

Het Open Source Fenomeen

Jos Visser
josv@osp.nl

Open Solution Providers
<http://www.osp.nl>

OPEN SOLUTION PROVIDERS

19 november 2002

© Copyright 2002 Uniteam B.V.

Open Solution Providers is een handelsnaam van Uniteam B.V.

Dit document is gemaakt met VIM¹ en L^AT_EX². Never change a winning team!

De diverse handelsmerken gebruikt in dit document behoren toe aan hun respectievelijke eigenaren.

UniTeam B.V.
Dalsteindreef 16
1112 XC Diemen

Geformateerd op 19 november 2002.

Licentievoorwaarden

De copyright houder (Uniteam B.V.) verstrekt hierbij het algemene recht aan een ieder om dit document in zijn geheel en zonder wijzigingen te vermenigvuldigen en te distribueren in het elektronische formaat waarin u het heeft ontvangen en op papier. Aanpassingen en gedeeltelijke overname zijn niet toegestaan. Formaatconversies zijn alleen dan toegestaan indien dit geen onderdelen van het document onleesbaar maakt of versluiert. Op de geconverteerde versies van dit document zijn dezelfde voorwaarden van toepassing als op het origineel.

¹<http://www.vim.org>

²<http://http://www.latex-project.org>

Inhoudsopgave

1	Inleiding	5
1.1	Over Open Solution Providers en de auteur	6
2	Elf mythes over Open Source Software	7
3	Wat is Open Source Software (precies)?	10
3.1	Copyright en licenties	10
3.2	Typische licenties	11
3.3	Licentiemanagement	11
3.4	Entrée: Open Source Software	12
3.4.1	Commerciële Open Source Software	12
3.5	Open Source Software licenties	13
3.5.1	GNU GPL: Free software	13
3.5.2	BSD licentie: Copycenter	14
3.5.3	De Netscape Public License	14
3.6	Public Domain en shareware	15
3.7	Semi Open Source	15
4	Voorbeelden van Open Source Software	16
4.1	GNU software	16
4.2	Linux kernel	17
4.3	X Window System en aanverwante	17
4.4	BSD	18
4.5	Samba	18
4.6	Apache	18
4.7	OpenOffice	19
4.8	GIMP	19
4.9	XINE	20
4.10	XMMS	20
4.11	PostgreSQL	20
4.12	Perl	20
5	De organisatie van de Open Source gemeenschap	22
5.1	De Open Source ontwikkelmethodologie	23

5.2	Wie zijn de Open Source ontwikkelaars?	23
6	Kwaliteit van Open Source Software	24
6.1	Veiligheid van Open Source Software	24
7	Waarom ontwikkelt men Open Source Software ?	26
7.1	Therapeutische redenen	26
7.2	Scratching an itch.	26
7.3	Hardware leveranciers	27
7.4	Voor gebruikers	27
7.5	Strategische overwegingen	28
7.6	Open Source als business model	28
8	Waarom Open Source Software gebruiken?	29
8.1	Licentiekosten	29
8.2	Kwaliteit	29
8.3	Makkelijker verkrijgbaar	29
8.4	Betere beheersing van de oplossing	30
9	Aandachtspunten bij gebruik van Open Source Software	31
9.1	Leverancierondersteuning	31
9.2	Externe consultants	31
9.3	Feature shoot-out	32
10	Conclusie	33
A	Literatuurlijst	34

1 Inleiding

Al sinds het ontstaan der mensheid zijn er verlichte geesten die het leuk, aardig, plezant, interessant, statusverhogend of winstgevend vinden om de resultaten van hun intellectuele arbeid gratis en voor niets met anderen te delen. Kunstenaars en wetenschappers zijn daar excellente voorbeelden van, maar ook op andere gebieden laten getalenteerde individuen met regelmaat van zich laten horen en verblijden zij de mensheid met hun soms verbluffende inzichten en de vruchten van hun inspanningen.

De nieuwste rage op dit gebied is “Open Source Software” (OSS). Je kan al een jaartje of wat geen IT vakblad meer openslaan of er wordt verwezen naar Linux, Apache en andere prominente Open Source projecten. Je zou als leek geneigd zijn te denken dat Open Source Software iets van de afgelopen paar jaar is; een nieuwerwets concept dat met het verschijnen van Linux van start is gegaan en de wereld stormerderhand heeft veroverd. Niets is echter minder waar! Open Source Software is al zo oud als de IT zelf! Zonder dat de meeste mensen het weten vindt men Open Source Software in de kern van het Internet (toch ook al weer zo’n 30 jaar oud) en populaire besturingssystemen als Unix en Windows.

Kort door de bocht gesteld is Open Source Software software waarvan de broncode vrij beschikbaar is om te worden bestudeerd en aangepast. De software wordt ontwikkeld door vrijwilligers (zowel privé personen als werknemers van bedrijven in het kader van hun functie) die zich om één of andere reden geroepen voelen om bij te dragen aan de ontwikkeling of vervolmaking van de genoemde software. Het zou zo op het eerste gezicht toeschijnen dat op die basis nooit een compleet en praktisch bruikbaar systeem kan ontstaan. De ontwikkeling van software voor professionele doeleinden leek tot voor kort exclusieve domein van software bedrijven die voor het voorrecht om de aldus ontstane software te gebruiken soms flinke bedragen vroegen.

Met de algemene invoering van het Internet (in ongeveer 1994) is ook Open Source Software steeds nadrukkelijker in beeld gekomen. Zo rond de eeuwwisseling groeide die aandacht uit tot een ware hype. Tot verbazing van velen bleek er inmiddels een reusachtige verzameling van Open Source Software te zijn waarmee complete systemen konden worden gebouwd zonder ook maar één Euro aan software licenties uit te geven! De hype-achtige aandacht voor Open Source Software had trouwens ook tot gevolg dat veel bedrijven en ontwikkelaars zich tot het Open Source “geloof” bekeerden. Dit had weer tot gevolg dat veel nieuwe Open Source Software werd ontwikkeld en bestaande Closed Source Software Open Source werd gemaakt (b.v. de populaire Netscape browser).

De hype rondom Open Source Software heeft inmiddels dusdanige vormen aangenomen dat veel niet-IT organisaties zich zijn gaan afvragen of het niet verstandig is om voor het geheel of delen van hun IT-infrastructuur over te stappen op Open Source Software. Deze gedachte wordt natuurlijk gevoed door het feit dat grote leveranciers als Hewlett-Packard en IBM ook Open Source Software zijn gaan gebruiken, verkopen en ondersteunen.

Dergelijke beslissingen mogen natuurlijk niet lichtvaardig worden genomen. Open Source Software is een bijzondere categorie programmatuur en aan het gebruik er van kleven tamelijk specifieke voor- en (soms ook) nadelen. Helaas wordt de berichtgeving rondom Open Source nogal vaak gekleurd door mythes en vooroordelen. Commerciële software leveranciers hebben natuurlijk veel te verliezen en weinig te winnen bij de

opmars van Open Source Software . Zij doen dan ook vaak hun best om Open Source Software in discrediet te brengen³. Open Source proponenten hebben echter ook nog wel eens de neiging om de zaken rooskleuriger voor te stellen dan ze in werkelijkheid zijn.

In dit “white paper” wordt het Open Source fenomeen van diverse kanten belicht en worden antwoorden gegeven op de vragen die veelal leven bij diegenen die pas met Open Source Software in aanraking zijn gekomen. Aan de orde komen:

1. Wat is Open Source Software ?
2. Wie ontwikkelt Open Source Software ?
3. Wat zijn de voor- en nadelen van het gebruik van Open Source Software ?

1.1 Over Open Solution Providers en de auteur

Open Solution Providers⁴ (OSP) is een gespecialiseerd IT consultancy bedrijf met uitgebreide expertise op het gebied van ondermeer Unix, Linux, Internet, C/C++ en Java. Wij gebruiken zelf veel Open Source Software en passen het waar nuttig, nodig en toepasbaar ook toe in onze opdrachten. Voor meer informatie over de diensten van OSP kunt u contact opnemen met Erik Meinders (erik@osp.nl) of raadpleegt u onze web site.

Jos Visser⁵ is senior consultant op het gebied van Unix, Java, Internet en Open Source bij Open Solution Providers. Hij adviseert klanten, ontwikkelt software (Closed Source en Open Source), verzorgt trainingen en geeft presentaties en workshops op technische conferenties. Hij is voormalig bestuurslid van de Nederlandse Unix User Group (⁶) en initiator van de OpenComal⁷ en ProxyTunnel⁸ projecten.



Figuur 1: Jos Visser ziet af op Boat Quay in Singapore

³<http://www.opensource.org/halloween>

⁴<http://www.osp.nl>

⁵<http://www.josvisser.nl>

⁶<http://www.nluug.nl>

⁷<http://www.josvisser.nl/opencomal>

⁸<http://www.josvisser.nl/proxytunnel>

2 Elf mythes over Open Source Software

Alhoewel de rest van dit document uitgebreid ingaat op de diverse aspecten van Open Source Software wil ik graag beginnen met het ontzenuwen van elf bekende Open Source mythes:

1. Open Source Software wordt ontwikkeld door langharige hackers op cafeïne op slecht verlichte zolderkamertjes (zie ook sectie 5.2 op pagina 23).

De meeste Open Source Software wordt ontwikkeld door ervaren software engineers die bij gerenommeerde bedrijven op de loonlijst staan. Zij (of hun werkgevers) hebben meestal een goede reden om de aldus ontwikkelde software Open Source te maken en te houden.

Veel Open Source Software wordt inderdaad ontwikkeld door mensen met discutabele sociale vaardigheden, maar heeft u recentelijk eens een blik geworpen in de codemijnen van uw favoriete Closed Source Software leveranciers? :-)

2. Er is geen support voor Open Source Software (zie ook sectie 9.1 op pagina 31).

Dit is niet waar. Op de meeste Open Source Software kan een support contract worden afgesloten.

3. Open Source Software is een hype, maar wordt in feite niet breed gebruikt.

Wat statistieken: 95% van alle Internet mail wordt gerouteerd met Open Source producten (meestal Sendmail⁹ maar ook PostFix¹⁰ en anderen). De naamgeving van Internet en intranet hosts wordt vrijwel volledig verzorgd door de BIND¹¹ name server software. Apache¹² is 's werelds meest populaire web server (vaak op Open Source besturingssystemen zoals Linux en FreeBSD). De T_EX en L^AT_EX pakketten zijn krachtige tekstverwerkingspakketten die door ontelbare uitgevers worden gebruikt. En dan tot slot: Alle grafische Unix workstations draaien het Open Source X Window System.

4. De kwaliteit van Open Source Software is dubieus (zie ook sectie 6 op pagina 24).

Het tegenovergestelde is meestal het geval. Er is natuurlijk ook slechte Open Source Software, maar de meeste Open Source Software is van uitzonderlijke kwaliteit.

5. Open Source Software is niet veilig (zie ook sectie 6.1 op pagina 24).

Door de algemene beschikbaarheid van de broncode van Open Source Software is het tegenovergestelde meestal het geval! Beveiligingsproblemen worden in alle openbaarheid gevonden en opgelost. De in de Closed Source wereld niet ongebruikelijke praktijk van het geheimhouden van een beveiligingsprobleem is gewoonweg niet mogelijk. Hierdoor weet in principe iedereen (de "bad guys" en de "good guys") wat er aan de hand is (in tegenstelling tot alleen de "bad guys" en de leverancier (hopelijk)).

⁹<http://www.sendmail.org>

¹⁰<http://www.postfix.org>

¹¹<http://www.isc.org/products/bind>

¹²<http://www.apache.org>

6. Als je een probleem hebt met Open Source Software stel je vraag dan op het Internet en een antwoord verschijnt magischerwijs in je inbox.

Deze mythe is helaas ook nogal hardnekkig. Het is zeker zo dat de meeste Open Source Software door een actieve gemeenschap van gebruikers en ontwikkelaars wordt ondersteund. Echter, een organisatie zou natuurlijk wel gek zijn om daar al te zeer op te vertrouwen.

7. Open Source Software is niets voor ons want wij willen de source code toch niet aanpassen.

En gelukkig maar! Het te pas en te onpas aanpassen van de source code leidt tot versplintering en moeilijk te onderhouden software. De belangrijkste drijfveren voor het gebruik van Open Source Software vloeien voort uit de diverse voordelen van de algemene beschikbaarheid van de broncode maar omvatten niet de wens/mogelijkheid om de software zelf te hooi en te gras aan te passen.

8. Niemand verdient geld met Open Source Software .

De economische achtergrond van Open Source is een interessant gebied. Er is zeker geen economisch succes à la Microsoft of Oracle op het gebied van Open Source Software , maar er zijn diverse winstgevendende ondernemingen die Open Source Software ondersteunen en verkopen.

9. Open Source Software is niet van toepassing voor “mission critical” doeleinden.

Waarom niet? Eerder in deze sectie benoemde ik al de diverse brede toepassingen van bijvoorbeeld Sendmail, Apache en BIND (allen kritiek voor het Internet of specifieke ondernemingen). Diverse grote banken gebruiken Open Source componenten voor hun e-commerce omgevingen en de GUI van Lucent's NFM telefooncentrale software (onder andere in gebruik bij KPN Telecom) is gebaseerd op Apache. De geldautomaten van de Braziliaanse BanRisul bank lopen allemaal onder Linux.

10. Open Source Software is gratis, dus we kunnen met Open Source Software significante kostenreducties bewerkstelligen.

Dat ligt er maar aan. Meestal wel, soms niet. Het hangt er maar van af. Als je duizend keer Linux installeert in plaats van Windows 2000 dan bespaar je je inderdaad duizend W2K licenties (en de schepper mag weten hoeveel “Client Access Licenses”). Als we aan de andere kant één web server inrichten dan zijn de software licentiekosten verwaarloosbaar ten opzichte van de gehele implementatiekosten.

Gebruik van Open Source Software is meestal goedkoper, maar de kostenverdeling van IT is meestal zo diffuus dat het niet makkelijk te voorspellen is hoeveel goedkoper.

11. Open Source Software heeft geen leverancier en de continuïteit is dus niet gewaarborgd.

Open Source Software bestaat bij de gratie van het feit dat er gebruikers zijn. Als niemand het meer gebruikt zal de Open Source gemeenschap rondom die software vanzelf oplossen. Zolang er gebruikers zijn is er dus “leven in de brouwerij”. En, aangezien de broncode algemeen beschikbaar is heeft u tot aan het

einde der tijden de garantie dat aan de basisvoorwaarden voor het ondersteunen van de software is voldaan. Desnoods kunt u consultants inhuren om de ondersteuning te doen of om problemen op te lossen. Bij Closed Source Software die uit de handel is genomen heeft u deze mogelijkheid nooit en te nimmer.

Voorts wil ik er graag op wijzen dat de continuïteit van Closed Source Software ook allerm minst is gewaarborgd. Leveranciers hebben in nogal wat gevallen software die voorheen “strategisch” was verklaard alsnog uit de handel genomen. Ook in geval van faillissement of overname van de software leverancier staat u als gebruiker veelal “in de kou”.

3 Wat is Open Source Software (precies)?

Open Source Software is software waarvan de broncode vrij beschikbaar is. Er zijn echter veel manieren waarop broncode beschikbaar kan zijn en er is ook nog sprake van allerlei aan Open Source verwante verschijnselen zoals Free Software, Public Domain, Shared Source en Shareware.

Om aan alle verwarring een einde te maken heeft de Open Source Initiative¹³ (OSI) de “Open Source Definition” gepubliceerd. Zonder onmiddellijk in allerlei juridische touwtrekkerij te vervallen kunnen we stellen dat een software pakket pas Open Source Software is als het wordt geleverd met een software licentie die is goedgekeurd door de Open Source Initiative.

3.1 Copyright en licenties

Ook over software copyright en licenties is altijd veel verwarring. Om die reden wil ik eerst even ingaan op deze (helaas ietwat droge) materie alvorens verder te gaan met wat nu precies Open Source Software is.

Copyright¹⁴ (Nederlands: auteursrecht) is het automatische recht wat de auteur van een beschermd werk (b.v. boek, muziekstuk of computerprogramma) heeft om te bepalen hoe en onder welke voorwaarden dat werk mag worden gepubliceerd, gekopieerd en verspreid. Als ik een computerprogramma ontwikkel dan valt het copyright mij toe en mag ik in principe zelf bepalen wat er mee gebeurt. Op die toekenning van copyright zijn wat uitzonderingen. Indien ik een bijvoorbeeld een beschermd werk produceer als onderdeel van mijn baan dan komt het copyright in negen van de tien keer toe aan mijn werkgever. Verder is het mogelijk dat het copyright wordt overgedragen of verkocht aan een derde. De copyright houder heeft het exclusieve recht om te bepalen wie het werk mag gebruiken, en onder welke voorwaarden.

Op Open Source Software zit dus “gewoon” copyright. De copyright houders zijn meestal de ontwikkelaars of een stichting die voor dat doel is opgezet.

Een **licentie** is een overeenkomst tussen een copyright houder en een licentienemer waarin de licentienemer bepaalde rechten verkrijgt met betrekking tot het beschermde werk. Als ik een videoband koop dan krijg ik (tegen betaling) van de copyright houder het recht om het daarop vastgelegde werk in de huiselijke kring zo vaak te bekijken als ik maar wil. Die licentie verbiedt echter meestal dat ik de film publiekelijk vertoon of dat ik de band kopieer en doorverkoop.

Dienovereenkomstig zijn softwarelicenties overeenkomsten tussen de copyright houder van de software en de klant/gebruiker waarin wordt bepaald hoe en onder welke voorwaarden de software mag worden gebruikt (meestal na de betaling van de licentiekosten).

¹³<http://www.opensource.org>

¹⁴<http://www.templetons.com/brad/copymyths.html>

Let dus op dat u vrijwel nooit software (of andere beschermde werken) koopt! Wat u aanschaft is het recht om de beschermde werken op een bepaalde manier te vertonen of te gebruiken.

3.2 Typische licenties

De meeste commerciële licenties zitten ongeveer als volgt in elkaar:

- De gebruiker betaalt de copyright houder een bepaald bedrag aan geld.
- De gebruiker krijgt alleen de binaire (gecompileerde) versie van de software. Disassembleren, decompileren en andere reverse engineering is meestal verboden (voor zover wettelijk toegestaan).
- De software mag niet worden aangepast, doorverkocht, uitgeleend of gekopieerd.
- De gebruiker mag de software op één of meer computers uitvoeren.
- Garantie tot de deur.

Veel software licenties zijn enorm restrictief. Er staan niet zelden bepalingen in die wettelijk helemaal niet zijn toegestaan. Zo is het in Europa bijvoorbeeld wettelijk toegestaan een programma te ontleden (reverse engineering) als het doel daarvan is om compatibiliteit met een ander programma of systeem te bewerkstelligen.

Het goed lezen van de licentie wil nog wel eens interessante dingen aan het licht brengen. Zo las ik eens in de licentie van de Borland C/C++ compiler dat met die software geen software mag worden gebouwd die concurreert met een Borland software product. En in de licentie van de Oracle database manager stond (en wellicht staat) dat het niet is toegestaan de software te gebruiken voor het geven van Oracle cursussen, het publiceren van benchmarks of het maken van biologische, chemische en nucleaire wapens.

De meeste software leveranciers (en ook Open Source Software auteurs) hebben een “take it or leave it” attitude ten opzichte van de licenties. Of je gaat akkoord met de licentie, of je koopt c.q. gebruikt de software niet.

3.3 Licentiemanagement

Software is natuurlijk eenvoudig te kopiëren en her en der te installeren. Het is voor de copyright houders vrijwel onmogelijk om vast te stellen of de licenties worden nageleefd (en de verschuldigde licentiebedragen zijn voldaan). Illegaal gebruik van software is de commerciële copyright houders dan ook een doorn in het oog (hoofdzakelijk door het mislopen van de licentieinkomsten). Door allerlei technische, administratieve en strafrechtelijke maatregelen proberen de leveranciers de gebruikers te dwingen om het juiste aantal licenties aan te schaffen. Voorbeelden van dit soort maatregelen zijn:

- Software codenummers die bij installatie moeten worden ingevoerd.

- Registratie en activatie over het Internet.
- Hardware sleutels (dongles) die aanwezig moeten zijn bij uitvoering van het programma.
- LAN licentieservers voor het tellen van het aantal in gebruik zijnde licenties.
- Kliklijnen voor het rapporteren van illegaal software gebruik.

Alhoewel de leveranciers natuurlijk volledig in hun recht staan om te verlangen dat voor ieder gebruik conform de licentie wordt betaald is het bijhouden van hoeveel licenties waarvoor nu precies nodig zijn voor de meeste organisaties inmiddels een uiterst complexe en tijdrovende (en dus ook kostbare) zaak geworden. Ook het feit dat de meeste Closed Source Software met geheel eigen licenties komen maken het niet eenvoudig om het overzicht te houden hoeveel licenties nu precies aangeschaft dienen te worden.

3.4 Entrée: Open Source Software

Open Source Software is software waaraan de copyright houder een licentie heeft gehangen die in overeenstemming is met de Open Source Definition . Op Open Source Software zit dus “gewoon” copyright. De houder van dit copyright is negen van de tien keer de auteur van de software. De crux zit hem echter in de licentie! Aangezien de copyright houder mag bepalen wat er met een beschermd werk gebeurt mag de copyright houder ook beslissen dat het werk vrij mag worden gedistribueerd en aangepast. Het systeem van copyright wordt dus in de Open Source wereld gebruikt om het vrije karakter van de software te garanderen.

Een Open Source Definition compatible licentie bevat de volgende kenmerkende elementen:

- De software mag worden gekopieerd en weggegeven zonder dat daar aan de copyright houder een vergoeding voor verschuldigd is (i.e. de software is gratis).
- De source code van de software is vrij beschikbaar (wordt bijvoorbeeld meege-distribueerd of kan worden gedownload).
- Iedereen mag de source code aanpassen en een “afgeleid werk” maken.

Open Source Software auteurs geven dus nogal wat rechten ten aanzien van hun software weg aan de gebruikers: iedereen mag het weggeven, verkopen, op zijn site zetten, de source code bestuderen, aanpassingen maken, bugs oplossen en de software gebruiken als basis voor nieuwe software (een zogenaamd “afgeleid werk”).

3.4.1 Commerciële Open Source Software

Merk op dat Open Source Software dus mag worden verkocht zonder dat daar aan de copyright houder een vergoeding voor verschuldigd is. Dit lijkt wat tegenstrijdig maar die mogelijkheid ligt expliciet in de Open Source Definition besloten. Echter, omdat de

software overall en nergens gratis beschikbaar is moet er voor de gebruiker natuurlijk een goede reden zijn om voor diezelfde software geld neer te tellen.

Linux distributeurs verkopen bijvoorbeeld gratis software (het GNU/Linux besturings-systeem inclusief alle toeters en bellen) door daar iets extra's bij in te pakken zoals bijvoorbeeld een gebruikershandleiding, installatieondersteuning of specifieke Closed Source Software .

3.5 Open Source Software licenties

Er zijn nogal wat verschillende Open Source Software licenties in omloop. Deze zijn door verschillende instellingen, universiteiten, consortia en andere Open Source Software belanghebbenden in het leven geroepen om de beschikbaarheid van hun software te regelen. Deze licenties verschillen met name in of ze bijzondere rechten voor de copyright houder bevatten en in beperkingen ten aanzien van het weer Closed Source maken van de software. De twee bekendste Open Source Software licenties zijn de GNU GPL en de BSD licentie.

3.5.1 GNU GPL: Free software

De “Free Software Foundation”¹⁵ (FSF) is een instelling die het gebruik van “free software” stimuleert. De licentie die bij free software hoort is de “GNU General Public Licence”. De GPL¹⁶ is gedeeltelijk software licentie en gedeeltelijk een politiek manifest waarin wordt opgeroepen om software te delen en geen beperkingen in het gebruik opgelegd te krijgen (door de copyright houders).

De essentie van “free software” is dat wijzigingen aan “free software” automatisch ook weer “free software” zijn. Het is dus niet mogelijk om een “free software” programma te verbeteren en die verbeteringen voor jezelf te houden (i.e. weer Closed Source) te maken.

Er zijn nogal wat commerciële software leveranciers die dit “virale” aspect van “free software” gebruiken om angst te zaaien tegen gebruik ervan. In hun FUD (Fear, Uncertainty, Doubt) campagnes tegen Open Source Software doen zij voorkomen dat gebruik van Open Source Software inhoudt dat al je bedrijfsgeheimen automatisch ook “op straat komen te liggen”. Deze aantijgingen slaan werkelijk helemaal nergens op, maar zijn conform de “Wizard’s First Rule”¹⁷ toch nog verbazingwekkend vaak effectief.

De GNU GPL is onder andere van toepassing op de Linux kernel en op de Perl programmeertaal.

Het “free” in “free software” slaat niet op het feit dat de software gratis is maar betreft het “vrije” karakter ervan. “Free as in freedom, not as in beer”.

De GPL is de meest extreme Open Source licentie. Er is ook een “Lesser GPL” (LGPL)

¹⁵<http://www.fsf.org>

¹⁶<http://www.gnu.org/licenses/gpl.html>

¹⁷Uit de ‘Sword of Truth’ boeken van Terry Goodkind: ‘People will believe anything; the first step to believing something is wanting it to be true or being afraid of it. . .

die van toepassing is op “free software” bibliotheken van algemeen beschikbare routines Closed Source Software mag tegen een door de LGPL beschermde library aange-linkt worden zonder dat die software dan geheel Open Source hoeft te worden. Deze variant was nodig omdat het anders niet mogelijk zou zijn om Closed Source Software op een Open Source besturingssysteem als Linux te gebruiken.

Naast de GPL en de LGPL heeft de FSF ook een “Free Documentation License” welke van toepassing is op software documentatie.

3.5.2 BSD licentie: Copycenter

De “Berkeley Software Distribution” (BSD) licentie is wellicht de eenvoudigste Open Source licentie die er is. In het kort (en in de woorden van Jordan Hubbard, de president van het FreeBSD consortium) komt de BSD licentie neer op het volgende:

1. Hier is de software (inclusief source code).
2. Wil je alsjeblieft onze copyrights niet verwijderen?
3. Doe er verder mee wat je wilt.
4. Klaag ons niet aan als het omvalt.

Het is altijd grappig om advocaten de BSD licentie te zien lezen (met name als ze het papier omdraaien en op zoek gaan naar de kleine lettertjes :-).

De BSD licentie wordt onder andere gebruikt door de Apache Foundation en voor alle BSD operating systemen (FreeBSD, OpenBSD, NetBSD).

3.5.3 De Netscape Public License

De “Netscape Public License”¹⁸ (NPL) is een licentie die is ontwikkeld door Netscape voor de broncode van de Netscape Navigator (Internet browser) die in de loop van het jaar 2000 Open Source werd. Het is een Open Source licentie waarin bijzondere rechten zijn opgenomen voor Netscape (intussen onderdeel van America Online), met name het recht om aan de broncode (inclusief uitbreidingen) ooit nog eens onder een andere licentie te hangen. Dit maakt het in principe mogelijk dat Netscape (en alleen Netscape) op basis van de vrij beschikbare broncode nog eens een Closed Source Software product fabriceert (bijvoorbeeld als er weer geld te verdienen valt in de browser markt :-).

De Open Source gemeenschap staat in principe afwijzend tegenover dit soort licenties, maar conform het adagium “whoever writes the code gets to choose the license” (uitspraak van Linus Torvalds) valt er weinig tot niets aan te doen. De aldus vrijgemaakte Netscape broncode is opgegaan in het Mozilla¹⁹ project. De Mozilla browser is inmiddels een complete en functionele Internet browser die standaard onderdeel is van veel Linux distributies.

¹⁸<http://www.mozilla.org/MPL/NPL-1.1.html>

¹⁹<http://www.mozilla.org>

3.6 Public Domain en shareware

Veel mensen trekken automatisch de vergelijking tussen Open Source Software en “Public Domain” en “Shareware” software. Toch is dit niet terecht.

Public Domain is een (Angelsaksisch) concept waarbij op een (potentieel beschermd) werk geen copyright (meer) zit, bijvoorbeeld omdat de auteur afstand heeft gedaan van het copyright of omdat het copyright vervallen is (b.v. als de auteur van een boek lang overleden is). Aangezien op een werk in het publieke domein geen copyright zit is er niemand die licenties hoeft te verstrekken. Iedereen mag het werk dan kopiëren, aanpassen en verspreiden. Op public domain software zit dus per definitie geen licentie.

Shareware is een Closed Source concept waarbij de software eerst mag worden uit-geprobeerd. Nadat de proefperiode voorbij is dient een licentie te worden aangeschaft (meestal tegen een vrij laag bedrag) bij de copyright houder. Die licentie is meestal een typische commerciële licentie waarin openbaring van de source code, doorverkoop en aanpassing niet zijn toegestaan.

3.7 Semi Open Source

Onder druk van de populariteit van Open Source Software zagen een aantal software leveranciers zich genoodzaakt om ook stappen te zetten in de richting van het openbaar maken van de broncode van hun programmatuur. Het volledige Open Source maken van die broncode is voor hun natuurlijk niet mogelijk, maar ze willen toch hun klanten een aantal voordelen van het beschikbaar zijn van die source code niet onthouden.

Deze leveranciers hebben mogelijkheden (en licenties) ontwikkeld waarmee een beperkte opening van de broncode kan worden bewerkstelligd voor selecte klanten en onder zeer stricte voorwaarden. Voorbeelden hiervan zijn de “Sun Community Source Licensing”²⁰ (voor Java) en het Microsoft “Shared Source”²¹ initiatief.

Het “openbaar” maken van de source via deze initiatieven valt niet onder de Open Source definitie zoals de Open Source Initiative die heeft opgesteld. Meestal wordt de broncode slechts aan geselecteerde klanten en individuen ter beschikking gesteld die dan (met name in het Microsoft “Shared Source” programma) een geheimhoudingsverklaring moeten ondertekenen. Ook is het niet toegestaan om de broncode aan te passen, te distribueren of afgeleide werken te maken. Hierdoor vallen een groot aantal voordelen van Open Source Software weg.

²⁰<http://www.sun.com/software/communitysource>

²¹http://www.opensource.org/advocacy/shared_source.php

4 Voorbeelden van Open Source Software

Er zijn in de loop der jaren ongelofelijke hoeveelheden software geschreven die als Open Source Software aan de mensheid ter beschikking zijn gesteld. Intussen is voor vrijwel iedere algemene toepassing een Open Source invulling te vinden. In sommige gevallen is de Open Source variant voor een bepaald toepassingsgebied zelfs de wereldwijde marktleider!

De hoeveelheid Open Source Software is inmiddels te groot om op te noemen. Op moment van schrijven staan er bij Freshmeat²² 25.186 Open Source projecten geregistreerd. Die projecten beslaan vrijwel het gehele vlak van de software ontwikkeling: operating systemen, programmeertalen, office applicaties, multimedia software, Internet toepassingen, grafische applicaties, spelletjes, databases, wetenschappelijke software, noem maar op!

Het is intussen zonder meer mogelijk om een compleet (en bruikbaar!) systeem op te bouwen op basis van Open Source Software. Sterker nog, Shell (u weet wel, van die gele schelpen langs de weg) gebruikt in landen waar in het kader van exportrestricties geen Amerikaanse hard- en software aan mag worden verkocht (Libië en Iran e.d.) systemen (server, desktop, laptop) die volledig met Open Source Software zijn uitgerust!

Een korte doorsnede geven van een dergelijke hoeveelheid software is natuurlijk een bijna onmogelijke zaak. Om een goed overzicht te krijgen kan ik u aanraden om eens een uurtje over Freshmeat te browsen. Toch wil ik hier enkele hoogtepunten van vrij beschikbare software vernoemen.

4.1 GNU software

Het GNU (“GNU’s Not Unix”²³) project is één van de oudste Open Source projecten. Open Source coryfee Richard Stallman startte het project in 1984 met als doel om een volledig vrije variant van het Unix besturingssysteem te bouwen. Het GNU project startte met het herbouwen van de diverse tools die bij Unix horen: de C compiler (GCC), de C library (Glibc), de assembler (Gas) en alle andere “kleine” Unix commando’s (Yacc, lex, awk, ls, tar et cetera).

Met het verschijnen van de GNU Unix kernel (Hurd) is het systeem recentelijk gecompleteerd. Echter, de GNU tools zijn zo ontzettend goed dat ze in veel meer systemen worden gebruikt dan alleen het GNU systeem. De GNU toolset is inmiddels op vrijwel alle Unix en Windows systemen ter beschikking²⁴. De GNU tools worden bijvoorbeeld ook onder Linux en de meeste BSD systemen gebruikt.

Van origine richtte het GNU project zich hoofdzakelijk op operating system componenten zoals compilers, linkers en andere “technische componenten”. Recentelijk vinden we ook wat meer eindgebruikersachtige componenten in GNU zoals het “GNU Image Manipulation Program” (GIMP) en het “GNU Free Film Project” (virtuele filmstudio).

²²<http://www.freshmeat.net>

²³<http://www.gnu.org>

²⁴Red Hat/Cygnus heeft Cygwin ontwikkeld, een port van de GNU tools naar Windows.



Figuur 3: Richard Stallman als Sint Ignacius

4.2 Linux kernel



Figuur 4: Linus Torvalds

Het Linux besturingssysteem is zonder meer het vlaggeschip van de Open Source wereld. Linux is een compleet Unix compatible besturingssysteem dat geheel uit Open Source Software componenten bestaat.

De Finse informatica student Linus Torvalds startte de ontwikkeling van Linux in 1992. Hij maakte Linux vrij beschikbaar op het Internet onder de GNU General Public License (GPL, free software). De beschikbaarheid van Linux werd al vrij snel opgepikt door duizenden netizens (Internet bewoners), die zich vervolgens druk bezig gingen houden met het uitbreiden en aanvullen van Linux. Aldus groeide het Linux systeem in korte tijd uit tot een compleet Unix besturingssysteem, inclusief ondersteuning voor netwerken, grafische applicaties en multimedia toepassingen. Linux staat bekend om zijn uitstekende prestaties (zelfs op “low end” hardware), stabiliteit en flexibiliteit.

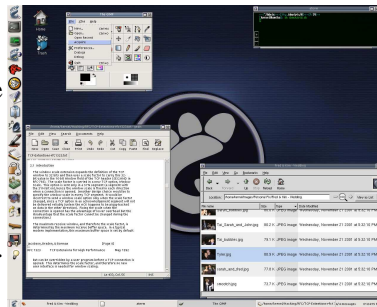
De naam “Linux” slaat in principe alleen op de kern van het besturingssysteem (de “kernel”). Een volledig Linux systeem bestaat uit de Open Source kernel aangevuld met andere (Open Source en soms ook Closed Source) componenten (zoals de GNU tool set). Om die reden staan diverse Open Source supporters erop om een compleet systeem GNU/Linux te noemen.

4.3 X Window System en aanverwante

Het “X Window System”²⁵ (kortweg: X) is de de-facto standaard voor grafische werkstations in de Unix wereld. Alle commerciële Unix leveranciers ondersteunen X en voor de Open Source Unix systemen is er het XFree86 project²⁶. X is initieel ontwikkeld door het Massachusetts Institute of Technology (MIT). Het copyright op X is sinds 1 januari 1997 in handen van de Open Group (voorheen de Open Software Foundation²⁷).

X verzorgt de volledige grafische (beeldscherm, toetsenbord, muis) interactie tussen applicatieprogramma’s en de gebruiker. Één van de meest krachtige eigenschappen van X is dat het een netwerkafhankelijk grafisch subsysteem is. Dit heeft tot gevolg dat de grafische programmatuur op andere computers in het netwerk kunnen draaien terwijl de interactie toch plaatsvindt op de desktop computer van de gebruiker!

Een volledige X systeem bestaat uit diverse onderdelen. De grafische “look and feel” van een X systeem wordt bepaald door de “Window Manager” component. Er zijn intussen tientallen Open Source window managers beschikbaar met de meest uiteenlopende (en soms uitzinnige) mogelijkheden. De bekendste van dit moment zijn KDE



Figuur 5: X desktop met de Gnome omgeving

²⁵<http://www.x.org>

²⁶<http://www.xfree86.org>

²⁷<http://www.osf.org>

(K Desktop Environment,²⁸) en GNOME⁽²⁹⁾). KDE en GNOME bieden een volledige grafische gebruikersomgeving inclusief file managers, browsers, grafische instellingspanelen, virtuele desktops en vele tientallen productiviteitsapplicaties (zoals “MineSweeper” :-).

Naast de “native” X implementaties onder Unix zijn er intussen ook X Window System clients en servers ontwikkeld die onder Windows en MacOS/X draaien. Hierdoor ontstaat de mogelijkheid om grafische client software op Unix systemen te draaien en de grafische interacties op Windows systemen te laten plaatsvinden.

4.4 BSD

Door alle commotie rondom Linux wordt nog wel eens vergeten dat er meer vrij beschikbare Unix systemen zijn op de aardbol. De belangrijkste groep (naast Linux) zijn de Berkeley Software Distribution (BSD) systemen. Deze groep besturingssystemen is afgeleid van de originele Unix implementaties van de universiteit van Berkeley (Californië).

De bekendste BSD implementaties zijn FreeBSD³⁰ en OpenBSD³¹. FreeBSD richt zich hoofdzakelijk op server stabiliteit terwijl OpenBSD beveiliging hoog in het vaandel heeft staan (waarmee niet is gezegd dat FreeBSD niet veilig en OpenBSD niet stabiel is :-).

De BSD systemen hebben een lange historie van uitmuntendheid op het gebied van performance en stabiliteit. Het ontwikkelingsmodel rondom de BSD's is wat anders dan dat bij bijvoorbeeld Linux. Hierdoor zijn de BSD's meestal iets conservatiever waar het nieuwe mogelijkheden aangaat, maar hebben ze ook minder te leiden van “bleeding edge” verschijnselen. 's Werelds grootste FTP server (ftp.cdrom.com) is een FreeBSD server en OpenBSD is volgens velen het meest veilige Internet server systeem van dit moment.

4.5 Samba

Samba³² is een Windows compatible file en print server applicatie voor Unix. Met Samba is het mogelijk om Windows clients te laten verbinden met servers op Unix (ook Linux en BSD) systemen. Voor de gebruikers is het vaak volledig transparant dat ze niet worden bediend door Windows NT (2000) servers maar door Unix systemen.

4.6 Apache

Apache³³ is een vrij beschikbare Internet web server voor Windows en Unix server systemen. Net als alle andere Open Source Software wordt Apache ontwikkeld en onderhouden door duizenden professionals die wereldwijd samenwerken via het Internet.

²⁸<http://www.kde.org>

²⁹<http://www.gnome.org>

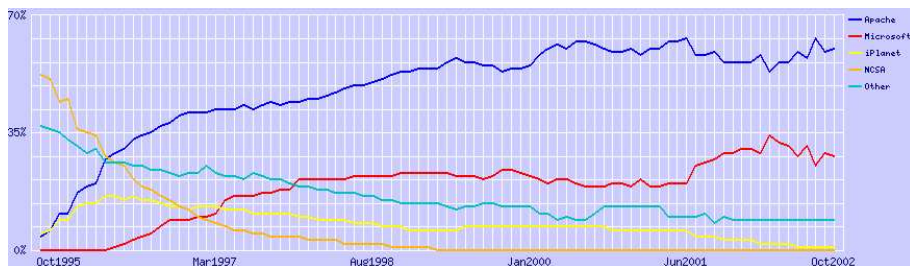
³⁰<http://www.freebsd.org>

³¹<http://www.openbsd.org>

³²<http://www.samba.org>

³³<http://www.apache.org>

Op moment van schrijven gebruikt ongeveer 60% van alle Internet servers de Apache web server (bron: Netcraft³⁴):

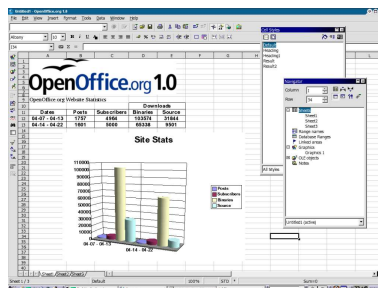


Naast de Apache web server ontwikkelt de Apache Foundation nog meer interessante web-gerelateerde software waaronder Tomcat (Java servlet en JSP server), Xerces (XML parser) en Xalan (XSL stylesheet processor).

4.7 OpenOffice

OpenOffice³⁵ is een “all singing all dancing” office suite met daarin geïntegreerd een tekstverwerker, een spreadsheet, een presentatieprogramma, een mail pakket en wat dies meer zij. OpenOffice is een tamelijk complete applicatie die zich qua functionaliteit goed kan meten met andere (Closed Source) oplossingen op dat gebied.

De oorsprong van OpenOffice ligt in het (Closed Source) StarOffice pakket, dat later door Sun Microsystems is overgenomen en door hen Open Source is gemaakt.

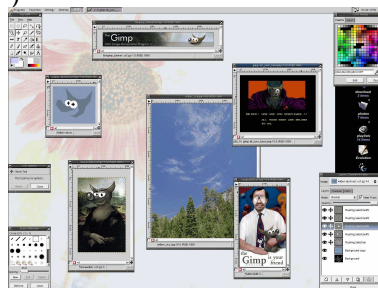


Figuur 6: OpenOffice screenshot

4.8 GIMP

Het “GNU Image Manipulation Program” (GIMP³⁶) is een Open Source pakket voor het manipuleren van foto’s, plaatjes en andere afbeeldingen.

Het sterke van de GIMP is dat het een modulaair systeem is wat via plug-ins kan worden uitgebreid. Voor allerlei spannende grafische effecten zijn intussen dergelijke plug-ins ontwikkeld. Het zelf uitbreiden van de GIMP (of een GIMP plug-in schrijven) is natuurlijk ook mogelijk (want: Open Source :-). GIMP is een uitermatig krachtig en (relatief) gebruikersvriendelijk pakket waarmee zowel eenvoudige als gecompliceerde grafische bewerkingen kunnen worden bewerkstelligd.



Figuur 7: GIMP screenshot

³⁴<http://www.netcraft.com>

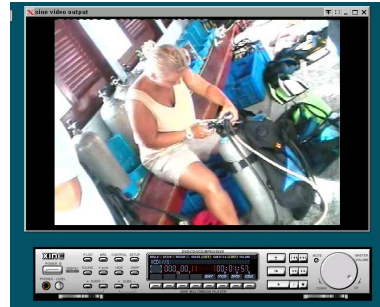
³⁵<http://www.openoffice.org>

³⁶<http://www.gimp.org>

4.9 XINE

Er zijn diverse Open Source oplossingen voor het manipuleren en weergeven van allerlei multimedia formaten. XINE³⁷ is in staat om diverse video en audio formaten (CD, DVD, Video CD, DivX) af te spelen.

Het weergeven van DVD's stuit nog wel eens op wat problemen in verband met de op commerciële DVD's toegepaste encryptie van de video streams. Het schijnt echter dat er op het Internet XINE plug-ins zijn te vinden waarmee die beperking kan worden omzeild.



Figuur 8: XINE speelt een Video CD

4.10 XMMS

Het “X MultiMedia System” (XMMS³⁸) is een applicatie voor het afspelen van geluidsbestanden zoals WAV, MP3 en OGG/Vorbis files. XMMS startte als een Open Source variant van het overbekende “winamp”.

XMMS heeft vrijwel precies dezelfde features als “winamp”, zoals ingebouwde equalizer, configureerbare “playlist”, afspelen van MP3 streams van het Internet (bijvoorbeeld Internet radio³⁹, “skins” (instelbaar uiterlijk voor de XMMS windows), spectrum analyzer en oscilloscoop.



Figuur 9: XMMS screenshot

4.11 PostgreSQL

PostgreSQL⁴⁰ is een Open Source object-relacionele database met volledige ondersteuning voor SQL, transacties, “user defined types” en andere geavanceerde database constructies. De oorsprong van PostgreSQL ligt in Ingres en de “cutting edge” database research aan de universiteit van Berkeley. De PostgreSQL server wordt ondersteund op Unix en Windows servers.

4.12 Perl

Perl⁴¹ is een vrij beschikbare programmeertaal die het beste van script taal en C in zich herbergt. Met Perl kunnen snel en relatief eenvoudig allerlei programma's worden ontwikkeld, variërend van scriptjes om bestanden te manipuleren tot en met grafische applicaties en netwerk client/server software. Door de modulaire opbouw van Perl kunnen krachtige features worden toegevoegd via externe modules. Het Consolidated Perl Achive Network (CPAN⁴²) is een verzameling sites waar duizenden Perl modules

³⁷<http://www.xine.org>

³⁸<http://www.xmms.org>

³⁹<http://www.shoutcast.com>

⁴⁰<http://www.postgresql.org>

⁴¹<http://www.perl.org>

⁴²<http://www.cpan.org>

zijn te vinden.

Veel Open Source Software is zelf in Perl ontwikkeld. Een goed voorbeeld daarvan is SpamAssassin⁴³, een Open Source pakket voor het filteren van ongewenste reclame email. Perl is momenteel ook de meest gebruikte taal voor het ontwikkelen van web server applicaties via de CGI methode.

⁴³<http://www.spamassassin.org>

5 De organisatie van de Open Source gemeenschap

Open Source Software wordt typisch ontwikkeld in projectvorm gecentreerd rondom een web site en een gedeelde source code repository (die vaak wordt bijgehouden met het eveneens Open Source CVS⁴⁴ (Concurrent Versioning System) versiebeheersysteem. De leden van het project communiceren met elkaar via het Internet (email, mailing lists, web sites, IRC) en ontwikkelen uitbreidingen, lossen bugs op en geven nieuwe versies vrij.

Meestal is er een groep “core developers” die al lange tijd bij het project zijn betrokken en die diepgaande kennis hebben van het onderhavige probleemgebied. Deze harde kern beslist meestal welke kant het opgaat met de ontwikkeling en welke verbeteringen worden geaccepteerd in de centrale source base en welke niet (b.v. indien het een feature betreft die te exotisch om in de “main stream” te worden opgenomen). De taakverdeling tussen die ontwikkelaars wordt meestal informeel vastgesteld. Wie ergens interesse in (en verstand van) heeft belast zichzelf meestal met een bepaald deelgebied van het project. Aangezien het geheel een vrijwillige inspanning is zijn de meesten al lang blij dat iemand zich bereid heeft verklaard om aan het project bij te dragen.

In de periferie van het project zitten vaak nog een aanzienlijk aantal mensen die vanaf de zijlijn meedoen. Zij dragen doorgaans slechts sporadisch bij maar houden “een oogje in het zeil”. Deze groep bestaat meestal uit (gevorderde) gebruikers van de software die bij willen houden hoe het ermee gaat en of het al tijd is om nieuwe versies te installeren en/of input te leveren.

Via een systeem van meritocratie promoveren ontwikkelaars in de informele hiërarchie van buitenstaander tot insider. Wie regelmatig goede bijdragen levert promoveert meestal automatisch tot lid van de kern, en wie langere tijd niets meer van zich laat horen verdwijnt meestal vanzelf uit beeld.

Nieuwe gebruikers worden vaak via een soort “self help” groepsmechanisme op pad geholpen. Vragen (via de mailing lists) worden bijvoorbeeld door andere gebruikers/ontwikkelaars bantwoordt en de nieuwe gebruikers worden geassisteerd bij het implementeren van de software. Het credo hierbij is wel dat men vooral help wie zichzelf helpt. Als mensen (b.v. uit luiheid) vragen stellen die eenvoudig in de documentatie zijn terug te vinden dan wordt in niet mis te verstane termen uitgelegd dat enige zelfinspanning op prijs wordt gesteld alvorens de goegemeente met je onkunde te belasten. De “How to ask questions the smart way” FAQ⁴⁵ geeft uitgebreide richtlijnen over hoe slimme vragen te stellen (en domme vragen te voorkomen :-).

Niet zelden ontstaat er in een Open Source project een stammenstrijd tussen partijen die in conflict zijn over hoe het verder moet met de software (Open Source ontwikkelaars zijn ook net mensen :-). Indien er geen middenweg of consensus te vinden is ontstaat er een “fork”. Aangezien we het over Open Source Software hebben mag iedereen de source code meenemen en wijzigingen aanbrengen. Niemand kan dus tegenhouden dat een subgroepje besluit om de source code op te pakken en een alternatieve versie te maken (met een eigen web site en dergelijke). Het is dan aan de gebruikers om te beslissen welke versie ze willen installeren. In het Darwiniaanse evolutiemodel van Open Source Software zal de tijd dan uit moeten wijzen welke variant overleeft (of dat er wellicht voor beiden plaats is). Een “fork” wordt in het algemeen gezien als een

⁴⁴<http://www.cvshome.org>

⁴⁵<http://www.tuxedo.org/esr/faqs/smart-questions.html>

slechte ontwikkeling aangezien het ontwikkelaarscapaciteit versplintert en verwarring zaait bij de gebruikers. Aangezien de vrijheid om een “fork” te maken echter in de Open Source definitie is besloten doen Open Source ontwikkelaars er alles aan om dit fenomeen te voorkomen.

5.1 De Open Source ontwikkelmethodologie

De Open Source manier van het ontwikkelen van software is zo succesvol gebleken dat er bedrijven zijn die die aanpak intern ook gebruiken. Grote bedrijven hebben vaak vele honderden (zo niet duizenden) ontwikkelaars, gebruikers en andere “stake holders” die betrokken zijn bij de ontwikkeling en implementatie van de software (hetzij intern, hetzij extern). Het los-vaste organisatieproces van de Open Source gemeenschap is in veel gevallen ook de optimale manier om het software ontwikkelingsproces binnen een organisatie vorm te geven. Door de toepassing van Internet technologie als web servers, mailing lists en CVS kan een zeer laagdrempelige omgeving worden geschapen via welke alle personeelsleden betrokken kunnen raken bij de software ontwikkeling en daaraan bij kunnen dragen (natuurlijk voor zover ze daar gezien hun functie de mogelijkheid en vrijheid toe hebben).

Het Amerikaanse bedrijf CollabNet⁴⁶ specialiseert zich (onder andere) in de implementatie van dit soort processen bij klanten. CollabNet is mede opgericht door Brian Behlendorf, de president en één van de grondleggers van de Apache Foundation.

5.2 Wie zijn de Open Source ontwikkelaars?

Een wijd verbreide misvatting is dat Open Source Software wordt ontwikkeld door langharige hobbyisten van zestien jaar oud op zolderkamertjes onder invloed van teveel pizza en cola. Niets is echter minder waar. Alhoewel er gelukkig ook een groot aantal enthousiastelingen van de bovenbeschreven soort in de Open Source gemeenschap rondhobbelen wordt de meeste Open Source Software ontwikkeld door professionele software engineers, hetzij in hun vrije tijd, hetzij als onderdeel van hun functie.

Dat laatste schijnt altijd een beetje ongelofelijk toe, maar als je er even over nadenkt dan valt het best wel mee. Een hard disk leverancier wil er bijvoorbeeld best wel wat inspanning tegenaan gooien om er voor te zorgen dat zijn hard disks goed worden ondersteund door Open Source besturingssystemen als Linux en FreeBSD. Om dezelfde reden zijn grote .com bedrijven (zoals Yahoo en Amazon) best genegen om bugs in Apache op te lossen en bij te dragen aan de ontwikkeling van die software in de voor hun gewenste richting. Ook consultants van bedrijven als Open Solution Providers zijn nooit te beroerd om als ze in het kader van een opdracht een fout in een Open Source component oplossen of een kleine uitbreiding realiseren om die terug te doneren aan het project om er zodoende zorg voor te dragen dat volgende versies van die software de voor die klant van belang zijnde functionaliteit biedt.

⁴⁶<http://www.collab.net>

6 Kwaliteit van Open Source Software

Er is nogal wat te doen over de kwaliteit van Open Source Software . Alhoewel er natuurlijk bij Open Source Software (net als bij Closed Source) goede en slechte software is, is de kwaliteit van de meeste software hoog tot zeer hoog. Daarvoor zijn een aantal oorzaken aan te voeren:

1. Open Source Software wordt niet geteisterd door release schema's en deadlines. De ontwikkelaars knutselen gewoon net zo lang door tot ze er tevreden mee zijn, niet gehinderd door oneigenlijke druk.
2. Aangezien de source code van Open Source Software vrij beschikbaar is kan iedereen op de wereld de kwaliteit van de software beoordelen. Een ontwikkelaar gaat dus in principe volledig met zijn billen bloot ten aanzien van zijn/haar deskundigheid. Een aangezien niemand natuurlijk zichzelf als prutser ten toon wil stellen wordt er doorgaans alleen maar software vrijgegeven waarmee je voor de dag kan komen.
3. Het proces van het vinden en oplossen van fouten is volledig transparant. Een bekend Open Source adagium is "Many eyeballs make every problem shallow". Er zijn niet zoveel mensen die een device driver kunnen ontwikkelen maar een (triviale) fout opsporen en verbeteren is een veel grotere groep gegeven. Daarbij is het terugrapporteren van de fout en de opname daarvan in de centrale source repository veel eenvoudiger. Bij veel Open Source projecten kan iedere dag de allernieuwste status van de software worden gedownload (vergelijk dit met de patch/service pack release schedules van commerciële software leveranciers).
4. Open Source projecten hebben niet/nauwelijks te lijden van marketing stress. De ontwikkelaars beslissen zelf waar ze hun tijd aan besteden, zonder te worden gehinderd door marketeers die de nadruk leggen op features die goed verkopen of die voor één bepaalde klant belangrijk zijn.
5. Open Source ontwikkelaars zijn veel eerlijker over welke features nog niet/nauwelijks werken of wat de randvoorwaarden zijn. Het volgende stukje commentaar is bijvoorbeeld te vinden in de source code van de Wavelan wireless network card driver in de Linux kernel:

```
/* Another glitch : Due to the way the GET_PACKET macro is written,  
 * we are not sure to have the same thing in skb->data. On the other  
 * hand, skb->mac.raw is not defined everywhere...  
 * For versions between 1.2.13 and those where skb->mac.raw  
 * appear,  
 * I don't have a clue...
```

Dit soort eerlijkheid helpt bij het analyseren van problemen en werkt dus kwaliteitsverhogend.

6.1 Veiligheid van Open Source Software

In het verlengde van kwaliteit wordt ook vaak gesproken over de veiligheids- en beveiligingsaspecten van Open Source Software . In zijn algemeenheid kan worden gesteld

dat Open Source Software veiliger is dan Closed Source Software . De reden hiervoor is dat de source code van Open Source Software algemeen beschikbaar is en dat beveiligingsproblemen veel eerder worden gevonden en opgelost dan in Closed Source Software .

Een veel geciteerde misvatting is dat door de beschikbaarheid van de source code kwaadwillende inbrekers ook veel sneller op de hoogte zijn van gaten in het systeem en dus sneller achter beveiligingsproblemen komen dan bij Closed Source Software . Je moet daarbij echter bedenken dat het verbeteren van beveiliging niet te maken heeft met het geheim houden van beveiligingsproblemen maar met het oplossen daarvan. Het efficiënte bug tracking en oplossingsproces van Open Source Software leidt aantoonbaar tot een veel kortere cyclus van het vinden en oplossen van beveiligingsproblemen.

Er zijn helaas veel te veel voorbeelden van dat leveranciers van Closed Source Software treuzelden met het erkennen en oplossen van beveiligingsproblemen in hun programmatuur. Open Source Software projecten leveren vaak in recordtijden na het bekend worden van de bug oplossingen (meestal binnen 24 uur).

's Werelds beste encryptie algoritmes zijn niet voor niets ook Open Source . Alleen de doorlopende "peer review" door betrokken partijen kan de veiligheid van software garanderen.

7 Waarom ontwikkelt men Open Source Software ?

Één van de grootste vraagstukken waarmee mensen die pas met Open Source Software in aanraking zijn gekomen worstelen is waarom men in vredesnaam tijd en moeite zouden stoppen in het (door)ontwikkelen van iets om dat vervolgens gratis weg te geven. Men vermoed veelal charitatieve of andere redenen van wereldverbeterende aard. 's Werelds meest onvermoeibare free software missionaris Richard Stallman (van GNU en de Free Software Foundation) wordt ook niet voor niets door zijn tegenstanders uitgemaakt voor communist en/of socialist⁴⁷.

Bij nadere beschouwing blijken er echter diverse sociaal en economisch verantwoorde redenen te zijn om Open Source Software te ontwikkelen of te onderhouden.

7.1 Therapeutische redenen

Het is ontegenzeggelijk waar dat behoorlijk wat Open Source Software (tenminste initieel) is ontwikkeld om “therapeutische redenen⁴⁸”. Dat gedeelte van de wereldbevolking dat de kunst van het programmeren niet machtig is kan zich waarschijnlijk de therapeutische waarde van software ontwikkeling moeilijk indenken. Edoch, waar sommigen zich ontspannen met het bijknippen van bonsai boompjes en anderen sigarenbandjes verzamelen is er voor veel leden van het IT-gilde weinig dat zo stressverminderend werkt als het directe één-op-één duel tussen het eigen IQ en de CPU.

Zo ben ik zelf ooit eens (in 1992) om therapeutische redenen begonnen met de ontwikkeling van een Open Source interpreter voor de Comal programmeertaal⁴⁹. Deze is inmiddels op het Internet beschikbaar en één van mijn (vele :-) vrijetijdsbestedingen is het onderhouden en uitbreiden ervan.

7.2 Scratching an itch...

Één van de bekendste redenen om Open Source Software te ontwikkelen is het oplossen van een opspelend probleem (“scratching an itch”; jeuk is erger dan pijn). Veel Open Source Software is primair ontwikkeld om iemands eigen probleem op te lossen. Linux is in eerste instantie ontwikkeld omdat Linus Torvalds meerdere terminal emulatoren tegelijk wilde kunnen draaien en Larry Wal ontwikkelde Perl omdat hij een goede programmeertaal miste om een bepaald soort problemen mee op te lossen. Sendmail is bijvoorbeeld ontstaan omdat Eric Allman gek werd van de mail routeringsproblemen tussen verschillende soorten mail netwerken.

Het komt echter zelden voor dat je de enige bent op de planeet met een bepaald probleem. Een goede oplossing voor zo'n probleem is dan ook voor veel meer mensen interessant. Veel programmeurs hebben helemaal niet de mogelijkheid of interesse om een software bedrijfje te beginnen (nog los van de vraag of iemand geld voor je (halve) oplossing over zou hebben). Door de software gratis weg te geven verminder je de bruikbaarheid van die software voor jezelf niet en heb je grote kans dat anderen verbeteringen toevoegen waar je dan zelf weer gratis de beschikking over krijgt.

⁴⁷Wat in de Verenigde Staten zo ongeveer de ergste scheldwoorden zijn die de taal rijk is :-)

⁴⁸Ook wel: ‘hobby’, maar dat klinkt toch weer minder professioneel :-)

⁴⁹<http://www.josvisser.nl/opencomal>

Zo heb ik ooit eens een module ontwikkeld om de WU FTP server software te koppelen aan een LDAP server voor user authenticatie (probleem speelde bij een klant). Die module was tamelijk eenvoudig maar voldeed precies aan mijn wensen. Na publicatie van de software op het Internet heb ik van diverse mensen verbeteringen toegezonden gekregen waar ik dan op mijn beurt weer gebruik van kan maken (ook in de OpenComal interpreter zitten intussen routines die door derden zijn gedoneerd).

De Open Source beweging is wat dat betreft een soort “real life” variant van het “prisoner’s dilemma”: als iedereen meedoet dan wint iedereen. Als men “vals speelt” dan verliest iedereen. . .

7.3 Hardware leveranciers

Hardware leveranciers moeten vaak device drivers ontwikkelen om hun apparatuur te laten samenwerken met diverse besturingssystemen. Voor die leveranciers is de ontwikkeling van die software een belasting. Het is natuurlijk glashelder dat niemand die software los koopt; de software is puur een “enabler” voor het gebruiken van de hardware. Door de software Open Source te maken verliezen ze in principe geen marktaandeel en verkrijgen ze wel de voordelen van Open Source .

Ik heb in een separaat artikel⁵⁰ wel eens betoogd dat Unix leveranciers als Hewlett-Packard en IBM er in principe goed aan zouden doen om hun bedrijfseigen Unix versies (HP-UX en AIX) los te laten en over te stappen op Linux. De redenering is ongeveer gelijk: niemand koopt HP-UX “los”; het gaat altijd in samenhang met de verkoop van een HP9000 RISC server of workstation. HP is echter wel volledig verantwoordelijk voor de uitbreiding en onderhoud van HP-UX en alle aanverwante componenten: niemand koopt immers een HP9000 zonder dat daar een besturingssysteem voor beschikbaar is! Door over te stappen op Linux zou HP haar software ontwikkelingskosten aanzienlijk kunnen reduceren terwijl ze toch hun klanten een volledig functioneel besturingssysteem kunnen bieden.

Een aardig voorbeeld van wat ik hier bedoel is dat Sun Microsystems voornemens is om de Open Source GNOME desktop mee te leveren met Solaris. Ik vermoed dat we steeds meer van dit soort leveranciersacties zullen gaan zien.

7.4 Voor gebruikers

Gebruikers van Open Source Software hebben de mogelijkheid om zelf de software aan te passen, bijvoorbeeld voor het oplossen van fouten of het maken van uitbreidingen. In veel gevallen zijn gebruikersorganisaties daar helemaal niet op ingericht, en moeten ze dat dan ook vooral niet doen.

Veel grotere bedrijven hebben echter grote IT-afdelingen waar genoeg expertise rondloopt om triviaal onderhoud zelf te plegen. Zo hebben OSP consultants op een project bij een grote bank bijvoorbeeld enkele cruciale verbeteringen in Apache aangebracht en die weer teruggeschikt naar de Apache foundation. Door deze laatste actie kunnen we (c.q. de bank) er zeker van zijn dat toekomstige versie van Apache nog steeds doen wat we willen. Verder blijkt het zelf doen in de meeste gevallen factoren sneller en goedkoper dan de interactie met de supportorganisatie van een software leverancier.

⁵⁰<http://e-zine.nluug.nl/hold.html?cid=99>

Je hoort nogal eens dat door dat soort verbeteringen terug te doneren aan de Open Source gemeenschap een organisatie in principe ook haar concurrenten een handje helpt. Dat is inderdaad waar, maar ook hier gaat (weer) op dat vooruitgang het handigst kan worden bewerkstelligd door zelf vooruit te hollen en niet door de tegenstanders tegen te houden. Daarnaast geldt die anti-concurrentieredenering natuurlijk ook voor het aanmelden van bugs bij software leveranciers.

7.5 Strategische overwegingen

Sommige organisaties zijn zo afhankelijk van Open Source Software dat ze significante inspanning leveren om die software uit te breiden en te onderhouden. Dit geldt bijvoorbeeld voor .com bedrijven die veel geld investeren aan het bijhouden van de cruciale web server software waar hun organisatie van afhangt. Organisaties als UUNET en America Online (Internet providers) gebruiken veel Open Source Software en doen dus goed mee in de Open Source gemeenschap om er voor te zorgen dat de software en de protocollen “de juiste kant op bewegen” en goed functioneren.

Het volledig in eigen beheer ontwikkelen en onderhouden van de software is meestal niet haalbaar. Voor dit soort organisaties biedt gebruik van Open Source Software de optimale combinatie van kosten en controle.

7.6 Open Source als business model

Er worden momenteel manmoedige pogingen gedaan door diverse bedrijven om een valide business model rondom Open Source Software op te zetten. De achterliggende gedachte is meestal dat er wellicht meer geld te verdienen is met service en support dan met het verkopen van software licenties. Door de software Open Source te maken wordt een potentieel grote klantenkring opgebouwd waaraan diensten kunnen worden gesloten.

Goede voorbeelden hiervan zijn een groot aantal Linux distributeurs en de consultancy groep rondom OpenNMS (Open Network Management System⁵¹). Eerlijk gezegd is nog niet geheel duidelijk of dit een valide business model is of niet. Er zijn in de Internet/Open Source hype van een paar jaar geleden flink wat bedrijven (de meeste eerlijk gezegd gebaseerd op ronduit belachelijke ideeën) op deze basis over de kop gegaan, maar een aantal bestaan nog steeds en doen steeds betere zaken.

⁵¹<http://www.opennms.org>

8 Waarom Open Source Software gebruiken?

De hamvraag van dit epistel is natuurlijk: “Waarom zouden wij Open Source Software moeten gebruiken?” Het antwoord is wat mij betreft verbluffend eenvoudig: “Als de gewenste functionaliteit met Open Source Software tegen lagere kosten kan worden gerealiseerd”.

Op zich is de “Total Cost of Ownership” (TCO) discussie altijd een beetje gevaarlijke want er zijn een groot aantal manieren om die vergelijkingen te manipuleren. Toch is TCO de enige goede manier om een gefundeerde beslissing te nemen; de kunst is alleen om de juiste inschatting te maken.

Het is allereerst belangrijk om in te zien dat het niet nodig is om een complete voor/tegen beslissing te nemen. Op ieder punt in de IT infrastructuur kan/moet worden gekeken welke oplossing aldaar de meest geëigende is. Dit leidt meestal (zoals bijvoorbeeld op het OSP-kantoor) tot een gemengde Open Source /Closed Source omgeving.

8.1 Licentiekosten

Open Source Software is gratis. Dit betekent:

1. Geen licentiekosten.
2. Geen kosten om bij te houden hoeveel licenties er in omloop zijn.
3. Geen problemen door verlopen van licentiesleutels of andere gerelateerde ongen.

Het “gratis” aspect van Open Source Software moet echter niet worden overdreven. In de gemiddelde IT-omgeving vormen de licentiekosten slechts een fractie van de TCO.

8.2 Kwaliteit

In veel (maar ook weer niet alle) gevallen is de kwaliteit van Open Source Software hoger. Die hogere kwaliteit uit zich dan onder andere in betere stabiliteit, betere beveiliging, hogere performance en minder bugs/problemen. Het kwantificeren van die voordelen valt niet mee maar zou in principe moeten leiden tot minder menselijke inspanning benodigd voor installatie, implementatie en trouble shooting.

8.3 Makkelijker verkrijgbaar

In veel organisaties is het langs officiële weg verkrijgen van software een nachtmerrieachtige lijdensweg met afdelingen inkoop, contractonderhandelingen, budgetgoedkeuringen, standaardproductlijsten en aanverwante verschijnselen. Open Source Software kan gewoonweg worden gedownload en “Klaar is Cornelis”. Met name voor researchachtige toepassingen, prototyping en “proof of concept” projecten is dat een niet te versmaden voordeel van Open Source Software .

8.4 Betere beheersing van de oplossing

Een ander groot voordeel van het gebruik van Open Source Software is dat de gebruiker veel meer beheersing (Engels: “control”) heeft over de uiteindelijke oplossing. Door het Open Source karakter heeft de gebruiker namelijk veel beter inzicht in het gedrag en de (on)mogelijkheden van de software.

Idealiter zouden alle eigenschappen van de software natuurlijk uit de documentatie te halen moeten zijn. Echter, ik ben nog nooit een pakket tegengekomen met de perfecte documentatie. Documentatie is meestal incompleet, niet gedetailleerd genoeg (met name in hoe de software zich gedraagt in uitzonderingsgevallen) en loopt niet zelden achter. Bij twijfelgevallen is de broncode natuurlijk de ultieme documentatie!

In het meeste extreme geval heeft deze betere beheersing tot gevolg dat kritieke problemen of tekortkomingen (provisorisch) zelf in de source kunnen worden verholpen! Deze mogelijkheid verkleint de omlooptijd van probleem tot en met oplossing aanzienlijk!

9 Aandachtspunten bij gebruik van Open Source Software

9.1 Leverancierondersteuning

Het meest gehoorde argument voor Closed Source Software en tegen Open Source Software is de afwezigheid van leverancierondersteuning voor Open Source oplossingen. Het is ontegenzeggelijk waar dat bij Open Source Software er meestal geen leverancier is waar een support contract kan worden afgesloten en die in geval van nood kan worden gebeld. De Open Source gemeenschap is wat dat betreft geen vervanger voor de leverancier, omdat de vrijwilligers van de Open Source projecten geen garantie geven dat op je vragen een antwoord zal komen. Je hebt immers geen recht op een antwoord!

Veel organisaties staan daarom huiverig tegenover Open Source Software . Zij weten dat ze niet de expertise in dienst hebben om zelf aan de (vaak complexe en omvangrijke) broncode te knutselen en vinden daarnaast (terecht) dat dat niet hun (primaire) business process is.

Toch is de situatie niet zo grimmig als lijkt: Allereerst kunnen op Open Source oplossingen wel degelijk support contracten worden afgesloten. Diverse grote en kleine ondernemingen verkopen ondersteuning voor Open Source Software met daarin elementen als de vraagbaak, ondersteuning bij problemen en advisering over beveiligingsproblemen en nieuwe versies.

Open Source Software trekt het speelveld wat dat betreft gelijk voor alle spelers: iedereen heeft de beschikking over de broncode en iedereen kan daardoor effectief ondersteuning verkopen. Hierdoor kunnen leveranciers van support contracten gaan concurreren en die concurrentie levert betere support tegen een lagere prijs. Daarnaast staat dit model toe dat de klant na verloop van het support contract het contract niet verlengt maar met een andere aanbieder in zee gaat zonder dat dat tot gevolg heeft dat de software moet worden vervangen! Dit is een heel groot voordeel van Open Source ten opzichte van Closed Source !

Daarbij ben ik ook van mening dat de waarde van de support contracten op Closed Source Software door de leverancier daarvan schromelijk wordt overdreven. Vrijwel iedere automatiseerder heeft de nodige “sterke verhalen” (edoch helaas meestal op waarheid berustend) op zak omtrent de ineffectiviteit en traagheid van de helpdesk van leveranciers...

9.2 Externe consultants

Het open karakter van Open Source Software zorgt ervoor dat externe consultants vaak vele malen effectiever voor hun klant kunnen optreden. Consultants van bedrijven als OSP hebben vaak wel de expertise om de broncode van (Open Source) programmatuur te doorgronden en aan te passen. Dit heeft tot gevolg dat wij bij voorkomende problemen niet met een leverancier in de slag hoeven maar meteen tot de kern van de zaak kunnen doordringen. Voor de klant heeft dit tot gevolg dat problemen sneller worden verholpen en implementaties sneller verlopen. Gezien het “te huur per uur” karakter van de meeste consultancy opdrachten betekent dit een aanzienlijke besparing.

9.3 Feature shoot-out

Één van de meest frustrerende verschijnselen in de IT is de “feature shoot-out”, het vergelijken van features van software producten om (mede) te kunnen bepalen welke oplossing de voorkeur verdient. Productselecties op basis van matrices met gewichten suggereren mijns inziens vaak een objectiviteit die slechts zelden aanwezig is (wie heeft er niet meegemaakt dat de gewichten achteraf werden aangepast omdat de “verkeerde” oplossing uit de matrix kwam :-).

Voor Open Source adepten zijn “feature shoot-outs” des te frustrerender vanwege het feit dat Open Source Software “feature wise” anders in elkaar zit. Closed Source Software bevat doorgaans de features die **verkopen** terwijl Open Source Software doorgaans alleen maar de features bevat die we **gebruiken** (c.q. nodig hebben). Doordat software meestal niet door de functionarissen wordt aangeschaft die het implementeren, onderhouden en gebruiken is dit nogal een verschil. Open Source Software lijkt vaak minder geavanceerd door het ontbreken van features die (naar wij in de praktijk weten) eigenlijk nauwelijks worden gebruikt of per saldo van minder waarde zijn dan initieel toeschijnt.

10 Conclusie

Q: Is Open Source Software toepasbaar in de IT-organisatie van een professionele organisatie?

A: Kort door de bocht: Ja!

A: Iets langer door de bocht: Er is uitermate veel uitstekende Open Source Software die zonder meer kan worden gebruikt in uw IT-organisatie. Daarvan mogen natuurlijk geen wonderen worden verwacht. Open Source Software moet daar worden toegepast waar het de gewenste functionaliteit tegen de laagste kosten kan bewerkstelligen.

Open Source Software biedt veel voordelen vanwege de gemiddeld hoge kwaliteit (stabiliteit, beveiliging, functionaliteit), de flexibiliteit en de lage (of afwezige) aanschafkosten. Daarentegen moet de organisatie natuurlijk ook worden ingericht op het gebruik van Open Source Software (training, onderhoud beleggen, ervaring opdoen).

A Literatuurlijst

Naast de al in deze “white paper” gerefereerde URLs wil ik graag nog de volgende boeken ter nadere bestudering aanbevelen:

1. The Cathedral and the Bazaar; Eric S. Raymond.
2. Open Sources, voices from the Open Source revolution; O'Reilly .
3. Open Source, the unauthorized white papers; Donald K. Rosenberg.
4. Embracing insanity, Open Source Software Development; Russell Pavlicek and Robin Miller.