako poslovne kot akademske sfere pomembno združitev teh manjših licenc in pokrivajočih si licenc v večje splošno uporabljene licence. S tem bi zmanjšali zmešnjavo na področju licenciranja. Prav tako k nastali situaciji ne pripomore razkol v odprto kodni skupnosti med pripadniki Free Software Foundation(FSF) in Open Source Initiative (OSI).

Privržence prve organizacije predvsem moti sprejetje tržnih pravilnosti druge organizacije. FSF zagovarja GPL licenco kot neko splošno, ki bi morala biti uporabljena v večini primerov in s tem zagotavljati da odprta koda ostane odprta. Sprejetje nekih skupnih pravil na tem področju in združitev manjših licenc v večje bi po mojem mnenju močno pripomoglo pri reševanju nastalih nejasnosti na področju odprto kodnega licenciranja.

LITERATURA

1. Rosen, Lawrence. Open Source Licensing. Prentice Hall, New jersey, 2004
2. Laurent, Andrew. Understanding Open Source and Free Software Licensing. O'Reilly Media, 2004
3. Lerner, Tirole. The simple economics of open source. URL= »<http://www.hbs.edu/research/facpubs/workingpapers/papers2/9900/00-059.pdf>«, 10.8.2000
4. Fitzgerald, Basset. Legal Issues Relating to Freeand Open Source Software. URL=»http://www.law.qut.edu.au/files/open_source_book.pdf«, Brisbane, 2003
5. Polančič, Horvat. Prednosti in slabosti uporabe odprto-kodnih ogrodij. URL = »http://lisa.uni.mb.si/%7Epolancic/si/raziskovalnoDelo/publikacije/Prednosti_in_slabosti_uporabe_odprtokodnih_ogrodij.pdf« 2004
6. Björk, Bo-Christopher. Open Source, Open Science, Open Courseware. Swedish School of Economics and Business Administration, Finland, 2001
7. Smeh, Aljoša. Odprta koda-priložnost ali nevarnost. Comtron e-izdaja, Ljubljana, 2005
8. Sinur, Alojzij. Odprta koda skozi čas. Kiberpipin računalniški muzej, Ljubljana, 2008
9. Peeling, Satchell. Analysis of the Impact of Open Source Software. Qinetiq, 2001 URL= »http://www.govtalk.gov.uk/documents/QinetiQ_OSS_rep.pdf«
10. Vetter, Greg. Claiming copyleft in open source software: what if the free software foundation's general public license (gpl) had been patented. Houston, 2007 URL= »http://opensource.mit.edu/papers/Vetter.Claiming_CopyleftInOSS-What_IfGPL_Patented.2008Mich.St.L.Rev.279.pdf«
11. Sonnenburg, Braun, Ong, Bengio, Bottou, Holmes, LeCun, Müller, Pereira, Rasmussen, Rätsch, Schölkopf, Smola, Vincent, Weston, Williamson. The Need for Open Source Software in Machine Learning. Journal of Machine Learning Research, št. 8, str. 2444-2466, 2007
12. Brejc, Miha. Ljudje in organizacija v javni upravi, Fakulteta za upravo, Ljubljana, 2004

13. Pivec, Franci. *Politična ekonomija odprte kode*. Zavod IPROK, Maribor, 2005
14. Kim, E. *An introduction to open source communities*. Blue Oxen Associates, April 2003 URL = »<http://opensource.mit.edu/papers/blueoxen.pdf>«
15. Moore, Gordon. *Craming more components onto integrated circuits*. *Electronics*, Volume 38, Number 8, April 19, 1965
16. Weeler, David. *Make Your Open Source Software GPL-Compatible. Or Else*. URL= »<http://www.dwheeler.com/essays/gpl-compatible.html>«, 2002
17. Stallman, Richard. *GNU manifesto*. URL= »<http://www.gnu.org/gnu/manifesto.html>«, 1985
18. Raymond, Eric. *The Art of UNIX Programming*. Addison-wesley, 2004

VIRI

19. Free Software Foundation URL= »<http://www.fsf.org/about/what-is-free-software>«, 8.8.2009
20. GNU Operating System URL= »<http://www.gnu.org/gnu>«, 10.8. 2009
21. Free Software Licensing Resources URL= »<http://www.fsf.org/licensing/education>, 15.8.2009
22. Open Source Initiative URL= »<http://www.opensource.org>«, 13.8.2009
23. Open Source Licenses URL= »<http://www.opensource.org/licenses>«, 15.8.2009
24. Open Source Licence URL= »http://en.wikipedia.org/wiki/Open_source_license«, 8.9.2009
25. Open Source, Wikipedia URL= »http://en.wikipedia.org/wiki/Open_source«,
26. Center odprte kode Slovenija URL= »http://www.coks.si/index.php5/Glavna_stran«, 9.8.2009
27. Benefits of using open source URL= »<http://open-source.gbdirect.co.uk/migration/benefit.html>«, 10.9.2009
28. Understanding Open-Source Licensing URL= »<http://openacs.org/about/licensing/open-source-licensing>«, 11.8.2009
29. Berlecon Research GmbH, *Free/Libre and Open Source Software: Survey and Study*, URL = »<http://www.infonomics.nl/FLOSS/report>«, 12.8.2002
30. *Budapest Open Acces Initiative*, URL= »<http://www.soros.org/openaccess/read.shtml>«, 12.8.2002
31. Sun Microsystems, *Free and open source licensing*, White paper, URL = »http://www.sun.com/software/opensource/whitepapers/Sun_Microsystems_OpenSource_Licensing.pdf«, 15.8.2009
32. *Predstavitev projekta OKO* »http://www.zrss.si/pdf/MAT_Predstavitev_projekta_OKO.pdf« 22.8.2009

KAZALO SLIK

Slika 1: Različne kategorije programske opreme.....	4
Slika 2: Vrste strežnikov, ki prevladujejo na trgu (Avgust 1995 - April 2007)	8
Slika 3: Neskončni krog uporabe in prispevanja razvoju odprte kode	32
Slika 4: Različni vidiki odprto kodnih licenc.....	41
Slika 5: Prikaz svobode originalnega avtorja in avtorja izpeljanega dela	44
Slika 6: Odprto kodne licence se lahko razlikujejo med sabo	46

KAZALO TABEL

Tabela 1: Število paketov programske opreme v posameznem repozitoriju	25
Tabela 2: Primeri bolj znanih in priljubljenih programov s posameznega področja ..	26
Tabela 3: Primerjava posameznih karakteristik vsake kategorije	35
Tabela 4: Različne kombinacije recipročnosti licenc	43

IZJAVA O AVTORSTVU IN NAVEDBA LEKTORJA

Študent Sašo Zupanec izjavljam da sem avtor tega diplomskega dela z naslovom Odprto kodne programske rešitve in njihovo licenciranje, ki sem ga izdelal samostojno pod mentorstvom docenta dr. Ljupča Todorovskega in dovolim njegovo objavo na fakultetnih spletnih straneh.
Lektor diplomskega dela je Jure Šink.

V Ljubljani 1.9.2009

Podpis: _____