

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N**

Danijel Galić

Moj OpenSource projekt na temu „Forum“

ZAVRŠNI RAD

Varaždin, 2010.

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N**

Danijel Galić

Redoviti student

Broj indeksa: 34793/2005.

Smjer: Informacijski sustavi

Preddiplomski studij

Moj OpenSource projekt na temu „Forum“

ZAVRŠNI RAD

Mentor:

mr. sc. Markus Schatten, asistent

Varaždin, veljača 2010.

Sadržaj

Sadržaj

1. Uvod	1
2. Open Source	2
2.1 Povijest open sourcea	4
2.2. Open Source Initiative	6
2.3. Open Source licence	7
2.4. Prednosti i mane open sourcea	11
3. PHP	14
3.1. Web tehnologije	18
3.1.1. Serverske tehnologije.....	18
3.1.2. Klijentske tehnologije	19
3.1.3. Skriptni jezik	20
3.2. Apache mrežni HTTP server	22
4. MySQL.....	23
5. SourceForge.net.....	26
5.1. SVN.....	26
5.1.1. Modeli kontrole verzija.....	30
5.1.2. Revizije	32
5.2. Wiki.....	33
6. Primjer	36
6.1. MySQL.....	36
6.2. PHP	37
6.3. SF.net	44
6.4. SVN.....	45
6.4.1. SVN u Command Promptu.....	45
6.4.2 SVN na SF.net-u	49
6.5. Wiki.....	51
7. Procjena alata	52
7.1. PHP i ostali konkurenti.....	52
7.2. MySQL i PostgreSQL	52
7.3. SVN i CVS	54
8. Zaključak	55
Literatura.....	56

1. Uvod

Tema ovog rada je „Moj OpenSource projekt na temu „Forum“. Tu temu sam odabrao prvenstveno zbog toga što sam htio temu u kojoj se programira i u kojoj se koriste baze podataka. Odabrao sam i programski jezik i sustav za upravljanje bazama podataka sa kojima nisam prije radio, zbog želje da naučim nešto. Zbog tih razloga u početku rada nisam ni davao veliku pozornost pojmu open source, koji me naknadno jako oduševio.

Pod pojmom open source se podrazumjeva programski kod koji je slobodan za upotrebu široj javnosti. To znači da sa njim mogu raditi što god žele, bilo da ga koriste u privatne ili komercijalne svrhe. Danas u svijetu gotovo da nema osobe koja nije čula za pojam open source, te se njegova primjena sve više koristi.

U ovom radu ćemo u prvom dijelu obraditi open source, te ćemo pritom spomenuti sve njegove karakteristike. Obradit ćemo povjest open sourca, gdje ćemo vidjeti kako je nastajao od samog početka informatizacije čovječanstva. Zatim, ćemo spomenuti neke od glavnih licenci koje se koriste zajedno sa open source programima, te pri tom spomenuti grupu i dokument kojim se licence održavaju. Na kraju dijela o open sourcu ćemo vidjeti i sve njegove prednosti, ali i mane.

U drugom dijelu ćemo obraditi open source program koji sam koristio za izradu foruma, kao primjera u ovom radu. Taj programski jezik je PHP. Spomenut ćemo njegove glavne karakteristike, prednosti i mane. U tom dijelu ćemo obraditi i mrežne tehnologije, budući da PHP koristi te mrežne tehnologije. U tom dijelu ćemo obraditi i Apache mrežni server, budući da se većinom uz PHP koristi Apache kao mrežni server. Apache je također open source program.

U trećem dijelu ćemo obraditi MySQL koj je open source sustav za upravljanjem bazama podataka. MySQL ne predstavlja općenito najbolji sustav za upravljanje bazama podataka, ali je se u kombinaciji sa PHP najbolji za mrežnu tehnologiju, te se tako najčešće i koristi.

U četvrtom djelu ćemo objasniti internetsku stranicu za open source projekte, a to je SourceForge.net. Unutar tog dijela ćemo objasniti open source sustav za kontrolu verzije SVN i open source sustav za dokumentaciju wiki.

U petom djelu ćemo prikazati primjenu svega prethodno navedenog kroz primjer foruma. Kratko ćemo predstaviti forum, te primjenu PHP-a, MySQL-a, SF.net-a, SVN-a i wikija.

Nakon toga ćemo procijeniti alate uz usporedbu sa njihovim open source konkurentima, pa ćemo u zaključku ocijeniti open source kao tehnologiju.

2. Open Source

Open Source je pojam koji predstavlja sav software koji se distribuira zajedno sa svojim izvornim kodom, koji je otvoren i pristupačan svakom korisniku. Još jedan naziv za takav software je i free software, gdje free ne znači besplatno, već slobodno. To znači da korisnici takav software mogu slobodno koristiti kako žele, za bilo kakvu svrhu, na koliko god računala žele, i u bilo kakvoj situaciji. Imaju pravo na izmjenjivanje koda, dodavanje nove funkcionalnosti, ispravljanja bugova, te proučavanja. Također imaju pravo na redistribuciju. Razmjena programerskog znanja najbolja je za učenje iz tuđeg rada, te se tako svoj doprinos može ponuditi na korištenje drugima na opću korist. Razmjena dokumenata u otvorenom formatu pospješuje i pojednostavljuje suradnju i razmjenu informacija. Pitanja specifične prilagodbe, lokalizacije, integracije različitih programskih rješenja, uporaba na različitim platformama postaju rješiva.

Svaki stvoreni program može biti open source: od operativnih sustava, preko www i e-mail poslužitelja i klijenata, uredskih programa do programa bilo kakve specifične namjene. Najpoznatiji takvi programi danas su operativni sustavi (Linux), web poslužitelji (Apache), DNS poslužitelji (BIND), web i email preglednici (Mozilla), uredski alati (OpenOffice) itd.

Kao što je već iz početka teksta vidljivo open source ima neke svoje karakteristike. Četiri glavne karakteristike open source softwarea su:

1. Izvorni kod stoji slobodno na raspolaganju: Ili je već uključen u samoj software distribuciji, ili je na drugačiji način napravljen pristupačnim (npr. putem Interneta)
2. Slobodna daljnja distribucija trećim licima: Što se tiče distribucije softwarea, ne postoji ograničenje od strane autora (ili više njih).
3. Proizvoljne modifikacije: Može se vršiti proizvoljna modifikacija, kao i ponovna redistribucija softwarea. Autor (ili više njih) mogu zahtijevati izmijenjenu verziju softwarea u vidu posebnog patch-a kako bi se ista integrirala u originalnu verziju.
4. Neograničeno korištenje softwarea. Korištenje jednog softwarea se ne mora vezati za korištenje ili nekorisćenje nekog drugog softwarea.

Osnovna ideja koja stoji iza open sourcea nije izrađivanje slobodnih (beznaknadnih) softwarea, nego razrađivanje visoko kvalitetnih softwarea i izbjegavanje izrade softwareskih komponenti uvijek na novo ukoliko iste već postoje (ovo je često slučaj sa Closed Source Software). U poređenju sa open source softwarea, postoje drugi tipovi softwarea koji će ukratko biti objašnjeni. Bitno je poznavati razlike između slijedećeg:

Freeware - Autor programa daje pravo za javno korištenje i daljnu distribuciju. Izvorni kod nije uvijek na raspolaganju javnosti. Ukoliko je izvorni kod dodijeljen javnosti na korištenje, u većini slučajeva nije dodijeljeno pravo za modifikacije istog. Autor može postaviti bilo koja ograničenja. Često autor dozvoljava komercijalno korištenje programa ili postavlja posebnu dozvolu za komercijalno korištenje. Dobro poznati primjer za freeware je Acrobat Reader.

Shareware - Shareware predstavlja tek posebni model za promociju komercijalnog softwarea. Korisnik može testirati software izvjesni vremenski period (uglavnom 30 dana). Kada istekne probni period, korisnik mora kupiti dozvolu za daljnji rad ukoliko je odlučio nastaviti sa korištenjem softwarea.

Komercijalni software - Komercijalni software je izrađen sa ciljem stvaranja profita. Klauzule iz licence o korištenju definiraju isto. Korisnik mora platiti licencu za korištenje ukoliko želi raditi na danom softwareu. U većini slučajeva nije dozvoljena daljnja distribucija niti je izvorni kod stavljen na raspolaganje.

U većini zemalja, pa tako i u Hrvatskoj, automatski imate prava na svoj rad. Licenca je autorov način dopuštanja korištenja njegovog djela (u ovom slučaju softwarea) drugima, na njemu prihvatljive načine. Ostaje na autoru da uključi licencu koja određuje na koje se načine software smije koristiti. Naravno, različite okolnosti traže različite licence.

„Software tvrtke žele zaštititi svoju imovinu pa objavljuju samo kompajlirani kod (kojeg ljudi ne mogu čitati) i na korištenje softwarea stavljaju mnoga ograničenja. S druge strane, autori slobodnog softwarea općenito traže neku mješavinu sljedećega:

1. **Nedopuštanje korištenja njihovog koda u vlasničkom softwareu** - Budući da svoj kod objavljuju kako bi ga svi mogli koristiti, ne žele vidjeti da ga netko krađe. U ovom slučaju, korištenje koda se smatra pitanjem povjerenja: smijete ga koristiti dok god igrate po istim pravilima.
2. **Zaštita identiteta autorstva koda** - Ljudima njihovo djelo predstavlja velik ponos i ne žele da netko drugi može iz njega ukloniti njihovo ime ili tvrditi da ga je on napisao.
3. **Distribucija izvornog koda** - Jedan od problema s većinom komercijalnog koda jest to što ne možete ispraviti bugove ili ga prilagoditi sebi jer izvorni kod nije dostupan. Također, tvrtka može odlučiti prestati podržavati hardver kojeg koristite. Mnoge slobodne licence zahtijevaju distribuciju izvornog koda. To štiti korisnika omogućivši mu prilagođavanje softwarea svojim potrebama. To također ima druge posljedice koje ćemo poslije raspraviti.

4. **Prisiljavanje svakog djela koje uključuje dio njihova djela na korištenje iste licence**
- Mnogi ljudi pišu vlastite licence. Na ovo se ne gleda s odobravanjem jer pisanje licence koja čini ono što želite uključuje sitničava pitanja. Prečesto je korišteni slijed riječi ili dvosmislen ili ljudi stvore uvjete koji si međusobno proturječe. Pisanje licence koja bi vrijedila na sudu još je teže. Na sreću, postoji više već napisanih licenci koje vjerojatno čine ono što želite. “ **[Balatinac, 2009., str. 2.,]**

”...Primjena vlasničkih standarda dovodi do ograničenja tržišne utakmice i često korisnika veže uz jednog ili ograničeni skup dobavljača. Takva ograničenja treba izbjegavati kada je god to moguće. Nasuprot tome, primjena otvorenih standarda omogućava interoperabilnost i modularnost rješenja, te olakšava razmjenu podataka čime se proširuje izbor rješenja za operacijske sustave i aplikacije...” **[www.e-hrvatska.hr, 2010.]**

2.1 Povijest open sourcea

Kada su se prvi put počeli stvarati programi postojao je samo slobodni software. Kasnije je razvijen vlasnički software koji je ubrzo preuzeo vodeću ulogu. To je dovelo do točke kada se vlasnički software smatrao jedinim modelom. Tek nedavno je softwareska industrija ponovo počela razmatrati slobodni software kao opciju.

Kada je IBM 60-tih godina prošlog stoljeća prodavao svoja prva komercijalna računala, ta su računala dolazila zajedno sa softwareom otvorenog izvornog koda. Krajem 60-tih situacija se mijenja, IBM, zaštićuje svoj software i sredinom 70-tih već je uobičajeno na računalima vlasnički software, u smislu da ga korisnici ne smiju redistribuirati, nemaju izvorni kod i mogućnost izmjene.

Na kraju 70-tih i u početku 80-tih, se uspostavljaju korijeni open sourcea. Na američkoj istočnoj obali, Richard Stallman, bivši programer u MIT AI Labu, pokrenuo je GNU¹ Projekt i Free Software Foundation. Konačni cilj GNU Projekta je bio izgradnja slobodnog operacijskog sustava. Richard Stallman je započeo sa kodiranjem nekih programskih alata (kompajler, editor i sl.). Kao zakonski alat, GPL je izrađena ne samo kako bi osigurala da software koji je izrađeno od strane GNU Projekta ostane slobodan, već i da promovira proizvodnju slobodnog softwarea. S filozofskog stajališta, Richard Stallman je napisao GNU Manifest u kojem izražava da dostupnost izvornog koda i sloboda za distribucijom i izmjenom softwarea su osnovna prava.

¹ Kratica GNU je rekurzivna kratica za „GNU nije Unix“

CSRG² kalifornijskog Sveučilišta u Berkeleyu je poboljšavala Unix sustav i pravila mnoge aplikacije za njega, te je tako nastao BSD³ Unix. Ti naponi su bili financirani od strane DARPA⁴ kontakata i guste mreže Unix hakera diljem svijeta koji su pomogli u debugiranju, održavanju i poboljšavanju sustava. Duže vrijeme taj software nije redistribuiran izvan zajednice pod Unix AT&T⁵ licencom. No, kasnih 80-tih, konačno je distribuiran pod BSD licencom, jednom od prvih open source licenci. Nažalost, u to vrijeme, svaki korisnik BSD Unixa je također trebao i AT&T Unix licencu, budući da su neki dijelovi kernela i nekoliko važnih alata, koji su potrebni za upotrebljiv sustav, bili pod vlasničkom licencom. Također jedan od bitnih open source projekata tog vremena bio je TeX (sustav za pisanje, razvijen od strane Donalda Knutha). Izrađen je od strane snažne zajednice i postoji danas.

Nakon toga open source software je nastavio svoj razvoj, u početku u nekoliko izoliranih skupina. Izgradila se velika zajednica korisnika koju su pomogli USENET i Internet. Polako, većina softwarea već razvijenog je bila integrirana, ujedinjujući rad tih skupina. Kao rezultat te integracije, čitava okruženja su se mogla napraviti na temelju Unixa koristeći open source licence. U mnogim slučajevima, sistemski administratori su čak zamijenili standardne alate za GNU programe. U to vrijeme, mnoge aplikacije su već bile najbolje u svom području (Unix alati, kompajleri, itd.). Posebno je zanimljiv slučaj X Window Systema, koji je jedan od prvih open source softwarea poduprt (financijski) od strane konzorcija tvrtki. Tijekom 1991-1992 dogodila su se dva važna događaja za zajednicu otvorenog, slobodnog softwarea. U Kaliforniji, Bill Jolitz je implementirao nedostajuće dijelove za kompletiranje Net/2 distribucije kako bi se mogla pokrenuti na i386 strojevima. Net/2 je bio rezultat rada CSRG-a kako bi napravili verziju BSD Unixa bez AT&T-ovog autorskog koda. Jolitz je svoj rad nazvao 386BSD koji je ubrzo postao cijenjen među BSD i Unix zajednicama. Uključivao je ne samo kernel, već i mnoge alate, čineći cijeli operacijski sustav. 386BSD je bio pokriven BSD licencom koja ga je učinila potpuno slobodnom softwareskom platformom. Također je uključivao i slobodni software drugih licenci (primjerice GNU kompajler). U Finskoj, Linus Torvalds, student računarskih znanosti, nezadovoljan Tanenbaumovim Minixom, je implementirao prve verzije Linux kernela. Ubrzo su mnogi ljudi sudjelovali kako bi učinili kernel upotrebljivijim te dodavanjem alata kompletirali GNU/Linux, pravi operacijski sustav. Kernel Linuxa, kao i GNU aplikacije pokriveni su GNU GPL licencom. 1993. i GNU/Linux i 386BSD se relativno stabilne platforme. Od tada, 386BSD je evoluirao u obitelj BSD baziranih operacijskih sustava: NetBSD, FreeBSD, OpenBSD, dok se

² Computer Science Research Group

³ Berkeley Software Distribution

⁴ Defense Advanced Research Projects Agency

⁵ American Telephone & Telegraph

Linux kernel razvija i koristi u mnogim GNU/Linux distribucijama: Slackware, Debian, Red Hat, SuSE, Mandrake i sl.

1998. u Kaliforniji je održan sastanak nakon kojeg se pojavio termin open source. Sastanak je održan 3. veljače, a na njemu su prisustvovali Todd Anderson, Chris Peterson iz Foresight Instituta, John Hall i Larry Augustin iz Linux International, Sam Ockman iz Silicon Valley Linux User's Group i Eric Raymond. Na sastanku se raspravljalo o strategiji iskorištavanja ozračja koje je nastalo priopćenjem Netscape korporacije o objavljivanju izvornog koda njihovog Internet preglednika. Naime, sudionici sastanka su Netscapeov potez vidjeli kao idealnu priliku za popularizaciju open source razvojne metode softwera u korporacijskom svijetu. Zaključak je bio da se treba riješiti konfrontacijskog stava, koji je prije bio povezan sa *free software* retorikom, i koristiti pragmaticne, poslovne argumente. U raspravi o taktici daljnjeg djelovanja, Chris Peterson je skovao termin *Open Source*. U narednom vremenu, radilo se na širenju vijesti o projektu, a uskoro se s projektom povezuju i novi ljudi - Linus Torvalds daje svoj "bлагослов" već idući dan, Bruce Perens registrira "open source" kao trademark i nudi *hosting* OSI sitea, Phil Hughes nudi suradnju sa Linux Journal-om, a čak je i sam Richard Stallman pričao o open source terminu, ali se predomislio. U mnogim granama, slobodni software je već sad najbolji izbor (primjerice Apache Web Server, kojega koristi više od 50% tržišta).

2.2. Open Source Initiative

Open Source Initiative (OSI) je neprofitna korporacija posvećena upravljanju i promociji Open Source Definition kriterija licencnih uvjeta, "za dobro zajednice", kako se navodi na Open Source Initiative stranicama. Temeljni način djelovanja je provedba OSI Certified Open Source Software certifikacijskog programa, a brinu se i za održavanje baze odobrenih open source licenci.

OSI program se temelji na zamisli, da, ako programeri imaju slobodu čitanja, izmjene i redistribucije softwareskog koda, računalni software će se razvijati brže nego što je to slučaj sa konvencionalnim metodama razvoja softwera. Za podršku toj tezi navodi se razlika u broju osoba koje, na taj način, sudjeluju u razvoju softwera, poboljšavajući i prilagođujući ga, te ispravljajući propuste.

Open Source Initiative pokušava ideju open source softwera popularizirati u komercijalnom svijetu, i njihovi najveći naponi su usmjereni u tom smjeru. Međutim, iako su

temeljni principi utvrđeni, implikacije open source principa u komercijalnom okruženju još uvijek se razmatraju, pa i na ovom polju OSI ostaje vjeran "otvorenosti" prema javnosti, dočekujući konstruktivne prijedloge otvorenih ruku.

2.3. Open Source licence

Dokument koji sadrži kriterije prema kojima se vrši certificiranje licenci za naziv Open Source License zove se The Open Source Definition. Nastao je, u početnom obliku, pod nazivom The Debian Free Software Guidelines, a okvirni sadržaj je napisao Bruce Perens. Dokument je dopunjen prijedlozima razvojnih timova Debian GNU/Linux distribucija, na e-mail konferenciji koja je trajala tijekom većeg dijela lipnja 1997. Zatim je, glasovanjem, dokument usvojen kao Debianovo javno stajalište u tom pogledu. U veljači 1998., na inicijativu OSI, dokument je revidiran i izmjenjeni su dijelovi vezani za Debian, te je nastao The Open Source Definition.

“Najbitnije kriteriji koji se nalaze unutar Open Source Definitiona, a koje kandidatkinje za naslov Open Source License moraju ispuniti su:

Slobodna redistribucija - Licenca neće ograničavati niti jednu stranku u prodaji ili poklanjanju softwarea kao komponente ili zbirne softwareske distribucije, sastavljene od različitih izvora. Licenca neće zahtijevati nikakve oblike plaćanja za takvu prodaju.

Izvorni kod - Program mora sadržavati izvorni kod i mora dozvoliti distribuciju u obliku izvornog koda, kao i u kompajliranom obliku. Ako neki oblik proizvoda nije distribuiran sa izvornim kodom, mora postojati dobro dokumentiran način pribavljanja koda, za razumnu cijenu pribavljanja - preporučljivo putem downloada sa Interneta, bez naplate. Izvorni kod mora biti preferirani način izmjene programa. S ciljem prikrivanja namjerno zakompliciran izvorni kod nije dopušten. Međuoblici, poput izlaza preprocesora ili prevodioca, nisu dopušteni.

Izvedeni oblici - Licenca mora dopustiti izmjene i izvedene oblike programa, i mora dopustiti njihovu redistribuciju pod istim uvjetima koje propisuje licenca originalnog softwarea.

Integritet autorovog izvornog koda - Licenca smije ograničiti distribuiranje izvornog koda u izmjenjenom obliku samo ako licenca dopušta distribuciju "zakrpa" (patch file) uz izvorni kod, s ciljem izmjene programa za vrijeme "izgradnje" (build time). Licenca mora eksplicitno dopustiti distribuciju softwarea izgrađenog od izmjenjenog izvornog koda. Licenca može zahtijevati da izvedeni oblici nose naziv ili oznaku inačice različite od originalne verzije.

Diskriminacija osoba i skupina ili područja rada – Budući da cijeli svijet koristi open source Licenca ne smije vršiti diskriminaciju bilo koje osobe ili skupine osoba. Licenca ne smije

nikoga ograničavati u iskorištavanju programa na određenom području rada. Primjerice, ne smije ograničavati korištenje programa u poslu, ili u korištenju za genetski inženjering

Distribucija licence - Ne smij se izdavati više licenci za jedan program kako bi se ograničila prava određenim osobama, već sve osobe moraju imati jednaka prava potrebe. Prava vezana za program ne smiju ovisiti o tome je li program dio određene softwareske distribucije. Ako je program izvađen iz te distribucije, i korišten ili distribuiran unutar uvjeta licence programa, sve stranke kojima je program redistribuiran trebaju imati ista prava kao ona koja su osigurana u sprezi sa originalnom distribucijom softwarea.

Licenca ne smije ograničavati drugi software - Licenca ne smije stavljati ograničenja na drugi software koji se distribuira zajedno sa licenciranim softwareom. Primjerice, licenca ne smije zahtijevati da svi drugi programi distribuirani na istome mediju budu open-source software.

Licenca mora biti tehnološki neutralna - Nijedna stavka licence ne smije biti zasnovana na bilo kojoj pojedinoj tehnologiji ili stilu sučelja.” [www.opensource.org, 2010.]

Kao znak raspoznavanja open source softwarea, OSI je registrirao certificacijsku oznaku **OSI Certified**, kao i graficku certificacijsku oznaku. Korištenje ovih oznaka je dopušteno na bilo kojem softwareu koji se distribuira pod OSI-odobrenom licencom. OSI certificirani software mora imati jednu od tri slijedeće naznake:

This software is OSI Certified Open Source Software.

OSI Certified is a certification mark of the Open Source Initiative

Kraca oznaka glasi: **OSI Certified Open Source Software**

Podljednja naznaka je graficki simbol prikazan na slici 2.1.



Slika 2.1. Grafički simbol koji predstavlja OSI certificirani software [www.opensource.org, 2010.]

Kako je već rečeno, OSI održava bazu odobrenih licenci, koje se mogu koristiti kao uvjeti korištenja i distribucije softwarea - potrebno je samo odabrati licencu koja najbolje odgovara prirodi softwarea i željama autora, a ako takva licenca ne postoji, OSI nudi mogućnost certificiranja novih licenci.

Trenutno se na OSI stranici mogu pronaći slijedeće licence: Academic Free License, Apache Software License, Apache License, 2.0, Apple Public Source License, Artistic license, Attribution Assurance Licenses, BSD license, Common Public License, CUA Office Public License Version 1.0, EU DataGrid Software License, Eiffel Forum License, Eiffel Forum License V2.0, Entessa Public License, Fair License, Framework License, GNU General Public License (GPL), GNU Library or "Lesser" General Public License (LGPL), Lucent Public License (Plan9), Lucent Public License Version 1.02, IBM Public License, Intel Open Source License, Historical Permission Notice and Disclaimer, Jabber Open Source License, MIT license, MITRE Collaborative Virtual Workspace License (CVW License), Motosoto License, Mozilla Public License 1.0 (MPL), Mozilla Public License 1.1 (MPL), Naumen Public License, Nethack General Public License, Nokia Open Source License, OCLC Research Public License 2.0, Open Group Test Suite License, Open Software License, PHP License, Python license (CNRI Python License), Python Software Foundation License, Qt Public License (QPL), RealNetworks Public Source License V1.0, Reciprocal Public License, Ricoh Source Code Public License, Sleepycat License, Sun Industry Standards Source License (SISSL), Sun Public License, Sybase Open Watcom Public License 1.0, *University of Illinois/NCSA Open Source License, Vovida Software License v. 1.0, W3C License, wxWindows Library License, X.Net License, Zope Public License, zlib/libpng license

Svatko može zatražiti izdavanje nove licence, te navesti razloge za to, a osnovna razlika između navedenih licenci je obično u važnosti koju autor daje slijedećim pitanjima:

- **Zaštita otvorenosti** - Neke licence postavljaju pravilo da bilo koji redistributor zadrži tu istu licencu. Stoga su primateljeva prava ista, bilo da je software primio direktno od autora ili od posrednika.
- **Zaštita moralnih prava** - Pod ovo spadaju autorska prava koja licence štite.
- **Zaštita vlasničkih prava** - Originalni autor prema nekim licencama ima vlasnička prava.
- **Kompatibilnost sa vlasničkim licencama** – Kompatibilnost je potrebna jer kod nekompatibilnih licenci može biti zabranjena redistribucija bilo kakvog softwarea koji je rezultat softwarea koji je pokriven vlasničkom licencom, a da se to nije željelo.

- **Kompatibilnost s ostalim open source licencama** – Licence se mogu kositi jedna sa drugom zato jer uvjeti jedne se ne mogu ispuniti ako se ispune uvjeti druge. U tom slučaju, najčešće je nemoguće spojiti software koji sadrži nekompatibilne softwareske licence.

Pojasnimo поближе neke od open source licenca:

- **BSD** - BSD licenca pokriva, između ostalog softwarea, BSD-ove proizvode. Ona je dobar primjer „dopustive“ licence, jer gotovo nikako ne uvjetuje korisniku što može činiti sa softwareom, uključujući naplaćivanje klijentima za binarne distribucije, bez obveze za uključivanjem izvornog koda. Redistributori mogu sa takvim softwareom učiniti što god žele, uključujući i korištenje za vlasničke proizvode. Autori traže samo priznavanje autorstva. U određenom smislu, takvo ograničenje osigurava određenu količinu besplatnog marketinga. Važno je uočiti da ovakva licenca ne garantira autoru da će i derivirana djela također ostati open source.
- **GPL**⁶ - Pod ovom licencom se distribuira software GNU projekta. To može dovesti do pogrešnog zaključka da se pod ovom licencom distribuira samo GNU projekat. Pod ovom licencom se distribuira i software koji nije razvijen od strane GNU projekta (primjerice kernel Linuxa). GPL je pažljivo razvijena kako bi promovirala proizvodnju slobodnog softwarea te stoga eksplicitno zabranjuje sve radnje koje bi mogle dovesti do ugradnje softwarea zaštićenog tog licencom u vlasnički software. GPL je bazirana na međunarodnom zakonodavstvu o autorskim pravima, što osigurava njenu primjenu. Osnovne karakteristike GPL-a su slijedeće. Dozvoljava se redistribucija izvršnog koda jedino u slučaju da je dobavlјivost izvornog koda garantirana. Dozvoljava se redistribucija izvornog koda (i prisiljava se na to u slučaju redistribucije izvršnog koda). Dozvoljava se izmjena bez ograničenja (u slučaju da je derivirani rad također dio GPL-a) i potpuna integracija samo sa ostalim softwareom koji je također pod GPL-om. To nije slučaj sa LGPL⁷, pod kojom se software može integrirati s bilo kojim drugim softwareom, uključujući vlasnički.

⁶ GNU General Public Licence

⁷ GNU's Lesser General Public Licence

2.4. Prednosti i mane open sourcea

Open source ima svoje prednosti i mane, koje ćemo opisati u ovom djelu.

Prednosti open source programa su već jasne iz prethodnog dijela, ali ovdje ćemo sumirati taj dio. Prednosti open sourcea su:

- **Prilagođavanje** - Izvorni kod je na raspolaganju, te može biti mijenjan, proširivan i može prilagođavati posebnim potrebama.
- **Ponovno korištenje izvornog koda** - Ponovnim korištenjem dijelova s izvornog koda u drugim proizvodima se štedi na vremenu. Mogućnost proučavanja izvornog koda drugih programera vodi do prijenosa knowhova u zajednici programera.
- **Bolja kvaliteta softwarea** - Postupak izrađivanja open sourcea se razlikuje od jednog do drugog komercijalnog softwarea. Izrađivanje open sourcea ne stoji ni pod kakvim pritiskom od strane tržišnih zahtjeva. Iz tog razloga ne postoji potreba da se izrađuju “brzinska i nečista” rješenja, kao što je to slučaj sa mnogim komercijalnim proizvodima. Činjenica je da su open source kodovi (uglavnom) čisti, dobro strukturisani i dokumentirani. Ovo je neophodno jer i drugi programeri mogu poželjeti da čitaju, ponovo koriste ili mijenjaju izvorni kod. CEO jedna od vodećih tvrtki u GIS sektoru, koja je odlučila da dio svog izvornog koda stavi javnosti na raspolaganje, potvrdila je da su stručnjaci koji su radili na izradi softwarea morali u velikoj mjeri promijeniti navike programiranja kako bi se osigurali gore navedene karakteristike kvalitetnog izvornog koda.
- **Neovisnost** - Open source software ne primorava korisnika na bilo kakvu vrstu ovisnosti o određenom prodavaču kao što je to slučaj sa nekoliko komercijalnih proizvoda.
- **Bolja sigurnost** - Kako izvorni kod stoji na raspolaganju, sigurnosni nedostaci se mnogo brže otkrivaju i otklanjaju nego što je to slučaj sa komercijalnim proizvodima. Najbolji dokaz za ovo je Linux kao jedan od najsigurnijih i najstabilnijih operativnih sustava.
- **Otvoreni standardi** - Zbog slobodne raspoloživosti izvornog koda format datoteka i standardi koji se koriste za razmjenu podataka su po definiciji “otvoreni”. Ovo vodi ka boljoj kompatibilnosti i međuopreativnosti.
- **Bez plaćanja naknade za licencu** - Ovo mnogi smatraju najboljom prednošću open source softwera. Ipak, treba se spomenuti da plaćanje naknada za licence predstavlja samo dio od ukupnih troškova za vlasništvo nad softwareom. Međutim, neplaćanje naknade za licencu može biti bitan ekonomski faktor.

Da nebi ispalo da samo hvalimo open source navest ćemo i njegove nedostatke. Nedostatci, odnosno mane open sourca su:

- **Bez povrata na garanciju** - U principu korisnici open source software ne mogu zahtijevati nikakvu garanciju od autora (ili više njih). Uobičajene open source licence ne daju nikakvu garanciju za pouzdanost softwarea. Treba se spomenuti da postoji sve veći broj prodavača komercijalnih softwarea koji pokušavaju izbjeći pravo na povrat zbog garancije putem posebnim klauzula koje vežu na korištenje. Iz tog razloga, garancija igra tek manju ulogu.
- **Ne postoji podrška od strane programera** - Samo u rijetkim slučajevima na zahtjev se pruža podrška od strane autora (ili više njih). U praksi se softwareski problem uglavnom jako brzo rješava u većini slučajeva jer je neko u zajednici stručnjaka već možda naišao na problem i već ponudio rješenje. Dobar način za dobivanje podrške je potpisivanje ugovora sa lokalnom tvrtkom koja daje podršku za open source software. Davanje podrške i prilagođavanje open source softwarea je zanimljiv poslovni model i pojačano se koristi u IT tvrtkama.
- **Veća potreba za obukom** - U mnogim sektorima je korištenje komercijalnih softwarea (još uvijek) uobičajenije nego korištenje open source alternative (npr. office software ili operativni sistemi). Ovo dovodi do situacije gdje je lakše naći ljude sa poznavanjem komercijalnih proizvoda. Ako bi se open source software trebao početi koristiti kao alternativa nasuprot ovih proizvoda, pojaviti će se potreba za obukom ljudi ili za zapošljavanjem novog osoblja.
- **Bez garancije za kontinuitet/nastavak softwarea** - Autor (ili više njih) može uvijek završiti rad na open source projektima. To može učiniti jer ne postoji nikakva obaveza prema korisnicima softwarea. Ovo predstavlja manu za open source pri dugoročnom planiranju korištenja softwareskog proizvoda. Za dobro postavljene open source projekte rizik prestanka projekta je jako malen. Ipak, treba spomenuti da ni komercijalni software nema potpunu garanciju za neograničeno korištenje i postojanje. U slučaju open source softwarea još uvijek postoji mogućnost održavanja softwarea zbog izvornog koda koji stoji na raspolaganju.
- **Nedostatak interoperabilnost sa komercijalnim proizvodima** - U rijetkim slučajevima samo proizvođači komercijalnih softwarea imaju interes da naprave interoperabilnost sa open source softwareom. Iz tog razloga oni ne daju na uvid u njihove formate i interfejs

datoteka. Ovo ponovo može dovesti do posebnih situacija gdje se open source software ne može primjeniti.

3. PHP

„PHP⁸ je nastao iz PHP/FI kojeg je 1995. godine napravio Rasmus Lerdorf, kombinirajući Perl skripte na svojim osobnim web stranicama. Taj software je nazvao 'Personal Home Page Tools / Forms Interpreter'. S vremenom je na to dodavao neke funkcije iz programskog jezika C za komunikaciju s bazama podataka i publiciranje dinamičkih web stranica. Rasmus je javno objavio kod svog PHP/FI da bi ga svi mogli koristiti, ali i ako žele sudjelovati u budućem razvoju i poboljšanju.“ **[Bagarić, Bačić, Bičanić i Bernić; 2008; str. 4]**

PHP je program otvorenog koda, te koristi serversku tehnologiju, odnosno kažemo da je server-side skriptni jezik za dinamičko generiranje HTML⁹ koda. Drugim riječima, PHP je skriptni jezik pomoću kojeg možete kreirati HTML stranicu na serveru prije nego što se ona, popunjena dinamičkim sadržajem, pošalje klijentu. Ovim načinom generiranja sadržaja klijent ne može vidjeti kod (skriptu) koji je generirao sadržaj koji gleda, već ima pristup čistom HTML kodu.

PHP je interpreterski jezik koji se obrađuje na serveru i ne ovisi o pregledniku ni o operacijskom sustavu posjetitelja web sjedišta. U trenutku pokretanja php skripte na serveru se pokreće novi proces. PHP generira HTML koji se ispisuje u pregledniku. On je jedna od najnaprednijih i najkorištenijih server-side skriptnih tehnologija danas u upotrebi. Po svojoj sintaksi je poput mnogih drugih sličnih jezika, čak i koristi funkcije nekih drugih jezika kao što su C ili Perl. To znači da jednu radnju možete izvesti korištenjem više različitih funkcija. Tako ova dva primjera rezultiraju istim prikazom:

Primjer 1:

```
<?
    echo 'Završni rad';
?>
```

Primjer 2:

```
<?
    printf('Završni rad');
?>
```

⁸ Hypertext preprocessor

⁹ Hypertext MarkupLanguage

PHP datoteke, budući da koriste c i Perl mogu sadržavati tekst, HTML tagove i skripte, te mogu imati ekstenziju: ".php", ".php3", ".phtml", ... PHP datoteka se izvršava na poslužitelju, a zatim se vraća pregledniku kao običan HTML.

PHP se može izvoditi na različitim operacijskim sustavima, te komunicira sa ostalim servisima i protokolima. PHP je hibridni jezik, podržava proceduralno i objektno orijentirano programiranje, te gotovo sve tipove podataka (Boolean, Integer, Floating-point number (float), String, Array, Object, Resource, NULL...)

Korištenje PHP-a je u velikom porastu. Broj stranica koje ga koriste raste iz dana u dan, a broj tvrtki koje žele primijeniti PHP na svojim stranicama je još veći. PHP je izvrstan jer pomoću njega s nevjerojatnom lakoćom možete stvoriti opširnu web aplikaciju s velikim količinama podataka. Vrlo popularan u upotrebi, alternativa je svom glavnom konkurentu - Microsoft's ASP-u¹⁰. Često se koristi uz Apache web server, na različitim operacijskim sustavima. Podržava ISAPI protokol za razvoj aplikacijskih sučelja i može se koristiti i na Microsoft's Internet Information Server-u (IIS) pod Windows op. sustavom.

PHP može raditi sa velikim brojem baza podataka budući da podržava i ODBC¹¹ standard. Podržana je i komunikacija s ostalim servisima korištenjem protokola kao što su LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM(Windows) i mnogi drugi. PHP podržava WDDX razmjenu kompleksnih podataka sa svim ostalim programskim jezicima za Web; moguće je koristiti i Java objekte kao svoje te ostale remote objekte putem CORBA¹² ekstenzija koje omogućuju komunikaciju s distribuiranim aplikacijama neovisno o OS-u ili hardverskoj podršci. PHP podržava XML¹³.

Od velikog je značaja mogućnost pristupa bazama podataka, jer je to svojstvo svakog modernog programskog jezika. Značaj pristupa bazama podataka je leži u činjenici da sistem za upravljanje bazama podataka osigurava jako puno funkcija za rad s bazama podataka. Jezik PHP sadrži bogat skup funkcija za pristup različitim bazama podataka. Jedna od velikih prednosti PHP-a nad ostalim jezicima je ta što podržava gotovo sve najkorištenije baze podataka. Trenutačno je moguće koristiti baze podataka koje su prikazane unutar 3.1. tablice:

10 Active Server Pages

11 Open Database Connectivity

12 Common Object Request Broker Architecture

¹³ eXtensible Markup Language

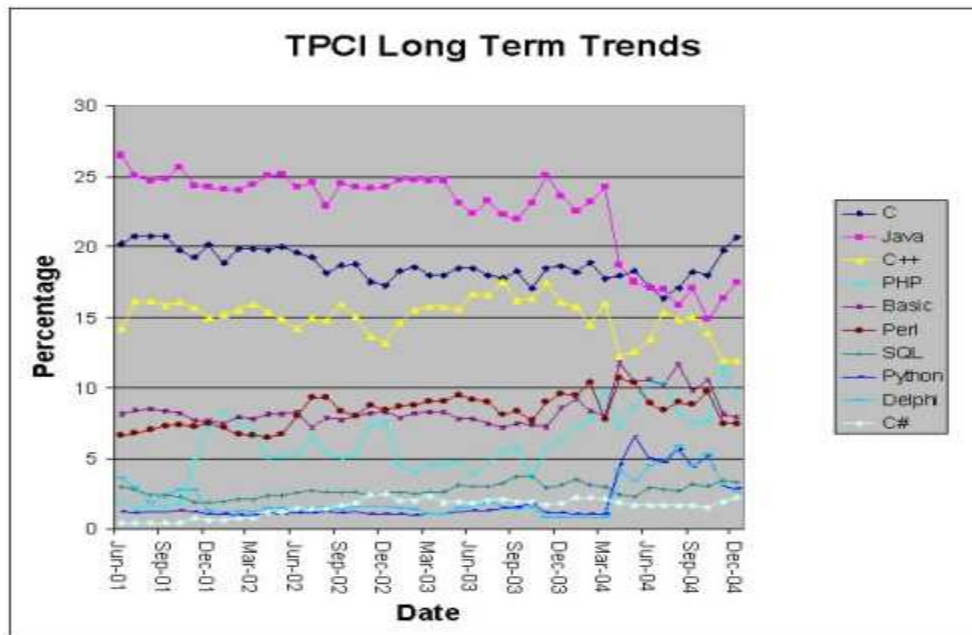
Tablica 3.1. Popis baza koje PHP može koristiti [Robeli, 2002, str. 19.]

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

“PHP je odnio pobjedu za 2004-u godinu kao najpopularniji programski jezik. Tvrtka Tiobe Software objavila je rezultate najpopularnijih tražilica na internetu, Google-a, MSN-a i Yahoo!-a po kojima je php uvjerljivo ispred svih ostalih.

TIOBE Programming Community pokazuje popularnost pojedinog programskog jezika. Nije riječ o izboru najboljeg programskog jezika ili jezika u kojem je najviše linija koda napisano već jednostavno prikaz popularnosti pojedinog.” [http://php.com.hr, 2005]

Rezultati s vidljivi na slici 2.1., gdje je vidljivo da PHP ima najveći porast u 2004. godini:

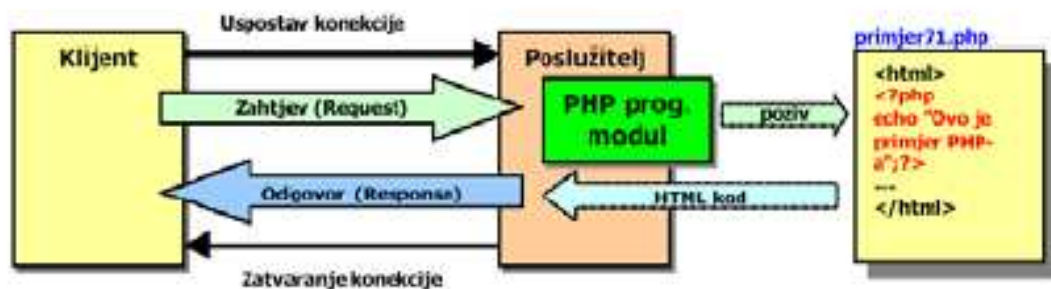


Slika 3.1. Rezultati istraživanja tvrtke Tiobe Software [http://php.com.hr/17, 2005]

Proces uključivanja PHP programa uključuje sljedeće korake ilustrirane na slici 3.2.:

1. Korisnik putem preglednika zahtijeva određeni resurs (PHP program, datoteku).

2. Zahtjev dolazi na poslužitelj koji prepoznaje da se radi o PHP programu.
3. Poslužitelj poziva PHP engine koji izvršava PHP program koji za rezultat ima HTML kod koji se onda nazad šalje korisniku.



Slika 3.2. Koraci uključivanja PHP programa [www.dead-pixel.info, 2005]

Osnovna razlika između PHP-a i samostalnih jezika je činjenica da je PHP prvobitno dizajniran isključivo za web programiranje, te radi isključivo u takvoj okolini – da bi ste uopće mogli pokrenuti svoje aplikacije, morate imati instaliran web server na svom računalu. Tek nedavno se PHP razvio dovoljno da se u njemu pišu samostalne aplikacije, a za to već postoje brojni drugi jezici. No, zato je PHP najbrži server-side skriptni jezik.

Kao i svaki drugi tako i ovaj programski jezik uz svoje vrline ima i neke mane. Jedan od nedostataka može biti to što u PHP-u deklaracija varijabli nije obavezna, što može biti izvor raznih grešaka i sigurnosnih propusta. Ugrađene funkcije nisu dosljedne po pitanju njihovog nazivanja ni po pitanju redoslijeda argumenata među sličnim funkcijama (primjer nazivanja: `strip_tags` i `html_entity_decode` nasuprot `stripslashes`, `htmlentities`). Isto tako funkcije nisu dosljedne u vraćanju rezultata - `false`, ali mogu vratiti i `0` ili `""`. Broj ugrađenih funkcija je velik (preko 3000) pa to otežava razvoj pogotovo jer dijele isto ime. Opcija `"register_globals"` automatski kreira varijable iz obrazaca pa to može postati sigurnosni rizik. Konfigurabilnost PHP-a koja je istovremeno i prednost i nedostatak jer jedna skripte na jednom serveru može raditi dok na drugom ne. Problem može biti i u tome što tabilnost PHP-a mnogo ovisi o vanjskim bibliotekama funkcija.

3.1. Web tehnologije

Budući da se PHP koristi isključivo za web programiranje, te koristi web tehnologije potrebno je objasniti i njih.

Dvije su vrste web tehnologija:

- Serverske tehnologije
- Klijentske tehnologije

Kako PHP spada u serversku tehnologiju, odnosno skriptni je jezik ovdje ćemo se najviše baviti s tim. Skriptni jezik se koristi kod obe prethodno spomenute tehnologije.

3.1.1. Serverske tehnologije

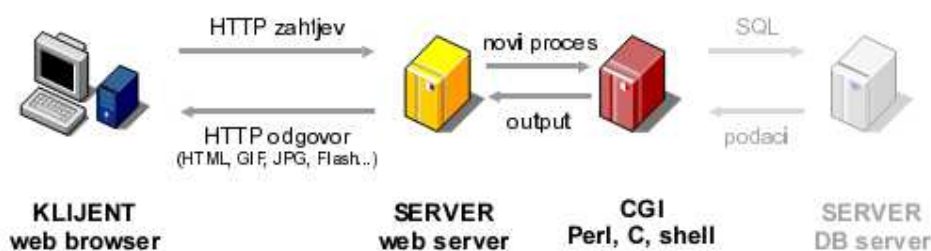
Postoji podjela i serverskih Web tehnologija, te se mogu ugrubo podijeliti na:

- samostalne programe
- serverske skripte
- specijalne jezike

Samostalne programe mogu predstavljati npr. CGI¹⁴ i Java servisi.

CGI se definira kao web server software, te predstavlja sučelje koje omogućava serverima da pokreću programe kao odgovor na HTTP zahtjeve. CGI specificira kako proslijediti argumente programu koji će se izvršiti kao dio HTTP zahtjeva. Također, CGI omogućava vraćanje rezultata u HTML formatu zavisno o parametrima konkretnog zahtjeva. Posao CGI skripte je, kao što je vidljivo na slici 3.3., da pročita informaciju koju je pretraživač poslao (preko poslužitelja) i generira neki oblik (obično vidljivog) odgovora. Kada završi zadatak, CGI skripta se gasi.

¹⁴ Common Gateway Interface



Slika 3.3. Princip rada CGI sučelja [http://www.irt.org, 2010]

Java servis služi za proširenje mogućnosti servera, a vlasnik Java servisa je Sun Microsystems kompanija. Gledajući Java servis u odnosu na CGI on radi nešto efikasnije, no zato i sporije (ujedno ima i problema s HTML ispisom).

Serverske skripte se interpretiraju, odnosno ne kompajliraju se, te se baziraju na miješanju koda aplikacije s HTML-om. Konačni rezultat je mješavina dinamički generiranog i statičnog HTML koda. Najčešće serverske skripte su ASP¹⁵, JSP¹⁶ te PHP. ASP je Microsoftov jezik razvijen 1996. godine, trenutno je aktivan ASP.NET. JSP je još jedna tehnologija Sun Microsystemsa koja miješa HTML kod s Javom.

Specijalnih jezika ima mnogo, ali nisu baš poznati, tu je najpoznatiji primjer ColdFusion (u vlasništvu Macromedie) – tehnologija osmišljena za mrežno prezentiranje informacija proizašlih iz neke od relacijskih baza podataka. Sintaksa je bazirana na ColdFusion Markup Languageu, koji se umeće unutar običnog HTML-a. Sintaksa pisanja CFML je praktički identična HTML-u.

3.1.2. Klijentske tehnologije

Klijentske tehnologije, kao što im i samo ime kaže, omogućuju pristup klijentima. Za razliku od serverskih Web tehnologija, mnogo je veći naglasak dan Web standardima, ponajprije zbog toga što je prilikom odabira serverske tehnologije sloboda skoro potpuna – sve dok je na izlazu HTML, a kao ulaz se koristi standardno sučelje prema bazi podataka. S druge strane, sve što znamo o klijentu je da koristi preglednik, koji prikazuje sadržaj ovisno o standardima. Za standardizaciju Web formata zadužen je W3C.¹⁷

Klijentske tehnologije mogu se ugrubo podijeliti na:

¹⁵ Active Server Pages

¹⁶ Java Server Pages

¹⁷ World Wide Web Consortium

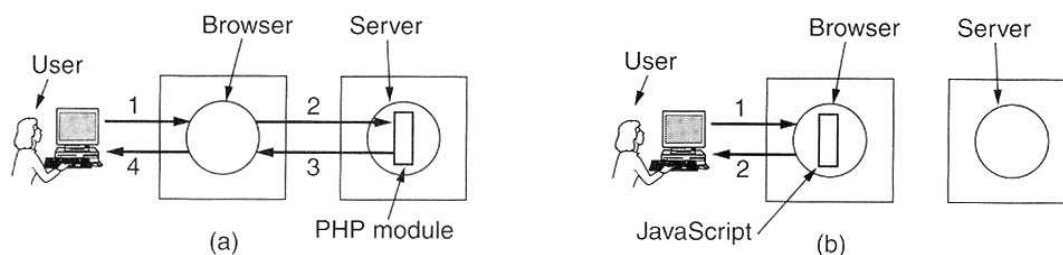
- opise sadržaja i prezentacije (HTML, XHTML, XML, CSS...)
- skriptne jezike (JavaScript, Jscript, VBScript...)
- DOM (Document Object Model)
- dodatne objekte (npr. slikovni objekti – GIF, JPG, PNG datoteke)

3.1.3. Skriptni jezik

Serverske skripte se interpretiraju, odnosno ne kompajliraju se, te se baziraju na miješanju koda aplikacije s HTML-om. Konvencionalni programi se trajno pretvaraju u izvršne datoteke prije nego što se pokrenu, dok se programi pisani u nekom skriptnom jeziku interpretiraju naredbu po naredbu tijekom izvođenja. Doduše, većina interpretiranih jezika ipak prevodi izvorni kod u „međukod“ kako bi povećali brzinu izvođenja.

Skriptne jezike se relativno često, neopravdano, svrstava u inferijornije programske jezike. Skriptni jezici često daju prednost lakoći razvoja i lakoći distribucije prije efikasnosti izvođenja. S tog gledišta primamljivo je misliti da su skriptni jezici inferiorniji u usporedbi sa standardnim programskim jezicima. No, rađe ih razmotrite sa stajališta da nude različiti asortiman prednosti i nedostataka naspram konvencionalnih programskih jezika. To je nešto što postane očito pri proučavanju praktičnih koristi skriptiranja. U većini slučajeva, lakše je pisati kod u skriptnom jeziku nego u kompajliranom, no sporiji su zbog toga što instrukcije ne obrađuje samo osnovni instrukcijski procesor. Skriptni jezici omogućuju laku komunikaciju s programima pisanim u drugim jezicima, brže pisanje programa jer su sažetiji te simuliraju dinamičku, interpretiranu radnu okolinu, gdje se dijelovi programa mogu mijenjati tokom izvedbe. Razvoj Interneta je također popularizirao skriptne jezike. Internet je ništa više nego gomila slijepljenih komponenata. Ne stvara nikakve nove podatke; samo čini veliki broj postojećih stvari lako dostupnima. Idealni jezik za većinu Internetskog programiranja bio bi onaj koji omogućava da sve povezane komponente rade zajedno, tj. skriptni jezik.

Kao što je već spomenuto u predhodnom dijelu, skriptiranje imamo i u client-side i u server-side tehnologiju. Glavna razlika je da se prvo izvodi na korisnikovom računalu, točnije u web browseru, dok se drugo izvodi na web poslužitelju (jednom od modula). Na slici 3.4. je ilustrativno prikazana ta razlika.



Slika 3.4. Dva načina izvođenja skripti: a) server-side (PHP); b) client-side (JavaScript)
[www.syntaxpolice.org, 2010]

Client-side skriptiranje napravljeno je za uklapanje u HTML dokument. Skripta predstavlja integralni dio HTML dokumenta kojeg server dostavlja klijentu. Interpretira se i pokreće unutar preglednika kojeg klijent koristi, tj. na korisničkom računalu. Najčešće korišteni client-side skriptni jezici danas su JavaScript i njegove varijante koje su izvedene iz Jave.

- JavaScript
- JScript
- ECMAScript (European Computer Manufacturer's Association)
- VBScript izveden is Visual Basic-a.

Server-side skriptiranje napravljeno je da bi bilo postavljeno, interpretirano i pokrenuto na mrežnom serveru koji predaje mrežnu stranicu. Pozivi na server-side skriptu dolaze iz preglednika no kod se interpretira i izvršava na serveru. Rezultati se predaju klijentu preko preglednika u obliku mrežnih stranica ili dodataka stranicama. Postoje mnogi server-side skriptni i programski jezici i njihova okruženja. Među najpopularnijim i najkorištenijim danas su:

- ASP
- CGI
- ColdFusion
- Java
- Perl
- PHP

- Python
- XML
- XSL¹⁸

3.2. Apache mrežni HTTP server

Za izvođenje PHP najčešće se koristi HTTP Server Apache, ili jednostavno Apache, koji predstavlja poslužitelja mrežnih stranica. Značajan je zbog igranja ključne uloge u prvobitnom rastu World Wide Web-a, te je u 2009. godini postao prvi mrežni server koji je prešao prag od 100 milijuna posluženih mrežnih stranica. Većina mrežnih servera (računala) koji koriste Apache rade pod operacijskim sustavom Linux. Još od travnja 1996. godine, Apache je najpopularniji HTTP server na internetu. Od ožujka 2009. poslužuje preko 46% svih mrežnih stranica i preko 66% od milijun najprometnijih. Apache se primarno koristi za posluživanje statičkih i dinamičkih mrežnih stranica na internetu. Mnoge mrežne aplikacije su napravljene s pretpostavkom da će se pokretati u okruženju koje Apache pruža. Apache se koristi i u brojnim drugim situacijama kada sadržaj treba biti dostupan na siguran i pouzdan način. Jedan primjer je razmjena podataka s osobnog računala preko interneta. Korisnik koji ima Apache na svom računalu može staviti proizvoljne dokumente u za to namijenjen direktorij te ih tako dijeliti, što je korisno ako su datoteke veće od 10 MB. Programeri koji razvijaju mrežne aplikacije često koriste lokalnu instalaciju Apache-a kako bi provjerili svoj kod tokom razvoja. Besplatan je i pod licencom otvorenog koda. Dostupan je na mnoštvu operacijskih sustava, uključujući Unix, Linux, Windows, Mac OS X, Solaris, FreeBSD te Novell Netware.

¹⁸ eXtensible Style Language

4. MySQL

MySQL je odličan SUBP¹⁹ koji je mogo dostupniji od od najvećeg broja konkurentnih proizvoda, ima široku primjenu, ujedno je i baza podataka koja se najviše koristi pri radu u PHP-u, te ima više od 11 milijuna instalacija. MySQL je bio vlasništvo švedske tvrtke MySQL AB koja je danas dio Sun Microsystems-a, koji drži copyright prava na većinu kodova.

Zanimljivo je da je MySQL prvobitno nastao kao interni projekt. Prva verzija ovoga software-a izdana je 23.05.1995., najnovija verzija 5.0.51a(Community server) izdana je 06.12.2007. Treba reći da je source kod ove aplikacije napisan u C i C++ programskom jeziku. Iz tih razloga ova aplikacija radi na mnogo različitih OS platforma.

MySQL je prije svega sustav za upravljanje relacijskom bazom podataka. Radi se o programu kojem se može pristupiti preko mreže na sličan način kao i web (HTTP) poslužiteljima, sa tom razlikom da se MySQL-u obično pristupa pomoću korisničkog imena i lozinke. Naime, MySQL popularni relacijski sustav za upravljanje bazom podataka za mrežne aplikacije dio je LAMP (i drugih) distribucije serverskih aplikacija. Njezina popularnost je usko povezana s popularnošću PHP-a, jer se često koriste zajedno. Nekoliko najprometnijih mrežnih stranica (uključujući Flickr, Facebook, Wikipediju, Google (no ne za pretraživanje), i YouTube) koriste MySQL za pohranu podataka i praćenje korisničkih podataka. MySQL se izdaje pod open-source licencom pa je stoga besplatan.

Na serveru može postojati veći broj, potupuno samostalnih, baza podataka. Jedna baza obično se sastoji od više tablica. Dok se unutar jednog projekta može baratati s podacima iz više baza na serveru. Naime, svakom korisničkom računu na serveru mogu se dodjeliti različita administracijska prava i to na čitav server ili pojedine baze. Pri instalaciji MySQL-a se stvara tzv. superadministrator (obično se zove root (korjen)) koji ima sva administracijska prava. Postoje još i pravo stvaranja novih baza, pravo uređivanja postojećih baza itd.

Postoje različite jako fleksibilne licencne opcije MySQL-a. Jedna od velikih prednosti MySQL-a je što postoje verzije za sve važnije operacijske sustave, te ih se distribuira pod GPL licencom što ga čini idealim alatom za učenje osnova relacijskih baza podataka.

MySQL je najviše korišten SUBP iz skupine programa otvorenog programskog koda. Njegova popularnost temelji se na mogućnosti besplatnog korištenja, podržanosti velikog broja platformi, relativnoj jednostavnosti, lakom održavanju i zadovoljavajućim performansama. MySQL može se koristiti u manjem obimu samo za podršku ugrađenih aplikacija baza podataka.

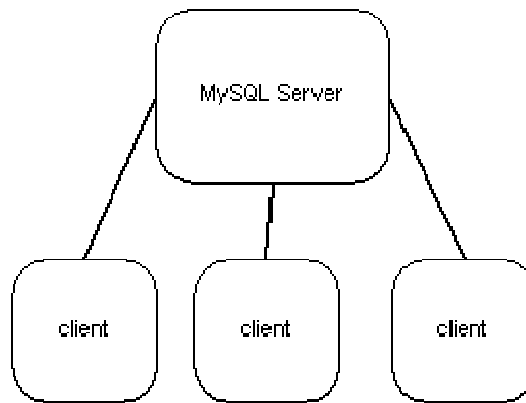
¹⁹ Sustav za upravljanje bazama podataka

Zbog te mogućnosti, mnogi pogrešno smatraju da je to baza manjih mogućnosti, za manju količinu podataka i manji broj korisnika, iako to nije slučaj.

Budući da se ovaj program koristi u skladu sa mnogim programskim jezicima paketi funkcija za pristup i korištenje MySQL-a dostupni su u svim većim programskim jezicima u okviru pripadnih API datoteka. Kao dodatak ODBC sučelje koje ima naziv MyODBC dozvoljava dodatnim programskim jezicima koji podržavaju ODBC sučelje da komuniciraju sa MySQL bazom podataka (kao na primjer ASP ili ColdFusion). MySQL server i službeni skupovi metoda (libraries) su uglavnom implementirani u ANSI C i ANSI C++ kodu.

MySQL je po svojoj primjeni isti kao i drugi SUBP-i za relacijske baze podataka, pa mu je glavna primjena MySQL-a je ista kao i primjena SQL-a s tom razlikom što je MySQL još više neovisan o platformi sustava u koji je instaliran i ima mnogo veću brzinu obrade podataka što ga čini jednim od boljih, ako ne i najboljim sustavom za kontrolu i korištenje baza podataka na web-u, gdje se očekuje brza i sigurna obrada podataka. Obično se MySQL bazom koristi preko PHP skripti koje se nalaze unutar web stranica, umetnute između HTML koda. Kako se koristi PHP koji se izvršava na serveru, a rezultati obrade se zatim vrlo velikom brzinom šalju korisniku, na sličan način radi i MySQL. Postoji MySQL server gdje se na nekome disku fizički nalaze baze podataka i korisnički računi osoba koje imaju određene stupnjeve ovlasti za izvršavanje određenih operacija na samoj bazi i podacima unutar nje. Sam prihvrat zahtjeva s korisničkih- klijentskih računala prima se na serveru i server provjerava da li taj određeni korisnik ima autorizaciju za tražene operacije, ako ih korisnik ima onda server izvršava zadane operacije na bazi i podacima i rezultat obrade šalje natrag klijentskom računalu. Time je napravljena jedna operacija na MySQL bazi podataka. Još jedna bitna karakteristika MySQL sustava je u tome što MySQL server može prihvatiti više zahtjeva za obradu od više korisnika u isto vrijeme i izvršiti ih u isto vrijeme potpuno paralelno, bez zastoja. Iz gore navedenih razloga i neovisnosti o platformi sustava u kojemu se koristi MySQL je vrlo popularan kod web aplikacija i radi kao komponenta baze podataka na platformama kao što su Linux, BSD, Mac, Windows-Apache-MySQL-PHP, Perl, Python i drugim.

Kao i kod svakog drugog mrežnog računalnog sustava MySQL se sastoji od dva glavna dijela, od MySQL servera i niza klijenata. Na slici 4.1. prikazana je osnovna struktura MySQL -a sustava. Koristi se jednostavni klijent-server model. Na MySQL serveru može postojati jedna ili više shema na koje se spaja jedan ili više klijenata. Klijent može biti MySQL klijent ili neka druga aplikacija koja koristi bazu podataka.



Slika 4.1. Struktura MySQL-a

MySQL, kao i sve drugo čemu se pristupa preko mreže, može biti i nesigurna baza podataka. Ako mu pristupa računalo koje je u internoj mreži automatski se postavljaju administratorska dopuštenja i zbog toga interni napadi mogu biti puno opasniji od udaljenih. MySQL sadrži brojne skripte koje u radu koriste privremene datoteke. U nekim slučajevima te se privremene datoteke stvaraju na nesigurnim mjestima i s predvidljivim imenima pa mogu biti zamijenjene simboličkim vezama prema kritičnim sistemskim datotekama. MySQL skripta prilikom prepisivanja sistemske datoteke koristi ovlasti MySQL procesa koji ju je pokrenuo. Za razliku od drugih SUBP-ova, kao što su npr. Oracle ili Sybase, MySQL izvorno sadrži prilično slabu mrežnu podršku. Zbog toga napadač, nakon proboja u MySQL poslužitelj, nema puno mogućnosti za proširivanje napada na ostatak računalne mreže.

5. SourceForge.net

SourceForge.net se predstavlja kao najveći svjetski ponuđač open source rješenja. SourceForge.net razvijaju programeri iz cijeloga svijeta tako što postavljaju svoje radove na stranicu SourceForge.net, te time ona postaje najveća zbirka open source alata i aplikacija na internetu.

SourceForge.net je stvorila tvrtka Geeknet kako bi povezala milijune programa iz cijeloga svijeta, kombinirajući korisnički razvijene sadržaje. Geeknet nije razvio samo SF.net, već i druge stranice za razmjenu informacija vezanih za open source kao što su Slashdot, ThinkGeek i freshmeat.net.

SourceForge.net je jedna jako zanimljiva stranica sa mnoštvom programa i detalja. Neki od tih detalja su i programi za kontrolu verzija, gdje je osnovni program SVN koji ćemo u daljnjem tekstu opisati, zatim SF.net predlaže i vođenje dokumentacije u obliku wikija ili bloga, pa ćemo opisati wiki sustav.

5.1. SVN

SVN²⁰ je besplatan program otvorenog koda za kontrolu verzije i razmjenu podataka. SVN omogućava praćenje i održavanje verzija dokumenata, u našem slučaju koda. Koristeći SVN, projekt se nalazi na jednom mjestu, u repozitoriju, čiju kopiju svaki od članova može „pokupiti“ (checkout) na svoje računalo. Nakon toga sve promjene koje se urade lokalno, mogu se poslati u repozitorij (commit), te se može ažurirati (update) lokalna verzija sa svim promjenama koje su u međuvremenu urađene (i poslane u repozitorij) od strane drugih osoba. SVN se brine za to da se šalju samo one promjene koje su načinjene, što znači da ako promijenite jednu liniju koda u nekom dokumentu, samo ta linija će biti poslana (isto vrijedi i za update). Kontrola verzija, znači da je moguće vratiti dokument na bilo koju verziju koja se nalazila u repozitoriju, što može biti korisno ako neke promjene uvedu bagove.

Subversion upravlja datotekama i direktorijima. Stablo datoteka stavlja se u repozitorij. Repozitorij je sličan datotečnom poslužitelju, s tom razlikom da pamti svaku promjenu ikad

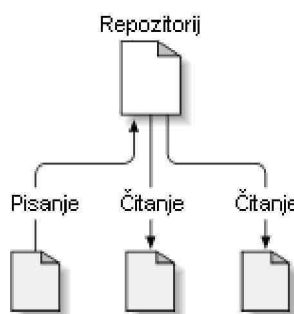
²⁰ Subversion

napravljen na svim datotekama i direktorijima. Za svaki direktorij i datoteku se vodi evidencija o njihovoj verziji, pa ako dođe do neke pogreške, jednostavno se ponište promjene koje su izazvale pogrešku.

Subversion je nastao kao zamjena za CVS, a razvila ga je grupa CollabNet. CVS²¹ je do tada bio standardan alat zbog toga što nije bilo boljeg alata, barem ne pod slobodnom licencom.

Subversion je centralizirani sustav za razmjenu informacija. Njegova jezgra je repozitorij koji sadrži sve podatke. Repozitorij sprema informacije u obliku datotečnog stabla tj. tipične hijerarhije datoteka i direktorija.

Klijenti se spajaju na repozitorij i onda pišu ili čitaju podatke sa njega. Na slid 2.1 prikazan je tipičan klijent/poslužitelj system.



Slika 5.1. Tipičan izgled klijent/poslužitelj sistema [Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, 2004.]

Subversion repozitorij je sličan datotečnom poslužitelju, s tom razlikom da pamti svaku promjenu ikad napravljenu na svim datotekama i direktorijima. Kada klijent pročita podatke sa repozitorija on vidi samo najnovije verzije datotečnog stabla, ali ima i mogućnost uvida u ranije verzije datotečnog sistema

Kako pamti svaku promjenu za svaku datoteku u radnom direktoriju Subversion sprema dvije bitne informacije u `.svn` direktorij:

1. Revizija na kojoj se radna datoteka zasniva, tzv. radna revizija
2. Vrijeme zadnjeg ažuriranja lokalne kopije

“Subversion razlikuje četiri stanja radne datoteke:

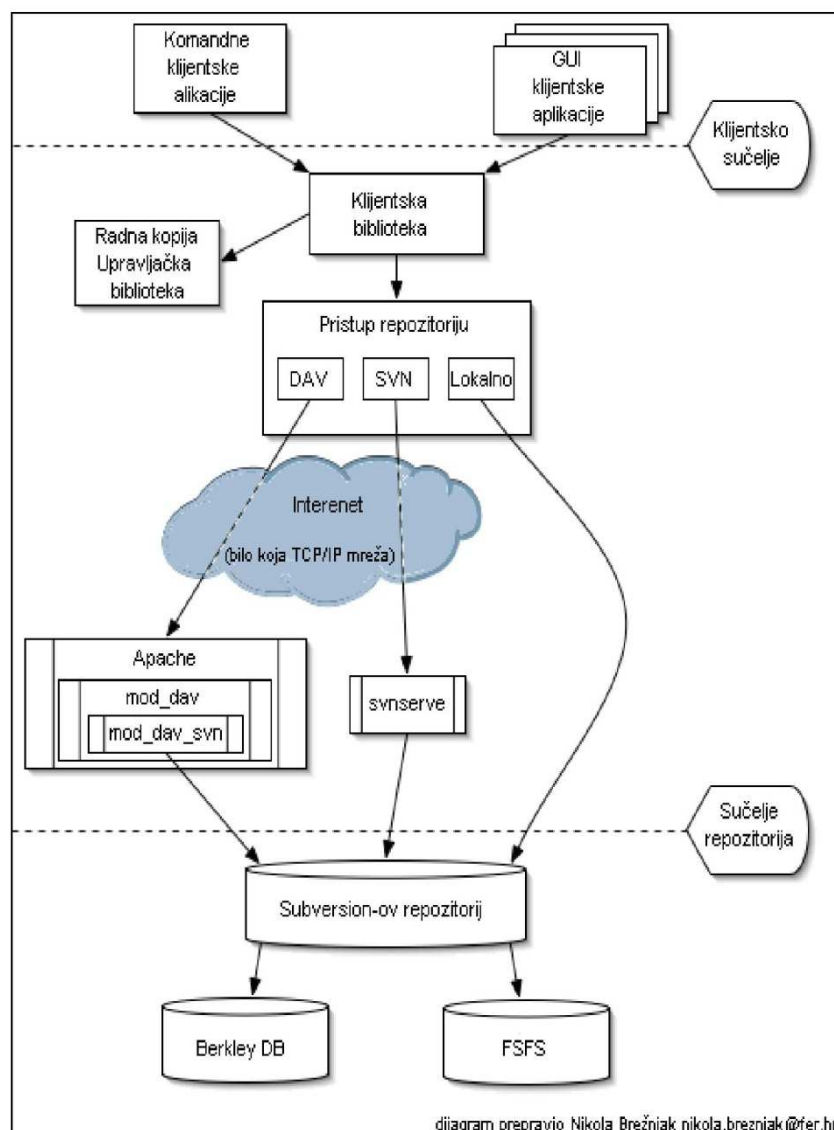
1. **Napromijenjena i tekuća** - datoteka je nepromijenjena u radnom direktoriju i nema promjena u repozitoriju. Naredbe `svn commit` i `svn update` neće napraviti ništa.
2. **Lokalno promijenjena i tekuća** - datoteka je promijenjena u radnom direktoriju i nije

²¹ Concurrent Versions Systems

mijenjana u repozitoriju. Naredba svn commit će uspješno objaviti promjene u repozitoriju, dok naredba svn update neće napraviti ništa.

3. **Nepromijenjena i zastarjela** - datoteka nije mijenjana u radnom direktoriju, ali je mijenjana u repozitoriju. Naredba svn update će ažurirati datoteku u radnom direktoriju, dok naredba svn commit neće napraviti ništa.
4. **Lokalno promijenjena i zastarjela** - datoteka je promijenjena u radnom direktoriju i repozitoriju. Naredba svn commit će dojaviti grešku o neažuriranosti datoteke, dok će naredba svn update pokušati spojiti lokalne promjene sa onima iz repozitorija. Ukoliko Subversion ne uspije automatski spojiti promjene, ostavlja korisniku da riješi konflikt.”

[Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, 2004.]



Slika 5.2. Arhitektura SVN-a [Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, 2004.]

U SVN arhitekturi na jednoj strani je Subversionov repozitorij u koji su smještene sve verificirane datoteke. Na drugom kraju se nalazi Subversionov klijentski program koji lokalno upravlja tzv. radnim kopijama. Između ova dva kraja nalazi se dio koji upravlja pristupom Subversion-ovom repozitoriju, koji može biti direktan ili preko mreže.

Tablica 5.1. Komponente SVN-a

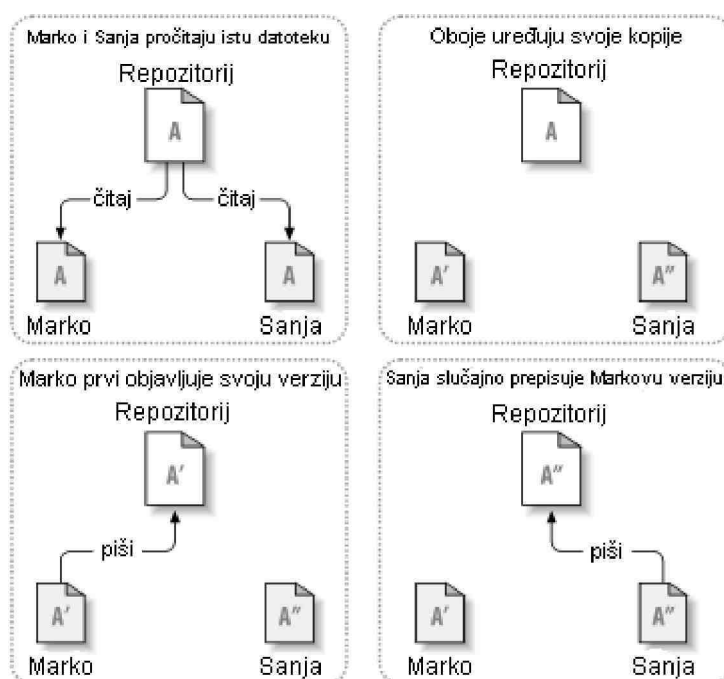
Naziv komponente	Opis rada komponente
Svn	komandni klijentski program
Svnversion	program za ispis stanja radne kopije
Svnlook	alat za pregledavanje repozitorija
Svnadmin	alat za stvaranje, podešavanje ili popravljavanje
Svnumfilter	program za filtriranje repozitorija
Mod_dav_svn	modul za Apache HTTP server, koji omogućuje dostupnost repozitorija preko mreže
svnserve	samostalan server, za pristup repozitoriju preko mreže

5.1.1. Modeli kontrole verzija

Postoji više različitih programa kojima se kontrolira verzija, ali bilo koji od njih mora korisnicima dati mogućnost da dijele informacije, ali isto tako moraju spriječiti slučajne izmjene tuđih informacija.

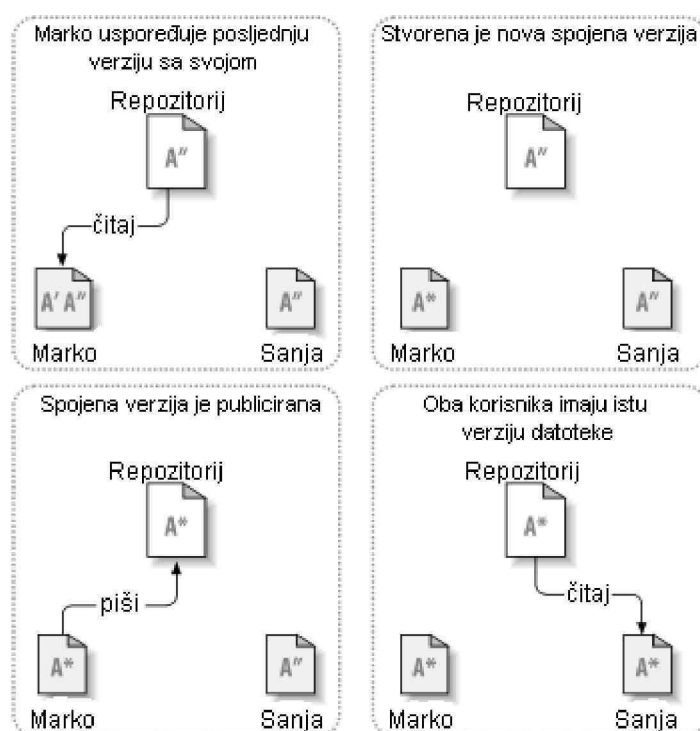
Postoje dva modela kojima se omogućavaju te situacije:

Model zaključaj-modificiraj-otključaj - U ovom modelu repozitorij dopušta samo jednom korisniku mijenjanje neke datoteke. Ovaj model koristi modele za zaključavanje. Pretpostavimo da imamo dva korisnika, jedan je Marko, a drugi Sanja. Marko mora “zaključati” datoteku prije nego što je može mijenjati. Sanja ne može zaključati datoteku ako ju je Marko zaključao prije nje. Sanja može samo čitati datoteku i čekati da Marko završi sa uređivanjem datoteke. Tek onda kada Marko otključa datoteku, Sanja može zaključati tu istu datoteku i obrađivati je. Ova situacija je prikazana na slici 5.3.



Slika 5.3. Model zaključaj-modificiraj-otključaj[Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, 2004.]

Model kopiraj-modificiraj-spoji - U ovom modelu svaki korisnik na svoje lokalno računalo postavlja svoju kopiju datoteka i direktorija iz repozitorija, ta kopija se naziva radna kopija. Korisnici mogu raditi istovremeno na svojim vlastitim kopijama. Konačno, privatne kopije se spajaju (*eng. merge*) u novu konačnu verziju (*eng. final version*). Sistemi za kontrolu verzije pomažu kod spajanja, ali je korisnik odgovoran da se to ispravno obavi. Recimo da Marko i Sanja naprave svaki svoju radnu kopiju istog projekta, iskopiranog iz repozitorija. Marko i Sanja naprave svoje izmjene u istoj datoteci naziva A, unutar svojih radnih kopija. Sanja prva snimi svoje promjene u repozitorij. Kasnije kada Marko pokuša snimiti svoje promjene, repozitorij ga obavještava da je verzija njegove datoteke zastarjela (*eng. out of date*), kao što je prikazano na slici 3. Drugim riječima, repozitorij ga obavještava da se je datoteka A nekako promijenila od vremena kada ju je on iskopirao iz repozitorija. Nakon toga Marko traži od svoj klijenta da spoji promjene datoteke A iz repozitorija sa njegovom radnom kopijom datoteke A. U slučaju da se Sanjine promjene ne preklapaju sa Markovim promjenama, Subversion spaja te dvije datoteke u jednu, te Marko tada može tu spojenu datoteku spremiti natrag u repozitorij. Na slid 5.4. je prikazan taj slučaj.



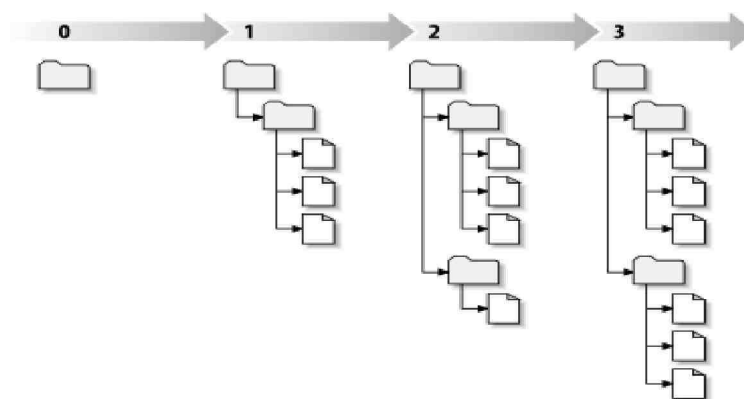
Slika 5.4. Model rješenja kopiraj-modificiraj-spoji[Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, 2004.]

U slučaju da se Sanjine promjene identične sa Markovim onda se to naziva konflikt i često ne predstavlja velik problem. Markova datoteka je označena da je u stanju konflikta i on može vidjeti promjene koje su uzrokovale konflikt. Subversion automatski otkriva stanje konflikta, ali ne može odabrati između tih konflikata. Konflikt mora riješiti korisnik. Tako u ovom primjeru Marko mora riješiti problem zbog kojeg je nastao konflikt (npr. nakon što se je savjetovao sa Sanjom), i tek nakon toga može spremiti datoteku nazad u repozitorij.

5.1.2. Revizije

Svaki puta kada repozitorij prihvati unos stvara se novo stanje sistemskog datotečnog stabla koje se naziva revizija. Svakoj reviziji je pridružen jedinstven prirodan broj, za jedan veći od prethodne revizije. Inicijalnoj reviziji novostvorenog repozitorija pridružena je vrijednost 0 i sadrži samo prazan korijenski direktorij (*eng. root directory*). Slika 5.5. prikazuje način na koji se može vizualizirati repozitorij.

Za razliku od puno ostalih sustava za kontrolu verzije, Subversion primjenjuje revizijske brojeve na cijelo stablo, a ne samo na pojedinačne datoteke. Drugim riječima, N-ta revizija predstavlja stanje datotečnog sistema repozitorija nakon N-tog uspješnog pisanja u repozitorij.



Slika 5.5. Način vizualizacije repozitorija[Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, 2004.]

Subversion omogućuje da radna kopija sadrži datoteke i direktorije koji imaju različite brojeve radnih revizija. Naredbe za postavljanje u repozitoriju (`svn commit`) i ažuriranje radne kopije (`svn update`) ne povlače jedna drugu, tj. rade odvojeno. To znači da je moguće imati datoteku čija je radna revizija npr. N ($N > M$), dok je revizija radnog direktorija, u kojem se ta datoteka nalazi,

M. Samo pokretanjem naredbe za ažuriranje radne kopije će se cijela radna kopija dovesti u stanje M-te revizije.

5.2. Wiki

SF.net posjeduje jako dobru podršku za Wiki. “Wiki” je skraćenica za “Wiki Wiki web” i potiče od havajskog izraza “wiki wiki” što znači brzo ili hitro. Wiki u osnovi predstavlja kolekciju web stranica koji su povezani preko hiperveza. Iako postoji širok spektar Wiki sustava (kao što su MediaWiki, MoinWiki, TWiki) sa različitim svrhama i korisnicima, svi oni dijele slijedeće zajedničke osobine:

- **Uređivanje preko pretraživača-čitača** - Sadržaj se obično mijenja preko jednostavnog sučelja pretraživača koji se može koristiti bez potrebe za instaliranjem nekog dodatnog (skupog) softwarea. To čini izmjenu/uređivanje lakim i jednostavnim i dopušta modificiranje strana sa bilo kojeg mjesta na svijetu uz minimum tehničkih zahtijeva. Dalje to znači da stvaratelji sadržaja mogu pristupiti i dopuniti Wiki sa bilo kojeg mjesta gdje se trenutno nalaze (sa posla, od kuće, sa konferencija, čak i dok putuju).
- **Jednostavnost Wiki sintakse** - Sadržaj je obično izražen u jednostavnom hipertekst formatu Wiki sintakse koji je nestručnim korisnicima (u tehničkom smislu) mnogo lakši za korištenje od na primjer HTML-a. Prema tome, oblikovanje/prezentacija ne zahtjeva poznavanje HTML-a.
- **Mehanizam povratka u prijašnje stanje** - Promjene sadržaja na Wikiju se provode svaki put kada se spremne, a prethodne verzije se čuvaju. Ovo dopušta povratak na ranije verzije. Također, većina Wiki sistema omogućava uspoređivanje dvije verzije, kako bi se što brže mogle utvrditi i izvršiti promjene.
- **Jako povezivanje** - Stranice u Wikiju su obično međusobno povezane jakim vezama preko hiperveza. To je posljedica lakog kreiranja veze prema drugoj strani pomoću jednostavne Wiki sintakse. Na primjer, u mnogim Wikijima veza je definirana preko riječi koja se nalazi između srednjih ([,]) zagrada ili koristeći takozvani „CamelCase“ gdje riječ koja predstavlja vezu se sastoji od nekoliko velikih slova. Veze prema nepostojećim stranama se obično prikazuju u drukčijoj boji. Ako korisnik klikne na takvu vezu sistem mu odgovara formom gdje može kreirati nepostojeću stranu. U nekim Wikijima ovo je čak jedini način da se kreira nova strana. Veze su u Wikiju najvažniji alati za navigaciju. Zbog toga mnogi sustavi dopuštaju ne samo praćenje veza u smjeru kako su definirane već i u njihovom suprotnom smjeru.

- **Neograničeno korištenje** - U većini Wiki sistema, korištenje je potpuno neograničeno, tj. svako može ispraviti, izmjeniti, dodati ili čak izbrisati bilo što. Dok se ovo može učiniti neobičnim, čak i opasnim sa tradicionalnog stanovišta, praksa pokazuje da ovakav sustav ispravno funkcionira. Sa jedne strane, korisnici sa lošim namjerama su jako retki, a sa druge strane sve promjene se lako mogu poništiti pomoću rollback mehanizma. Primjetimo da neki Wikiji još dopuštaju upotrebu kontrole pristupa ograničavanjem njihovog korištenja od strane korisnika i grupa kao što je to u slučaju tradicionalnog CMS-a.
- **Udružene izmjene** - Gore spomenute osobine čine Wikije idealnim alatom za udružene izmjene. Dok netko stvara sadržaj, drugi ga može iskoristiti kao prilog u svom sadržaju, proširiti ga, ispraviti greške itd. Većina Wiki sistema predviđa i dalje usavršavanje ovakve udružene izmjene, na primjer listama posljednjih izmena, sumiranjem promjena, diskusijama na forumu.

Wiki daje korisnicima potpunu slobodu razvoja sadržaja bez strogih pravila, zabrane korištenja ili predefiniranih struktura.

Kao najbolji alat za wiki se nameće MediaWiki. To je software koji pokreće Wikipediju i njegovo jednostavno sučelje i funkcionalnost su ključni za razvoj i popularnost Wikipedije. Svi članci imaju posebnu stranicu za razgovor o sadržaju članka te stranicu na kojoj su dostupne sve prethodne verzije članaka. To znači da svaka verzija članka u prošlosti je sačuvana i na nju se po potrebi može napraviti link koji će biti trajan.

„Uz standardne mogućnosti oblikovanja teksta neke od najkorisnijih komponenti ugrađenih u članke i wiki sintaksu su: wikilinks, references (izvori), external links (vanjske poveznice), categories, lists, templates, interlanguage links.

Wikilinks su linkovi koji povezuju određeni pojam u tekstu sa člankom na Wikipediji. References su linkovi na druge web stranice koji se dodaju kao potvrda ispravnosti sadržaja, ali i kao dodatna literatura. References se dodaju iza rečenice na koju se odnose i to kao direktni link (strelica) ili interni link na dno stranice (brojevi) gdje su onda izlistani svi izvori. References mogu biti i razni pisani izvori te video i audio sadržaji koji nisu dostupni online. External links se dodaju na kraju članka i to su obično službene web stranice organizacija koje imaju veze sa temom članka. Interlanguage links su veze na članke dostupne na drugim jezicima. Prikazuju se na lijevom dijelu stranice kao lista svih jezika na kojem je članak dostupan.

Categories, lists i templates služe za bolju navigaciju među člancima na Wikipediji. Kategorije su pojmovi koji su zajednički većem broju članaka. Svaki članak ima na dnu napisane sve kategorije kojima pripada. Svaka kategorija ima svoju stranicu na kojoj su izlistana imena svih članaka koji spadaju u tu kategoriju. Stranice kategorija se pune automatski kako netko doda ime kategorije u članak. Imena kategorija zadaje bilo koji urednik Wikipedije te ih po potrebi dijeli u podkategorije. Kategorija danas ima nekoliko desetaka tisuća pa samo iskusni korisnici Wikipedije znaju koje sve kategorije bi trebalo uključiti u članak. To znači da rijetko koji članak ima na dnu ispisane baš sve kategorije kojima pripada. Liste su slične kategorijama ali su pojmovi u listama ispisani logički i hijerarhijski, a ne abecedno. To ujedno znači da se njihov sadržaj ne puni automatski već o listama brinu urednici. Liste su za razliku od templates-a i kategorija obični članci te se na Wikipedijinoj tražilici izlistavaju među ostalim člancima. Template je tablica koja se stavlja u više članaka i ima funkciju sistematičnog grupiranja wikilink-ova odnosno članaka. Isto kao kategorije i liste svaki template ima svoju stranicu. Glavna funkcija templates-a je da se u njih pregledno i kategorički uvrste svi važniji pojmovi vezani uz ime samog templatea.” [www.wikipedia.org, 2010.]

Na slici 5.6. su prikazane osnovne wiki sintakse, koje se koriste za oblikovanje sadržaja.

Wiki text	Result
<code>"italic"</code>	<i>italic</i>
<code>'''bold'''</code>	bold
<code>''''bold and italic''''</code>	<i>bold and italic</i>
<code>==heading==</code> <code>===level 2===</code> <code>====level 3====</code> <code>=====level 4=====</code>	Headings in different sizes
<code>[[Link to another page]]</code> <code>[[Link different title]]</code>	Internal Link to another page on the wiki
<code>http://www.test.org</code> <code>[http://www.test.org Text]</code>	External link Link with description
<code>[[fr:Page en français]]</code>	Interwiki link to french Wikipedia (appears under “languages”)
<code>[[Category:Example]]</code>	Add article to category “example”
<code>----</code>	horizontal line
<code>* one</code> <code>* two</code> <code>* three</code>	Bullet list
<code># one</code> <code># two</code> <code># three</code>	Numbered list

Slika 5.6. Osnovne wiki sintakse [www.wikipedia.org, 2010.]

6. Primjer

Za primjer sam napravio internetski forum u PHP, te MySQL-u kao bazi podataka. Za MySQL sam koristio PHPMyAdmin grafičko sučelje. Programski kod sam postavio na SourceForge.net²², te sam koristio SVN za kontrolu verzija. Sve sam to dokumentirao uz pomoć wiki sustava u programu MediaWiki.

„Internetski forum je usluga na Internetu koja omogućava razmjenu mišljenja među sudionicima uporabom web preglednika. Sve poruke koje korisnik napiše i pošalje na forum vidljive su svim ostalim sudionicima foruma. To nalikuje na oglasnu ploču (engl. message board) na kojoj sudionici ostavljaju poruke. U načelu poruke na forumu mogu ostavljati i čitati sudionici interneta bez ograničenja. Sudionici su najčešće anonimni jer se pri slanju poruke na forum ne mora navesti pravi identitet. Zbog lakšeg snalaženja forum je obično podijeljen u nekoliko skupina prema temama razgovora. Jednostavnost uporabe i mogućnost anonimne rasprave o različitim temama jedan je od glavnih razloga popularnosti foruma. Za internetski forum se dovoljno služiti web preglednikom. Poruka poslana na jedan forum vidljiva je samo na tom forumu. Forum se uglavnom sastoji od mnogobrojnih kategorija unutar kojih se nalaze teme koje otvaraju i započinju korisnici. Svaka tema ima svoju diskusiju u kojoj se sudjeluje pisanjem, odnosno objavljivanjem postova. Svi razgovori/diskusije ostaju sačuvani na forumu dok ih ne obriše moderator (stvarna osoba).“ [www.wikipedia.org, 2010.]

6.1. MySQL

Počnimo redom. Za forum je prvo potrebno napraviti bazu u MySQL-u kako bi podaci koji se budu postavljali na forum ostali negdje spremljeni. Kod baze je slijedećeg oblika:

```
CREATE TABLE `baza1`.`forum` (
  `id` INT( 20 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `autor` VARCHAR( 255 ) NOT NULL DEFAULT ' ',
  `naslov` VARCHAR( 255 ) NOT NULL DEFAULT ' ',
  `diskusija` MEDIUMTEXT NOT NULL ,
  `vrijeme` VARCHAR( 255 ) NOT NULL DEFAULT ' ',
  `id_vrijeme` BIGINT( 20 ) NOT NULL DEFAULT '0',
  `osoba_odgovor` VARCHAR( 255 ) NOT NULL DEFAULT ' ',
  `br_odgovor` BIGINT( 20 ) NOT NULL DEFAULT '0',
  `id_roditelja` BIGINT( 20 ) NOT NULL DEFAULT '0',
  `id_vrijeme_odgovor` BIGINT( 20 ) NOT NULL DEFAULT '0'
);
```

²² <https://sourceforge.net/projects/hrforum/>

Baza sadrži stupce:

Id – primarni ključ koji je jedinstven za svaki komentar koji se postavlja na forumu.

Autor – autor članka na forumu

Naslov – naslov diskusije na forumu

Diskusija – tekst koji je autor postavio, bilo kao početak diskusije ili kao odgovor

Vrijeme – predstavlja vrijeme spremljeno u 'normalnom' formatu

Id_vrijeme – vrijeme predstavljeno u Unix formatu, omogućuje poveznicu između diskusije i posljednjeg odgovora

Osoba_odgovor – predstavlja osobu koja je zadnja odgovorila

Br_odgovora – Količina odgovora na postavljenu diskusiju

Id_roditelja – uspostavlja vezu između odgovora i početka diskusije

Id_vrijeme_odgovor – veza koja pokazuje na id_vrijeme posljednjeg odgovora

Slika baze unutar PHPMyAdmin izgleda kao na slici 6.1.

	Polje	Vrsta	Uspoređivanje	Atributi	Null	Zadano	Dodatno	Aktivnost							
☐	id	int(20)			Ne	None	auto_increment								
☐	autor	varchar(255)	latin2_croatian_ci		Ne										
☐	naslov	varchar(255)	latin2_croatian_ci		Ne										
☐	diskusija	mediumtext	latin2_croatian_ci		Ne	None									
☐	vrijeme	varchar(255)	latin2_croatian_ci		Ne										
☐	id_vrijeme	bigint(20)			Ne	0									
☐	osoba_odgovor	varchar(255)	latin2_croatian_ci		Ne										
☐	br_odgovor	bigint(20)			Ne	0									
☐	id_roditelja	bigint(20)			Ne	0									
☐	id_vrijeme_odgovor	bigint(20)			Ne	0									
↑ Označi sve / Ukloni sve oznake S odabirom:															

Slika 6.1. Tablica forum unutar programa PHPMyAdmin

6.2. PHP

Nakon napravljene baze možemo iskoristiti i drugi open source program. To je PHP. Za dizajn izgleda unutar PHP sam skinuo dokument „Style.css“:

```

<!--
body {

    a:link, a:visited, a:active { text-decoration: none}

    font-family:Verdana, Sans-serif;

    color: #000000;

    font-size: 12px
}

input,textarea, select,{
    color : #000000;
    font: normal 12px;
    border-collapse: collapse; border: 1px solid #000000;
}
.maintable {border: 10px ; width: 100%; padding: 15px; background-color: #FFFFFF} /*main
table for forum*/
.regrow {font-family: Verdana,Sans-serif; color: #000000; font-weight: bold; background-color:
#FFFFFF;font-size: 12px;} /*registration row, mainly here for symetry*/
.headline {font-family: Verdana,Sans-serif;font-weight: bold;color: #FFFFFF;background-
color: #A9A9A9;font-size: 11px;} /*headline row, the first row that says forum name, topics,
posts and such*/
.forumrow {font-family: Verdana,Sans-serif; color: #000000;background-color: #F2F2F2;font-
size: 12px;} /*color of the forum rows*/
.mainrow a:link, a:visited, a:active { text-decoration: none;}
.mainrow {font-family: Verdana,Sans-serif; color: #000000;background-color: #F2F2F2;font-
size: 12px; a:link, a:visited, a:active { text-decoration: none}} /*color of the forum rows*/
.maintables{background-color: #FFFFFF; width: 95%; padding: 0px; border: 1px solid;
cellspacing: no} /*main table for forum*/
--->[ www.barelyfitz.com, 2010.]

```

Za povezivanje sa bazom je bilo potrebno napraviti dokument unutar, te će se taj dokument koristiti pozivom u svim drugim dokumentima. Naziv dokumenta je „baza.php“:

```

<?php
$db = mysql_connect("localhost", "root", "") or die("Ne mogu se povezati!");
if(!$db)
    die("Nema odabranog servera!");

if(!mysql_select_db("Baza1",$db))
    die("Nije ispravno upisana baza!");
if(!get_magic_quotes_gpc()){
    $_GET = array_map('mysql_real_escape_string', $_GET);
    $_POST = array_map('mysql_real_escape_string', $_POST);
}

```

```

$_COOKIE = array_map('mysql_real_escape_string', $_COOKIE);
}
else{
$_GET = array_map('stripslashes', $_GET);
$_POST = array_map('stripslashes', $_POST);
$_COOKIE = array_map('stripslashes', $_COOKIE);
$_GET = array_map('mysql_real_escape_string', $_GET);
$_POST = array_map('mysql_real_escape_string', $_POST);
$_COOKIE = array_map('mysql_real_escape_string', $_COOKIE);
}
?>

```

Nakon svih dosadašnjih dokumenata konačno dolazimo do dokumenta koji će nam dati opipljive rezultate, odnosno koji će nam izvodeći PHP kod prikazati prvu vizualno vidljivu stranicu. To je naravno početna stranica pod nazivom „indeks.php“:

```

<?php
include "baza.php";
print "<link rel='stylesheet' href='style.css' type='text/css'>";
print "<A href='diskusija.php'>Nova diskusija</a><br>";
print "<table class='maintable'>";
print "        <tr                class='headline'><td                width=50%>Diskusija</td><td                width=20%>Pokrenuo</td><td>Broj                odgovora</td><td>Posljednji                put                odgovoreno</td></tr>";
$diskusija="SELECT * from forum where id_roditelja='0' order by id_vrijeme_odgovor DESC";
$skusija2=mysql_query($diskusija) or die("POGREŠKA!!! Ne mogu dobiti zahtjevane redove!!!");
while($diskusija3=mysql_fetch_array($diskusija2)){
    $diskusija3[naslov]=strip_tags($diskusija3[naslov]);
    $diskusija3[autor]=strip_tags($diskusija3[autor]);
    Print "        <tr                class='mainrow'><td><A                href='poruka.php?id=$diskusija3[id]'>$diskusija3[naslov]</a></td><td>$diskusija3[                autor]</td><td>$diskusija3[br_odgovor]</td><td>";
        echo date("d-m-Y h:i:s",$diskusija3[id_vrijeme_odgovor]);
        print"<br>Odgovoreno od <b>$diskusija3[osoba_odgovor]</b></td></tr>";
}
print "</table>";
?>

```

Vizualno stranica izgleda kao na slici 6.2.

Nova diskusija

Diskusija	Pokrenuo	Broj odgovora	Posljednji put odgovoreno
Tema3	Ljubica	0	10-02-2010 04:49:26 Odgovoreno od Ljubica
Tema2	Mario	0	10-02-2010 04:49:11 Odgovoreno od Mario
Tema1	Danijel	0	10-02-2010 04:48:49 Odgovoreno od Danijel

Slika 6.2. Vizualni izgled izvršavanja koda dokumenta „indeks.php“

Kao što je vidljivo na slici 6.2. početna stranica ima linkove „Nova diskusija“ i linkove na teme. Link „Nova diskusija“ povezuje početnu stranicu i dokument „diskusija.php“, dok se klikom na temu povezuje sa dokumentom „poruka.php?id“ gdje id predstavlja id teme koju otvaramo.

Dakle, kao što smo rekli, „diskusija.php“ se nadovezuje na početnu stranicu, te omogućava unos nove diskusije. Kod unutar dokumenta „diskusija.php“ izgleda ovako:

```
<?php
include "baza.php";
print "<link rel='stylesheet' href='style.css' type='text/css'>";
print "<table class='maintables'>";
print "<tr class='headline'><td>Upišite poruku</td></tr>";
print "<tr class='maintables'><td>";
if(isset($_POST['submit'])){
    $ime=$_POST['ime'];
    $poruka=$_POST['poruka'];
    $naslov=$_POST['naslov'];
    if(strlen($ime)<1) {
        print "Niste upisali ime.<br>";
        print "Povratak na <A href='indeks.php'>pocetnu</a><br>";
        print "Povratak na <A href='diskusija.php'>novu diskusija</a><br>";
    }
    else if(strlen($poruka)<1) {
        print "Niste upisali poruku.<br>";
        print "Povratak na <A href='indeks.php'>pocetnu</a><br>";
        print "Povratak na <A href='diskusija.php'>novu diskusija</a><br>";
    }
    else if(strlen($naslov)<1){
        print "Niste upisali naslov.<br>";
        print "Povratak na <A href='indeks.php'>pocetnu</a><br>";
        print "Povratak na <A href='diskusija.php'>novu diskusija</a><br>";
    }
    else {
        $datum=date("U");
```

```

    $vrijeme=date("d-m-Y h:i:s");
    $naslov=strip_tags($naslov);
    $ime=strip_tags($ime);
    $poruka=strip_tags($poruka);
    $insert="INSERT INTO forum(autor,naslov,diskusija,vrijeme,id_vrijeme,osoba_odgovor,
id_vrijeme_odgovor) values('$ime','$naslov','$poruka','$vrijeme','$datum','$ime', $datum)";
    mysql_query($insert) or die("POGREŠKA!!!");
    print "Poruka postavljena vratite se na <A href='indeks.php'>pocetnu</a>.";
}
}
else{
    print "<form action='diskusija.php' method='post'>";
    print "Vaše ime:<br>";
    print "<input type='text' name='ime' size='20'><br>";
    print "Naslov:<br>";
    print "<input type='text' name='naslov' size='20'><br>";
    print "Vaša poruka:<br>";
    print "<textarea name='poruka' rows='5' cols='40'></textarea><br>";
    print "<input type='submit' name='submit' value='SPREMI'></form>";
}
print "</td></tr></table>";
?>

```

Otvaranjem unutar pretraživača dokument „diskusija.php“ dobivamo formu za unos nove diskusije, te se ispravnim unosom u sva polja forme ti podaci spremaju u bazu.

Vizualni izgled poziva dokumenta „diskusija.php“ izgleda kao na slici 6.3.

Slika 6.3. Vizualni izgled izvršavanja koda dokumenta „diskusija.php“

Drugi dokument koji se nadovezuje na početni je „poruka.php“. Kod tog dokumenta izgleda ovako:

```
<?php
include "baza.php";
$id=$_GET['id'];
print "<link rel='stylesheet' href='style.css' type='text/css'>";
print "<A href='indeks.php'>Pocetna</a>-<A href='diskusija.php'>Nova Diskusija</a>";
print "<table class='maintable'>";
print "<tr class='headline'><td width=20%>Autor</td><td
width=80%>Diskusija</td></tr>";
$diskusija="SELECT * from forum where id='$id'";
$diskusija2=mysql_query($diskusija) or die("POGREŠKA!!! Ne mogu otvoriti diskusiju!");
$diskusija3=mysql_fetch_array($diskusija2);
print "<tr class='mainrow'><td valign='top'>$diskusija3[autor]</td><td
vakign='top'>Posljednji put odgovoreno $diskusija3[vrijeme]<br><hr>";
$poruka=strip_tags($diskusija3['diskusija']);
$poruka=nl2br($poruka);
print "$poruka<hr><br>";
print "</td></tr>";
$odgovor="Select * from forum where id_roditelja='$id' order by id asc"; //getting replies
$odgovor2=mysql_query($odgovor) or die("POGREŠKA!!! Ne mogu dobiti odgovore!");
while($odgovor3=mysql_fetch_array($odgovor2)){
    print "<tr class='mainrow'><td valign='top'>$odgovor3[autor]</td><td
vakign='top'>Posljednji put odgovoreno $odgovor3[vrijeme]<br><hr>";
    $poruka=strip_tags($odgovor3['diskusija']);
    $poruka=nl2br($poruka);
    print "$poruka<hr><br>";
    print "</td></tr>";
}
```

```
print "</table>";
print "<A href='odgovor.php?id=$id'>Odgovor<br>"
?>
```

Kod dokumenta „poruka.php“ omogućava prikaz svih odgovora unutar neke diskusije, tako što sve potrebne podatke za tu diskusiju poziva iz baze. Vizualno izvršavanje tog koda dobijemo situaciju kao na slici 6.4.

Pocetna-Nova Diskusija

Autor	Diskusija
Danijel	Posljednji put odgovoreno 10-02-2010 04:48:49
	poruka1
Mario	Posljednji put odgovoreno 10-02-2010 05:03:14
	poruka2
Ljubica	Posljednji put odgovoreno 10-02-2010 05:03:25
	poruka3

Odgovor

Slika 6.4. Vizualni izgled izvršavanja koda dokumenta „poruka.php“

Na prethodni dokument se nadovezuje dokument odgovor.php, koji je dosta sličan dokumentu diskusija.php:

```
<?php
include "baza.php";
print "<link rel='stylesheet' href='style.css' type='text/css'>";
print "<A href='indeks.php'>Pocetna</a>-<A href='diskusija.php'>Nova Diskusija</a>";
print "<table class='maintables'>";
print "<tr class='headline'><td>Odgovor</td></tr>";
print "<tr class='maintables'><td>";
if(isset($_POST['submit'])){
    $ime=$_POST['ime'];
    $poruka=$_POST['poruka'];
    $naslov=$_POST['naslov'];
    $id=$_POST['id'];
    if(strlen($ime)<1){
        print "Niste upisali ime!<br>";
        print "Povratak na <A href='indeks.php'>pocetnu</a><br>";
        print "Povratak na <A href='poruka.php?id=$id'>diskusiju</a><br>";
    }
    else if(strlen($poruka)<1){
```

```

    print "Niste upisali poruku!<br>";
    print "Povratak na <A href='indeks.php'>pocetnu</a><br>";
    print "Povratak na <A href='poruka.php?id=$id'>diskusiju</a><br>";
}
else{
    $datum=date("U");
    $vrijeme=date("d-m-Y h:i:s");
    $naslov=strip_tags($naslov);
    $ime=strip_tags($ime);
    $poruka=strip_tags($poruka);
    $insert="INSERT INTO
forum(autor,naslov,diskusija,vrijeme,id_vrijeme,osoba_odgovor,id_roditelja)
values('$ime','$naslov','$poruka','$vrijeme','$datum','$ime','$id')";
    mysql_query($insert) or die("POGREŠKA!!!");
    $updatepost="Update forum set br_odgovor=br_odgovor+'1', osoba_odgovor='$ime',
id_vrijeme_odgovor='$datum' where id='$id'";
    mysql_query($updatepost) or die("POGREŠKA!!!");

    print "Poruka postavljena, vratite se na <A href='poruka.php?id=$id'>diskusiju</a>.";
}
}
else{
    $id=$_GET['id'];
    print "<form action='odgovor.php' method='post'>";
    print "<input type='hidden' name='id' value='$id'>";
    print "Vaše ime:<br>";
    print "<input type='text' name='ime' size='20'><br>";
    print "Odgovor:<br>";
    print "<textarea name='poruka' rows='5' cols='40'></textarea><br>";
    print "<input type='submit' name='submit' value='SPREMI'></form>";
}
print "</td></tr></table>";
?>

```

Ovaj dokument dopušta unos odgovora na diskusiju koja je prethodno bila vidljiva. Pošto je dosta sličan dokumentu „diskusija.php“ nećemo vizualno prikazivati taj dokument.

6.3. SF.net

Slijedeći korak nakon stvaranja baze i PHP dokumenata, te isprobavanja rad, bio je postavljanje rada na stranice SourceForge.net, te je s tim program ušao u open source software. Stranica na kojoj je postavljen taj program je <http://hrforum.sourceforge.net/>. Program je postavljen pod GPL licencom. Na slici 6.5. su vidljive pogodnosti SF.net stranice uključene u moj projekt. To je samo dio pogodnosti koje nudi stranica SF.net.

The following features are currently enabled:

Enabled	Feature	Category	Status	Options
☒	Backups	Data Recovery	Always On	XML Export, Hosted Apps
☒	File Manager	Download	Always On	Manage
☒	Help Wanted	Recruiting	Always On	Submit, Manage
☒	Project Database (MySQL)	Web Hosting	Always On	Manage
☒	Project Web	Web Hosting	Always On	
☒	Virtual Hosts (VHOSTs)	Web Hosting	Always On	Manage
☒	Forums	Forums	Enabled	Manage
☒	Mailing Lists	E-Mail	Enabled	Manage
☒	MediaWiki	Wiki	Enabled	Manage, Grant Admin
☒	Project News	News / Blog	Enabled	Submit, Manage
☒	Screenshots	Media	Enabled	Manage
☒	Subversion	Source Control	Enabled	Manage
☒	Tracker	Bug / Defects	Enabled	Manage

Slika 6.5 Pogodnosti uključene u projekt „hrforum“ na SF.net-u

6.4. SVN

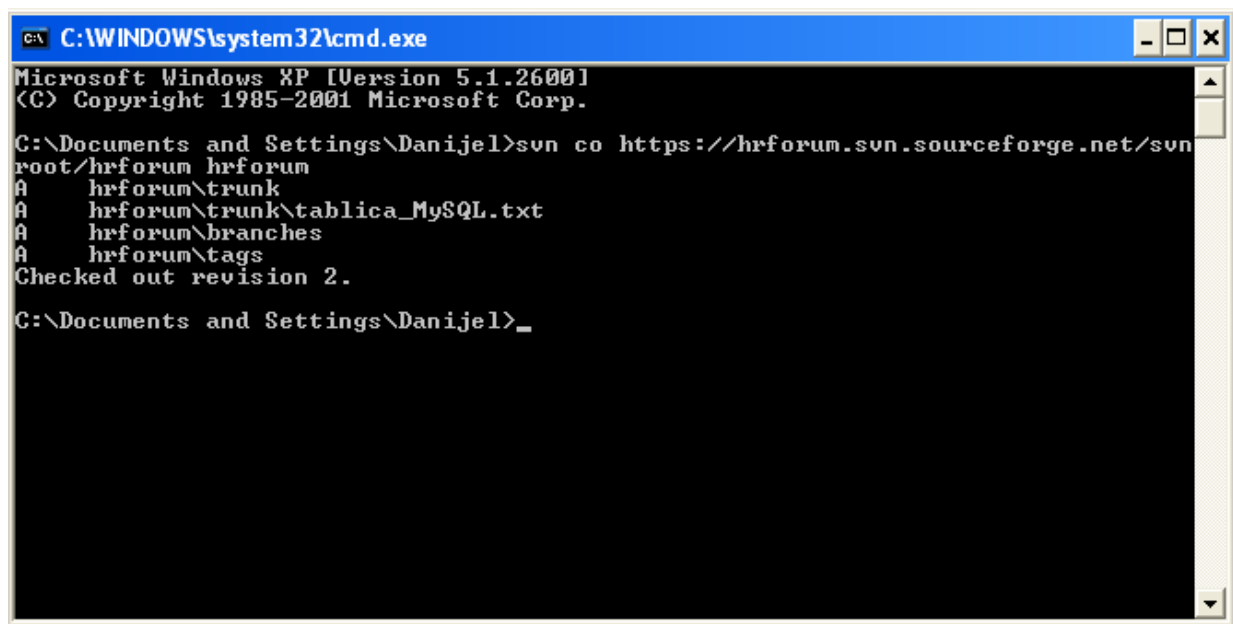
Kao što je prethodno vidljivo, na slici 6.5., na SF.net u sklopu moga projekta je uključen Subversion. Ranije smo objasnili što je SVN, a sada ćemo prikazati nekoliko jednostavnih naredbi za rad sa SVN-om.

Prilikom prijavljivanja na SF.net i uključivanja opcije Subverzion, automatski se dobiva repozitorij. Repozitorij za moj projekat je postavljen na Internet stranicu <https://hrforum.svn.sourceforge.net/svnroot/hrforum>. Sadržaj tog repozitorija se može pregledati preko Command Prompta ili odlazkom na prethodni link. U ovom primjeru ćemo vidjeti obe opcije.

6.4.1. SVN u Command Promptu

Repozitoriju kojeg sam dobio prilikom postavljanja projekta na SF.net se pristupa preko naredbe: `svn co https://hrforum.svn.sourceforge.net/svnroot/hrforum hrforum`.

Upisivanjem ove naredbe u CMD²³ dobije se sadržaj cijelog repozitorija. Na slici 6.6. je vidljiv sadržaj repozitorija, u kojem se nalazi tablica_MySQL.txt, te su napravljene dvije revizije. Prva prilikom samog postavljanja dokumenta, a druga ispravljanjem tog dokumenta.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Danijel>svn co https://hrforum.svn.sourceforge.net/svnroot/hrforum hrforum
A      hrforum\trunk
A      hrforum\trunk\tablica_MySQL.txt
A      hrforum\branches
A      hrforum\tags
Checked out revision 2.

C:\Documents and Settings\Danijel>_
```

Slika 6.6. pozivanje naredbe svn co unutar CMD-a

Ono što je bitnije od gledanja sadržaja repozitorija preko CMD-a je njegovo postavljanje, preuzimanje, te nakon toga sama ispravka i postavljanje natrag u repozitorij.

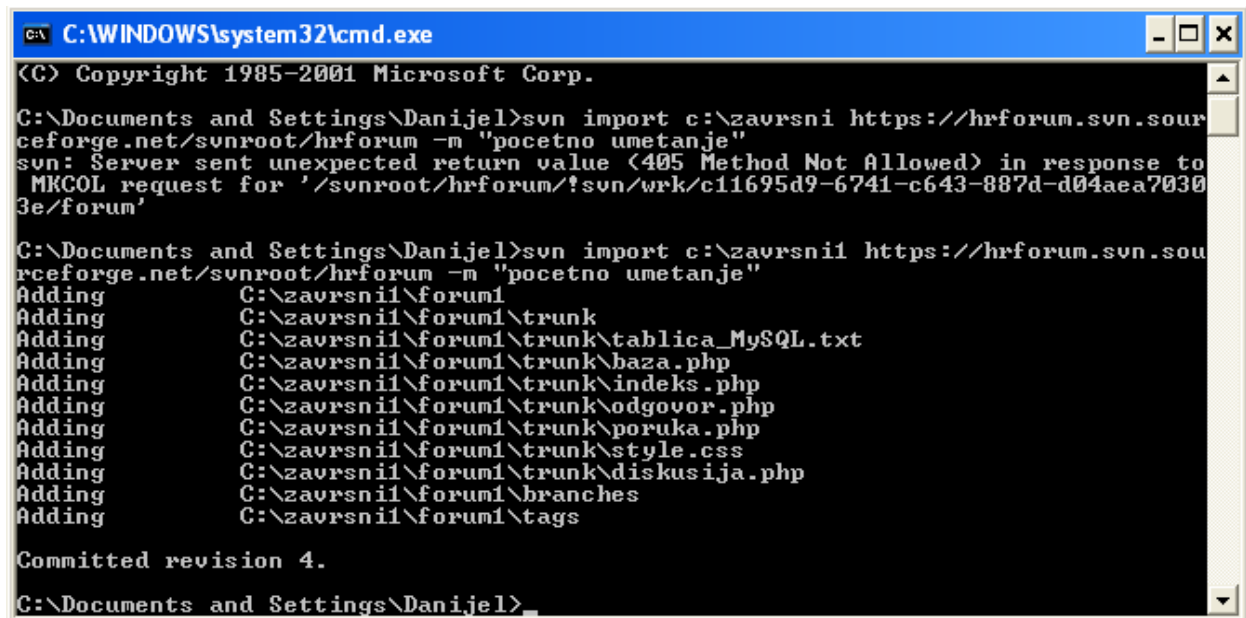
Pretpostavimo da imamo direktorij C:\zavrsni1\forum1 te unutar njega direktorije trunk, branches i tags (prilikom rada sa repozitorijem ta tri direktorija predstavljaju nepisani standard), te se dokumenti od našega projekta nalaze u direktoriju trunk (također nepisani standard).

Postavljanje direktorija u repozitorij se obavlja naredbom svn import. Točnije za naš projekt je potrebno upisati:

```
svn import c:\zavrsni1 https://hrforum.svn.sourceforge.net/svnroot/hrforum -m „pocetno umetanje“
```

Na slici 6.7. je vidljivo izvođenje tog upita u CMD-u. Budući da sam prethodno upisao direktorij c:\zavrsni vidljivo je i kako ne dopušta ponovno upisivanje tog direktorija.

²³ Command prompt



```
C:\WINDOWS\system32\cmd.exe
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Danijel>svn import c:\zavrsni https://hrforum.svn.sourceforge.net/svnroot/hrforum -m "pocetno umetanje"
svn: Server sent unexpected return value (405 Method Not Allowed) in response to MKCOL request for '/svnroot/hrforum/?svn/wrk/c11695d9-6741-c643-887d-d04aea70303e/forum'

C:\Documents and Settings\Danijel>svn import c:\zavrsni1 https://hrforum.svn.sourceforge.net/svnroot/hrforum -m "pocetno umetanje"
Adding C:\zavrsni1\forum1
Adding C:\zavrsni1\forum1\trunk
Adding C:\zavrsni1\forum1\trunk\tablica_MySQL.txt
Adding C:\zavrsni1\forum1\trunk\baza.php
Adding C:\zavrsni1\forum1\trunk\indeks.php
Adding C:\zavrsni1\forum1\trunk\odgovor.php
Adding C:\zavrsni1\forum1\trunk\poruka.php
Adding C:\zavrsni1\forum1\trunk\style.css
Adding C:\zavrsni1\forum1\trunk\diskusija.php
Adding C:\zavrsni1\forum1\branches
Adding C:\zavrsni1\forum1\tags

Committed revision 4.

C:\Documents and Settings\Danijel>
```

Slika 6.7. pozivanje naredbe svn import unutar CMD-a

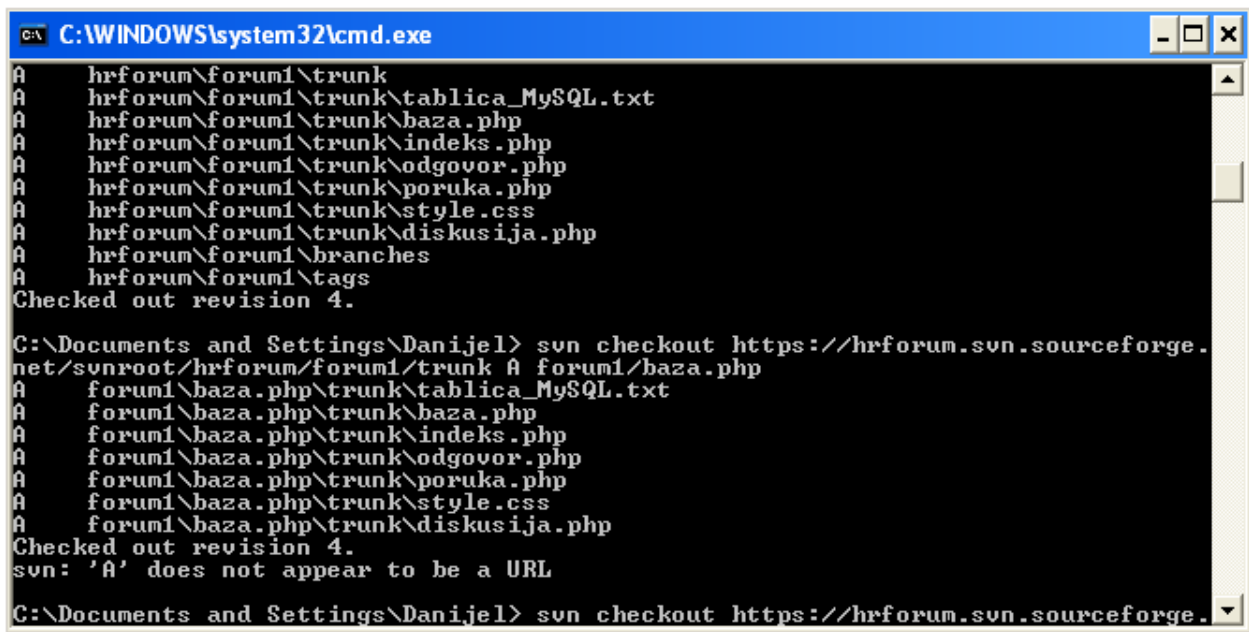
Dakle, naredbom svn import smo umetnuli direktorij forum u repozitorij pod imenom hrforum. Prekidač -m služi za dodavanje poruke u zapisnik, što će nam biti potrebije kod vizualnog repozitorija na stranicama SF.net.

Da bi mogli uređivati dokumente iz repozitorija potrebno ih je ‘skinuti’ na lokalno računalo u radnu kopiju. Radna kopija je privatna i sve promjene koje se u njoj naprave neće biti dostupne drugima tako dugo dok se eksplicitno ne stave u repozitorij. Također je moguće imati više radnih kopija istog projekta iskopiranog iz repozitorija.

Naredba za uzimanje dokumenata iz repozitorija je svn checkout. U našem slučaju trebamo pozvati naredbu:

```
svn checkout https://hrforum.svn.sourceforge.net/svnroot/hrforum/forum1/trunk forum1/baza.php
```

Naredbom svn checkout se stvara radna kopija cijelog direktorija trunk. Na slici 6.7. je vidljivo izvršavanje naredbe svn checkout.



```
C:\WINDOWS\system32\cmd.exe
A hrforum\forum1\trunk
A hrforum\forum1\trunk\tablica_MySQL.txt
A hrforum\forum1\trunk\baza.php
A hrforum\forum1\trunk\indeks.php
A hrforum\forum1\trunk\odgovor.php
A hrforum\forum1\trunk\poruka.php
A hrforum\forum1\trunk\style.css
A hrforum\forum1\trunk\diskusija.php
A hrforum\forum1\branches
A hrforum\forum1\tags
Checked out revision 4.

C:\Documents and Settings\Danijel> svn checkout https://hrforum.svn.sourceforge.net/svnroot/hrforum/forum1/trunk A forum1/baza.php
A forum1\baza.php\trunk\tablica_MySQL.txt
A forum1\baza.php\trunk\baza.php
A forum1\baza.php\trunk\indeks.php
A forum1\baza.php\trunk\odgovor.php
A forum1\baza.php\trunk\poruka.php
A forum1\baza.php\trunk\style.css
A forum1\baza.php\trunk\diskusija.php
Checked out revision 4.
svn: 'A' does not appear to be a URL

C:\Documents and Settings\Danijel> svn checkout https://hrforum.svn.sourceforge.net/svnroot/hrforum/forum1/trunk A forum1/baza.php
```

Slika 6.7. pozivanje naredbe svn import unutar CMD-a

Izvršavanjem naredbe `svn checkout` sa slike 6.7. se stvorio direktorij “C:\Documents and Settings\Danijel\forum1\baza.php\trunk”, te se unutar njega nalaze svi dokumenti projekta, a uz njih se nalazi i direktorij `.svn`. Direktorij `.svn` je skriveni direktorij, tzv. administrativni direktorij radne kopije, koji služi Subversionu da prepozna koje datoteke sadrže neobjavljene promjene i koje su datoteke zastarjele u odnosu na one u repozitoriju. Ovi dokumenti u radnoj kopiji se mogu mijenjati, te takvi promijenjeni opet staviti u repozitorij.

Pretpostavimo da dokumentu `indeks.php` dodamo red: `print "Pocetna stranica<br>";`. Tada taj dokument možemo vratiti u repozitorij naredbom `svn commit`. U našem slučaju treba upisati:

```
svn commit indeks.php -m “dodana linija print”
```

Na slici 6.8. je vidljiv prijenos dokumenta `indeks.php` u repozitorij. Prekidač `-m` označava, kao i prethodno njegovo korištenje, upisivanje poruke u zapisnik.

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Danijel>cd forum1
C:\Documents and Settings\Danijel\forum1>cd baza.php
C:\Documents and Settings\Danijel\forum1\baza.php>cd trunk
C:\Documents and Settings\Danijel\forum1\baza.php\trunk>svn commit tablica_MySQL
.txt -m "dodana linija"
Sending          tablica_MySQL.txt
Transmitting file data .
Committed revision 5.
C:\Documents and Settings\Danijel\forum1\baza.php\trunk>svn commit indek.php -m
"dodana linija print"
svn: Commit failed (details follow):
svn: 'C:\Documents and Settings\Danijel\forum1\baza.php\trunk\indek.php' is not
under version control
C:\Documents and Settings\Danijel\forum1\baza.php\trunk>svn commit indeks.php -m
"dodana linija print"
Sending          indeks.php
Transmitting file data .
Committed revision 6.
C:\Documents and Settings\Danijel\forum1\baza.php\trunk>
```

Slika 6.8. pozivanje naredbe svn commit unutar CMD-a

Pomoću command prompta je moguće i pozivanje prethodnih revizija. Na primjer da želimo ponovno pozvati onu reviziju koju smo pozivali prije prethodnih promjena i da je želimo spremiti unutar direktorija revl upisali bismo:

```
svn checkout -r 4 https://hrforum.svn.sourceforge.net/svnroot/hrforum revl
```

6.4.2 SVN na SF.net-u

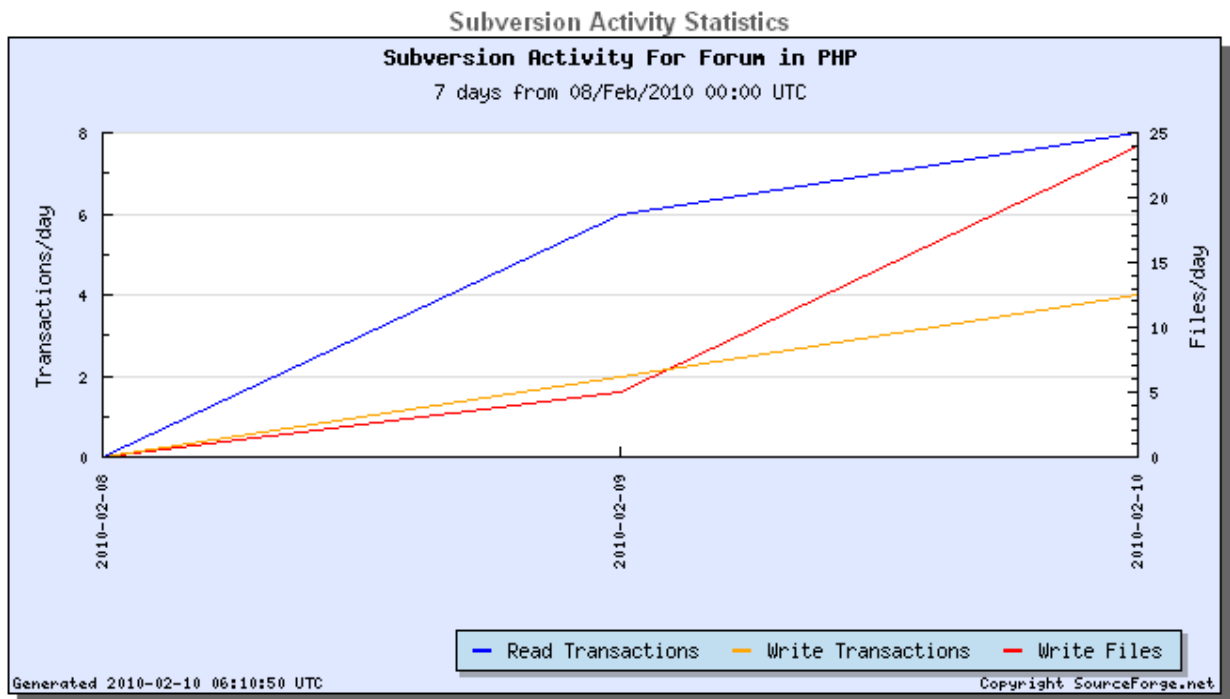
Repozitorij SVN-a je i vizualno dostupan na SF.net stranicama. Moj projekt je dostupan na linku <https://hrforum.svn.sourceforge.net/svnroot/hrforum>.

File ▲	Rev.	Age	Author	Last log entry
branches/	1	15 hours	danijelgalic	inicijalno umetanje
forum/	3	72 minutes	danijelgalic	pocetno umetanje dokumenata
forum1/	6	16 minutes	danijelgalic	dodana linija print
tags/	1	15 hours	danijelgalic	inicijalno umetanje
trunk/	2	15 hours	danijelgalic	dodana linija

6.9. Vizualni prikaz repozitoriaj SVN na SF.net

Na slici 6.9. je vidljiv repozitorij mog projekta na stranici SF.net, kao i zadnje promjene na dokumentima. To su one promjene koje smo radili unutar CMD-a.

Taj vizualni pristup SVN-u na SF.net je dobar zbog vođenja nekih statističkih detalja, budući da SF.net ima jako dobro statističko sučelje SVN-a. Slika 6.10. grafički prikazuje moje aktivnosti u posljednjih tjedan dana.



Slika 6.10. Grafički prikaz aktivnosti u SVN-u na SF.net-u

6.5. Wiki

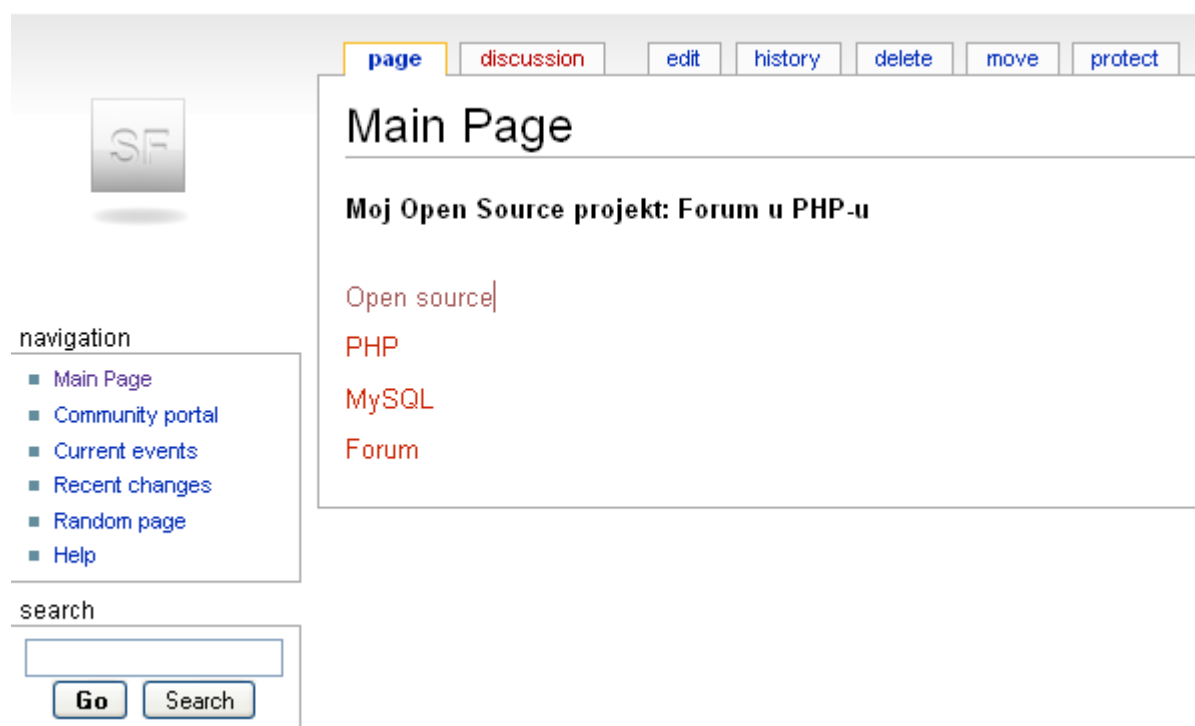
SF.net ima podršku od MediaWiki sustava na kojem se također uređuje i Wikipedija.

Wiki sustav mog projekta je na linku:

https://sourceforge.net/apps/mediawiki/hrforum/index.php?title=Main_Page

Budući da je sve o Wiki sustavu objašnjeno u prethodnim djelovima ovdje nećemo to ponovno objašnjavati, već ću samo prikazati početnu stranicu. Na slici 6.11. je prikazana početna stranica wiki sustava.

Na wikiju su objašnjene iste stvari koje se nalaze i u ovom projektu.



Slika 6.11. Početna stranica mog wiki sustava

7. Procjena alata

Procjenu alata ćemo najbolje objasniti tako da usporedimo alate sa njihovim konkurentima u open sourcu, pa ćemo u zaključku procjeniti open source kao sustav.

7.1. PHP i ostali konkurenti

Prvi konkurent PHP-u je Perl. Perl je skriptni jezik opće namjene što je dobro znati kada želite brzo napisati jednodne skripte za kompleksne transformacije teksta ili druge zadatke. Ima veliku bazu korisnika i tisuće modula koji omogućavaju provođenje kompliciranih zadataka u nekoliko redova. Perl će doduše obično biti sporiji od PHP-a i Pythona okruženju web aplikacija, no ako ga koristite s Apache-om, mod_perl jako ubrzava stvari. Perl je od milja nazvan "Švicarska motorna pila programskih jezika" zbog njegovih mogućnosti, fleksibilnosti i prilagodljivosti. Mana PHP-a je što nije previše koristan izvan okvira mrežnih aplikacija. Dokumentacija za stotine ugrađenih funkcija je dobra, no obično je puno teže pronaći i instalirati dobre PHP module nego Python i Perl module. No, ugrađene funkcije omogućuju vrlo laku provedbu većine mrežno-orijentiranih zadataka, kad bi ste trebali tražiti odgovarajuće Python i Perl module. PHP i Perl često rade jedno uz drugo, pošto se oba izvode na poslužitelju, no PHP ima prednost pred Perl-om na većini mrežnih stranica jer je PHP predprocesor obično dio mrežnih servera.

Drugi PHP konkurent je Python. Python je programski jezik opće namjene koji se lako koristi za obradu teksta, brojeva, slika, znanstvenih podataka, ili bilo čega što netko može pospremiti na računalo. Na dnevnoj bazi se koristi u operacijama Googleovog mehanizma za pretraživanje, Youtube-a, NASA-e itd.

Dakle, kada je riječ o mreži tu je PHP najbolji alat, ali ukoliko se želi bilo što drugo raditi sa njim to je već problem i bolje je opcija izabrati drugi programski jezik.

7.2. MySQL i PostgreSQL

Kao i kod svakog programa tako i kod ova dva, njihovo korištenje ovisi o potrebama. Svaka od navedenih baza podataka ima svoje prednosti i nedostatke koje ćemo ukratko opisati u ovom dijelu.

Uspoređivanje programa za baze podataka se obavlja preko ACID testa koji opisuje baze na temelju četiri svojstva koja bi trebala imati robusna baza podataka:

- *atomicity* - atomarnost
- *consistency* - konzistencija
- *isolation* - izolacija
- *durability* – izdržljivost

Osnovna provjera ovih svojstava je transakcija koja je osnovna jedinica rada koju korisnik baze može definirati. Transakcija u sebi može sadržavati više operacija nad bazom podataka. Baza podržava atomarnost ako za pojedinu transakciju rezultat može biti samo pozitivan ili negativan. Npr., ako definiramo transakciju koja u sebi sadrži naredbe UPDATE, INSERT i DELETE, sve tri naredbe promatraju se kao cjelina. Ili će se izvršiti sve, ili nijedna. Ovo svojstvo bitno je kod novčanih transakcija kada nije poželjno da se nakon DELETE naredbe u slučaju pada server naredba INSERT ne izvrši.

PostgreSQL je prema ovim svojstvima bolji od MySQL-a. PostgreSQL je ACID kompatibilan pa njegovo korištenje osigurava napredne opcije i prije svega sigurnost podataka. Ipak, upotreba mnoštva naprednih opcija usporava izvedbu aplikacije. MySQL po defaultu od ACID svojstava podržava samo atomarnost pomoću opcije table locks. Ostala ACID svojstva moguće je postići upotrebom različitih paketa za upravljanje tablicama poput BerkelyDB ili InnoDB.

Još su neke osobine bitne kod ovih baza, a koje se ne spominju u ACID testu. Tako moramo znati da MySQL ne podržava storne procedure niti ugnježdene upite, dok PostgreSQL to podržava, što mu daje znatnu prednost prilikom izrade velikih aplikacija u kojima se traži brzina. Storne procedure veoma ubrzavaju rad aplikacija pisanih u Javi. Java je poznata po svom sporijem izvođenju, pa je pomoću stornih procedura moguće određene funkcije staviti direktno na bazu. Zbog navedenih karakteristika, PostgreSQL preporučuje se korisnicima orijentiranim prema razvoju kompleksnih aplikacija ili onima koji prelaze sa ORACLE, Sysbase ili Microsoft SQL Server-a na drugu bazu podataka. Ako bazu podataka koristimo za manje zahtjevne potrebe (npr. Integracija sa webom) preporučuje se korištenje MySQL-a. Iako postoji mogućnost gubitka podataka ako npr. isključimo napajanje za vrijeme izvršenja transakcije, veća brzina i jednostavnija upotreba daju prednost MySQL-u kod manjih baza podataka u jednostavnijim sustavima.

Iz ovoga proizlazi da je MySQL bolji gledajući u pogledu našega projekta, dok je za neke naprednije stvari PostgreSQL ipak ispred MySQL-a.

7.3. SVN i CVS

Jednostavno rečeno SVN je napredniji od CVS. Razlika je što SVN omogućava neke funkcije koje CVS nema. Subversion omogućava:

1. Kontrolu verzij e datoteka i direktorija, dok CVS samo kontrolu verzije datoteka
2. Dodavanje, brisanje, kopiranje i preimenovanje datoteka i direktorija, dok je CVS-ova podrška za to loša
3. Pristup repozitoriju preko mreže
4. Prijenos svih ili ni jednih promjena u repozitorij, dok CVS to ne podržava

Konzistenciju prilikom prikazivanja razlika između datoteka koristeći binarni algoritam

8. Zaključak

Open source predstavlja jako dobru alternativu komercijalnom softwaru. Ni sam nisam slutio, kada sam počeo raditi na ovom radu, koje su mogućnosti open sourca. Misleći na open source većina ljudi, pa sam tako i sam, zamišljaju neke osnovne programe, ali radeći na ovom radu uvjerio sam se suprotno.

Ipak, iako open source programi predstavljaju jako dobru alternativu komercijalnom softwaru, ništa se neće promijeniti sve dok velike tvrtke ne počnu primjenu tih programa, a to će jako teško ići. Velike tvrtke već imaju razvijene programe, koje su im razvile komercijalne software tvrtke, te im je jako teško prijeći na open source programe, budući da nije sigurno da će im svi programi biti kompatibilni. Tvrtke koriste mnoge programe koji nisu kompatibilni sa Linux sustavom, pa su primorani na Windows.

U našoj regiji je piratstvo donekle i „legalno“, jer se nitko ne osvrće na to, pa tvrtke i pojedinci koriste komercijalne softwaere besplatno, preuzimajući software sa interneta. Zbog toga u našoj regiji se situacija u primjeni open sourca neće ubrzo ni popravljati, ali svjetske kompanije tu mogu učiniti veliki korak i smanjiti izdake za software rješenja.

U razvoj open sourca se mora uključiti država, pa bi se open source puno brže probio u upotrebu. Država bi trebala posebno financirati open source projekte, pa bi i programeri imali koristi od razvoja takvog softwarea.

Literatura

1. Gilmore J. (2008.) *Beginning PHP and MySQL: from novice to professional (Third edition)*. SAD: Apress.
2. Colins-Sussman B., Fitzpatrick B., Pilatio M(2006)., *Version Control with subversion: For Subversion 1.5: (Compiled from r3305)*. SAD: Red Bean.
3. 2004., *Free Software / Open Source: Information Society Opportunities for Europe (version 1.2.)*, SAD: Libre
4. Balatinac J. (2009), *OpenSource licence*
5. Radivojnović B (2009.), *Open Source pokret*,Srbija
6. Uran F. (2009.), *Open Source VPN (stručni rad)*
7. www.omegamagazin.com, 10.2.2010.
8. www.distrowatch.com, 10.2.2010.
9. www.linux.hr, 10.2.2010.
10. <http://lokalizacija.linux.hr>, 10.2.2010.
11. <http://happypenguin.net>
12. www.e-hrvatska.hr, 10.2.2010.
13. www.opensource.org, 10.2.2010.
14. www.php.co.ba, 10.2.2010.
15. www.w3schools.com/php/, 10.2.2010.
16. www.soros.org, 10.2.2010.
17. php.com.hr, 10.2.2010.
18. Valade J.(2002), *PHP & MySQL for dummies*, NewYork: Wiley Publishing.
19. Kofler M.(2001), *MySQL*, Berkeley: Apress
20. www.php.com, 10.2.2010.
21. http://en.wikipedia.org/wiki/Apache_HTTP_Server, 10.2.2010.
22. Robeli M. (2002), *PHP (seminarski rad)*

23. Bagarić, Bačić, Bičani, Bernić (2008), *Php programski jezik (seminaski rad)*
24. www.dead-pixel.info, 10.2.2010.
25. www.irt.org, 10.2.2010
26. www.syntaxpolice.org, 10.2.2010
27. SourceForge.net, 10.2.2010.
28. www.wikipedia.org, 10.2.2010.
29. www.barelyfitz.com, 10.2.2010.
30. <https://hrforum.svn.sourceforge.net/svnroot/hrforum>, 10.2.2010.