

Open source: een standaard rol voor de overheid?

Robbin te Velde

Senior onderzoeker, Zenc.

Robbin.tevelde@zenc.nl

Inleiding

Open source is zo langzamerhand een uitgekauwd onderwerp maar echt doorgebeten is er nog niet. Veel discussies ontwaarden al snel in een geloofsstrijd tussen gelovigen van het gesloten en van het open model. Ondertussen zien buitenstaanders door de bomen het bos niet meer en kunnen ze niet goed inschatten welke gevolgen de opkomst van het fenomeen open source heeft voor hun eigen beleid of strategie.

Een belangrijk deel van de verwarring lijkt voort te komen uit de ongekendheid met het onderwerp. Hierdoor wordt de nadruk gelegd op de nieuwigheid van het fenomeen open source en op de verschillen met het conventionele gesloten (proprietary) model. Juist het proprietary model is echter minder conventioneel dan vaak wordt aangenomen. Anderzijds worden de verschillen tussen proprietary en open source software schromelijk overschat. Software onderscheidt zich als marktproduct op een aantal belangrijke punten van conventionele fysieke producten. Proprietary en open source software delen deze karakteristieken. Voor zover er nieuw beleid nodig is ten aanzien van open source is dat omdat deze karakteristieken nooit goed zijn bestudeerd. Het open source model vergroot hooguit de consequenties van de kenmerken.

Het virtuele product software

Alles dat kan worden gedigitaliseerd is een informatiegoed.¹ Software is een informatiegoed par excellence. De waarde van een informatiegoed staat volledig los van zijn fysieke verschijningsvorm. Ze wordt geheel bepaald door de manier waarop de elementaire dragers van de informatie zijn gerangschikt. Een computerprogramma is een eindeloze partituur van nullen en enen. Het is de specifieke volgorde van de bits die de waarde van de code bepaalt. Door deze eigenschappen verschillen informatiegoederen in drie opzichten van conventionele fysieke goederen:

1. ze zijn niet-rivaliserend (als iemand anders dezelfde software elders gebruikt gaat dat niet ten koste van de oorspronkelijke consument);
2. hun productie gaat gepaard met enorme schaalvoordelen (de initiële productie van software is een kostbare aangelegenheid maar reproductie kost bijna niets);
3. Hun waarde wordt pas tijdens de consumptie ervaren (de consument kan vooraf de waarde van het goed voor haar of hem niet goed beoordelen – de verschijningsvorm zegt immers niets over de waarde van het goed).

Op grond van het eerste kenmerk kan software worden geclassificeerd als een semi-collectief goed. Of het een puur collectief goed is, dat wil zeggen of er naast non-rivaliteit ook sprake is van non-exclusiviteit, hangt niet af van het inherente karakter van informatiegoederen maar van de omringende juridische context. In theorie kan elk publiek goed exclusief worden gemaakt door allerhande technische (installatiecodes, dongles) of juridische (licenties) kunstgrepen.² Hierbij geeft de afweging tussen de kosten van de ingrepen en de baten van de bescherming uiteindelijk de doorslag. Zo bedragen de kosten die gepaard gaan met het tegengaan van de verspreiding en het gebruik van illegale (sic!) kopieën een veelfout van de

¹ C. Shapiro en H.R. Varian (1998). *Information rules: a strategic guide to the networked economy*. Harvard Business School: Harvard.

² R. Coase (1988). *The Firm, the Market, and the Law*. University of Chicago Press: Chicago.

reproductiekosten van software. De beslissing van enkele grote leveranciers (o.a. IBM, Hewlett-Packard) om zelf open source software te gaan ontwikkelen kan worden beschouwd als een pragmatische keuze om te besparen op beschermingskosten. De opbrengsten worden dan uit andere producten (complementaire proprietary software) of diensten (beheer, opleidingen) gehaald.

De problemen rond de eigendomsrechten van proprietary software zijn inherent aan het karakter van het informatiegoed software en hebben slechts ten dele te maken met de opkomst van open source. De laatste – non-exclusieve – variant van software verandert een semi-collectief alleen in een puur collectief goed. In beide gevallen zou onder normale marktomstandigheden het goed niet zonder hulp van buitenaf worden geproduceerd. Bij nadere beschouwing blijkt inderdaad het overgrote deel van software-innovaties voort te komen uit (semi-)publiek onderzoek.³ Is er wel sprake van significante productie van nieuwe software door de private sector dan heeft de overheid vaak een zware hand in het instandhouden van de exclusiviteit, bijvoorbeeld door creatief te boekhouden met wetgeving. Zo werden (software) algoritmes tot aan het midden van de jaren '80 door de US Patent Office nog gelijkgesteld aan wiskundige formules en dus als publiek eigendom beschouwd. Onder druk van de Amerikaanse industrie zijn ze daarna alsnog patenteerbaar – en dus toeigenbaar – gemaakt.

De virtuele markt voor software

Anno 2003 kijkt niemand er meer van op dat er forse bedragen moeten worden neergeteld voor wat CD's in een ruim uitgevallen kartonnen doos. Het idee dat er geld kan worden gevraagd voor software (dat wil zeggen voor gecompileerde broncode) is echter van relatief recente datum. Vanaf de opkomst van de mainframe computer tot aan het begin van de jaren '80 werd software vaak gratis bij de machine geleverd. Programmeurs wisselden veelvuldig code uit en er werd op grote schaal enthousiast in eigen beheer software ontwikkeld. Een situatie die veel overeenkomsten vertoont met de hedendaagse ontwikkeling van open source software. Pas toen computers op veel grotere schaal werden afgezet en de marges op de machines daardoor sterk daalden, gingen bedrijven broncode afschermen en geld vragen voor de gecompileerde binary⁴.

Bij de productie van software treden enorme schaalvoordelen op omdat de reproductiekosten verwaarloosbaar klein zijn.⁵ Dit geldt zowel voor proprietary als open source software. Dat kostenvoordeel heeft – althans in een markt met volkomen mededinging – de onvermijdelijke keerzijde dat de marginale opbrengsten ook verwaarloosbaar klein zijn. Nota bene, vaste kosten (waaronder de initiële kosten voor de ontwikkeling van de broncode) zitten niet in de vergelijking. Ze moeten of door een derde partij worden opgebracht (overheid) of op de consumenten worden afgewenteld door de prijzen kunstmatig hoog te houden⁶. Dat laatste kan alleen door de exclusiviteit te garanderen. Als aan die conditie is voldaan, treden er zowel aan de aanbod- als aan de vraagzijde *toenemende* meeropbrengsten op. Aan de aanbodzijde

³ Een selectie van voorbeelden: Unix (Bell Labs), TCP/IP (DoD/UCLA/Stanford), X Window System (MIT), BSD-familie (w.o. Unix 4.2) en PostgreSQL (Berkeley), MP3 (Fraunhofer).

⁴ De ontwikkeling van Unix is illustratief. In 1969 voortgekomen uit een hobbyproject van een aantal medewerkers van een semi-publiek R&D-lab bleef het tot aan het eind van de jaren '70 een open source product. Pas daarna doken de eerste proprietary varianten op.

⁵ P.M. Romer (1994). The Origins of Endogenous Growth. *Journal of Economic Perspectives* 8(1): 3-22.

⁶ Dit is althans de heersende opvatting binnen de economische wetenschap. In een recent (2002) invloedrijk artikel betogen BOLDRIN en LEVINE echter – op louter theoretische gronden – dat patenten en copyrights de ontwikkeling van nieuwe producten eerder *tegenwerken* dan stimuleren (*American Economic Review Papers and Proceedings* 92(2): 209-212). Eerder hadden BESSEN en MASKIN al op basis van empirisch onderzoek geconcludeerd dat in de meest innovatieve industrieën van dit moment de bescherming van intellectueel eigendom juist relatief zwak is (MIT Working Paper 11/99, MIT/Dept. Economics, 2000).

leiden de zeer lage marginale kosten ertoe dat de toename in winst nagenoeg evenredig is aan de toename in omzet. Fabrikanten die grote aantal kopieën verkopen kunnen daardoor kleinere partijen relatief makkelijk uit de markt duwen (of ze overnemen). Aan de vraagzijde leidt het wijdverspreid gebruik van bepaalde pakketten tot het ontstaan van de facto standaarden. Het bedrijfsmodel voor proprietary software draait daarom op grote aantallen. De bijhorende marktstructuur tendeeft naar een monopolie.

De hoge mate van concentratie in de markt voor proprietary software brengt een onverwacht voordeel mee voor de consument. Het derde fundamentele kenmerk van informatiegoederen – het feit dat het *experience goods* zijn – heeft tot gevolg dat het moeilijk is om vooraf te bepalen welk product het beste voldoet aan de verwachtingen van de consument. Voor een ex ante oordeel van de kwaliteit van het product moet de consument zich dan verlaten op de mening van deskundige derden (professionele reviewers) of op de reputatie van de leverancier.⁷ In het geval van een sterk geconcentreerd aanbod zijn dit soort keuzeproblemen beperkt – om de eenvoudige reden dat er minder te kiezen valt. De keuze wordt bovendien nog vergemakkelijkt als er, zoals in de neoklassieke economische theorie, een verband wordt gelegd tussen kwaliteit en marktaandeel. Als iedereen product X gebruikt moet het wel een goed product zijn. En vica versa. Een cirkelredenering waar geen spelt valt tussen te krijgen, als men tenminste aanneemt dat consumenten vrij zijn om te kiezen⁸. Dat laatste nu is niet altijd het geval.

In de race om kritische massa spelen gesloten standaarden een cruciale rol. In een markt waar alles draait om grote aantallen is het koppelen van nieuwe aan bestaande producten een van de meest beproefde strategieën om de kritische massa van een bedrijf te vergroten. Hoe wijder vertakt het stelsel van complementaire stukken software en hardware hoe hoger de kosten voor een gebruiker zijn om over te schakelen op concurrerende producten (*switching costs*). Zowel hardware als softwarefabrikanten maken zich schuldig aan dit soort koppelverkoop. Hewlett Packard verkoopt haar servers alleen met een eigen versie van Unix (HP-UX) en Microsoft heeft haar Mediaplayer tot een integraal onderdeel van Windows gemaakt. In het laatste geval kunnen gebruikers alleen nog audio- en videobestanden in 'gecontroleerde' bestandstypen afspelen. Dit soort harde koppelingen tussen verschillende softwarecomponenten van een en dezelfde fabrikant is alleen mogelijk door het gebruik van gesloten standaarden. Analoog aan de bescherming van afzonderlijke componenten door het afschermen van de broncode wordt in dit geval de koppeling tussen de componenten afgeschermd. Het is technisch erg moeilijk voor een derde partij om nieuwe componenten (bijvoorbeeld een hardwaredriver) aan een bestaande installed base toe te voegen als de standaard – de 'uitwisselingstaal' tussen de componenten, niet bekend is. Let wel, gegeven de inherente karakteristieken van informatieproducten is het gebruik van (zo min mogelijk) standaarden in het belang van iedereen. Dit geldt zowel voor proprietary als open source software. In het eerste geval zullen fabrikanten er echter alles aan doen om ervoor te zorgen dat hun eigen standaard de algemene standaard wordt. Ieder dient God op zijn manier. U op de uwe en ik op de Zijne.

De markt voor open source

Het conventionele bedrijfsmodel dat is gebaseerd op de verkoop van software staat en valt met de garantie van exclusiviteit en met de overwinsten (en dominante marktposities) die daardoor kunnen ontstaan. Het open source model zet deze conditie buitenspel en is daarmee een frontale aanval op het proprietary bedrijfsmodel. In het nieuwe model komen de inkomsten noch uit software noch uit complementaire hardware (mainframe era) maar uit ondersteunende diensten zoals pakketselectie, beheer, aanpassing en opleiding. Het grote nadeel van dit model is dat het niet de schaalvoordelen heeft van het voorgaande proprietary

⁷ C. Shapiro en H.R. Varian. Op.cit.

⁸ Zie bijvoorbeeld het werk van LIEBOWITZ en MARGOLIS over de – in hun ogen onterechte – beschuldiging van machtsmisbruik door Microsoft.

model. Tegenover de productie van elke extra eenheid diensten staat de inzet van een extra eenheid arbeid. Commerciële open source bedrijven opereren daarom vaak op basis van een hybride bedrijfsmodel: ze halen de basisinkomsten uit beheer en ondersteuning en de extra marges uit complementaire proprietary software.⁹ Ze zullen daarom nimmer de omvang halen van softwaregiganten zoals Microsoft of Oracle. Dit gebrek aan kritische massa wordt echter gecompenseerd door het feit dat ze bij de (door)ontwikkeling van code gebruik kunnen maken van soms uitgebreide virtuele netwerken van open source programmeurs. Deze programmeurs werken op hun beurt vaak als eenpitters.¹⁰ De opkomst van het open source model leidt zo tot een fragmentatie van de – voorheen zo geconcentreerde – aanbodzijde.

Deze fragmentatie wordt nog verder in de hand gewerkt doordat het gebruik van open source en open standaarden hand in hand gaan. Bedrijven die geen inkomsten halen uit de verkoop van software hebben geen strategische redenen meer om de interoperabiliteit met applicaties van concurrenten te frustreren. Sterker nog, hoe meer applicaties er draaien onder een bepaalde standaard hoe sneller die standaard een verspreiding vindt en hoe groter het potentiële aantal klanten dat om de ondersteunende diensten zal vragen. De ontwikkeling van open source applicaties verloopt over het algemeen nogal dynamiek. In tegenstelling tot de ontwikkeling van closed source software door een beperkt aantal grote leveranciers is er, door de versplintering van de aanbodzijde vaak sprake van een hoge mate van variatie. Hoewel ontwikkelaars van open source software er vaak alles aan doen om het ontstaan van zijtakken (*forks*) te voorkomen is dit fenomeen in de definitie van open source besloten.¹¹

De fragmentatie van de aanbodzijde gaat nog verder dan het uiteenvallen van de proprietary codefabrieken in duizenden aparte open source ontwikkelcommunities. In de markt voor proprietary software zijn alle functies rondom de software (ontwikkeling, aanpassing, kwaliteitsgarantie, onderhoud, beheer, opleiding, ondersteuning) vaak in een hand verenigd. Hooguit is er op generiek niveau een marktsegmentatie in ontwikkeling, implementatie en beheer. Een van de meest prominente argumenten tegen het gebruik van open source software is dat bijvoorbeeld ondersteuning en wettelijke aansprakelijkheid voor de producten ontbreken.¹² Dit valt eerder te wijten aan de onvolgroeidheid van de markt voor open source producten en diensten dan aan structurele oorzaken. Er zijn geen gegronde redenen om aan te nemen dat de functies die hierboven genoemd zijn niet *afzonderlijk* door bedrijven kunnen en zullen worden opgepakt. De ondersteuning van open source software vormt zelfs een interessante groeiemarkt voor nieuwe en gevestigde spelers omdat door de grotere mate van variatie in de markt functies zoals advies (de eerder genoemde professionele review), implementatie (meer ruimte voor maatwerk) en support (effectievere ondersteuning door open broncode) belangrijker zullen worden.

Het feit dat het open source model een veel breder aanbod van applicaties kent dan het gesloten model brengt het derde kenmerk van informatiegoederen wederom naar de voorgrond. Voor buitenstaanders brengt de grote mate van dynamiek inderdaad de nodige verwarring met zich mee en daar wordt door tegenstanders van het model dan ook dankbaar gebruik van gemaakt¹³. Voor professionele reviewers geeft deze dynamiek juist de

⁹ Zie de site van het open source bedrijf Extropia (<http://www.extropia.com>) voor een aantal inside stories over de worsteling met verschillende bedrijfsmodellen.

¹⁰ P. van Lith (2003). Community programming als ontwikkelmethode. *Informatie* 45 (juli-augustus): 23-25. Zie verder wederom <http://www.extropia.com>.

¹¹ J. Visser (2002). *Het Open Source Fenomeen*. <http://www.ososs.nl/attachment.db?998>

¹² <http://www.linuxworld.com/story/32698.htm>. Citaat van de gewraakte memo van Microsoft-topman Steve Ballmer: "IBM's endorsement of Linux has added credibility and an illusion of support and accountability, although the reality is there is no 'center of gravity,' or central body, investing in the health and growth of non-commercial software or innovating in critical areas like engineering, manageability, compatibility and security."

¹³ Gedoeld wordt hier op de beruchte FUD-strategie: het (doelbewust) genereren van angst (*fear*), onzekerheid (*uncertainty*) en twijfel (*doubt*) bij consumenten. Microsoft heeft geen patent op FUD. De

mogelijkheid om op basis van de vitaliteit van bepaalde ontwikkeltrajecten de kwaliteit van de software te beoordelen. Daarnaast zijn ze in staat om – althans in eerste oogopslag – de interne werking van componenten te beoordelen op basis van de structuur van de broncode, die immers vrij beschikbaar is¹⁴.

De rol van de overheid

De markt voor proprietary software is niet alleen geconcentreerd in termen van aantal bedrijven maar ook in termen van geografische spreiding. Van de 50 grootste softwareleveranciers is bijna 90% afkomstig uit Noord-Amerika.¹⁵ In Nederland vindt nauwelijks commerciële ontwikkeling van software plaats. De softwarehuizen die er zijn richten zich met name op implementatie of beheer. Voor deze bedrijven maakt het welbeschouwd weinig uit of de software van derden die ze gebruiken proprietary of open source is. De extreem onevenwichtige handelsbalans zorgt voor een stevige impuls voor het gebruik van open source software door de publieke sector in Europa. Men kan immers de vruchten plukken (geen torenhoge licentiekosten) zonder de kosten te hoeven dragen (uitholling van de eigen industriële basis). Het veelgehoorde argument dat dit soort gedrag de ontwikkeling van innovatieve software zou bedreigen snijdt weinig hout omdat het gros van het vernieuwend onderzoek reeds van oudsher door publieke instanties wordt gedaan. Hooguit zou, met name op Europees niveau, het publieke onderzoek in dit domein verder kunnen worden versterkt.

De opkomst van het open source model vormt inderdaad een tweede kans voor de Europese softwareindustrie. Voorwaarde is dan wel dat het gebruik van gesloten standaarden actief wordt ontmoedigd. Voor de Nederlandse softwarehuizen betekent dit een grondige herziening van hun bestaande bedrijfsmodel dat immers ook voor een belangrijk deel op het gebruik van gesloten standaarden steunt. Koppelverkoop in eendrachtige samenwerking met leveranciers is in het nieuwe regime niet langer mogelijk. Er is echter een wereld te winnen in de ondersteunende diensten voor het gebruik van open source. Het Nederlandse onderzoeks- en industriebeleid zou zich moeten richten op het versterken van de kennis en kunde van open source binnen Nederlandse bedrijven.

In de ontwikkelingsfase zou de overheid zelf een belangrijke rol kunnen spelen als lead user, niet alleen om op grote schaal software uit te testen maar ook om tegenwicht te bieden aan de FUD-strategieën van leveranciers van proprietary software. Dit vraagt om een poractieve houding ten aanzien van de ontwikkeling van software en om kundig opdrachtgeverschap¹⁶. Dit vraagt ook om een actieve en uitgesproken rol van de overheid in de totstandkoming van (open) standaarden en protocollen. Directe bemoeienis zoals in het geval van het EDISON-project is, zeker in een steeds volwassener markt, niet meer nodig. Private partijen kunnen op grond van hun expertise zorg dragen voor de selectie van leveranciers van software en ondersteunende diensten. De overheid zou hooguit een rol kunnen spelen in het garanderen

strategie werd in de jaren '60 en '70 reeds geperfectioneerd door IBM (nu een sympathisant van open source, zie voorgaande noot) en is sedertdien toegepast door hightechbedrijven van velerlei pluimage. WHATIS.TECHTARGET meldt verder dat de strategie effectiever is naarmate het bedrijf groter is (cf. argumentatie rond *branding* van proprietary software).

¹⁴ M. Nijdam (2003). Vijf adviezen voor selectie van oss-componenten. *Informatie 45* (juli-augustus): 28-30.

¹⁵ D. Lancashire (2001). Code, Culture and Cash: The Fading Altruism of Open Source Development. *Firstmonday 6*(21). http://www.firstmonday.dk/issues/issue6_12/lancashire

¹⁶ Een extreem voorbeeld hiervan is de uitgekende strategie van variatie en selectie door de Belgische overheid bij de ontwikkeling van het EDISON-project. Door middel van open en doeltreffende communicatie heeft de projectleiding leveranciers weten te bewegen mee te werken aan de vertaling van beleidsveranderingen in software. Naarmate het traject vordert worden de eisen aan de leveranciers opgeschroefd. De Belgische overheid stelt daarmee in feite de specificaties op maar ze neemt de software ook van de ontwikkelaars af en stelt deze ter beschikking aan de gebruikers (in casu scholen). Op deze manier dicteert ze de markt voor zowel gebruikers als ontwikkelaars.

van de deskundigheid en onpartijdigheid van deze bedrijven, bijvoorbeeld door middel van certificering¹⁷.

Wat betekent dit nu voor het beleid van de Nederlandse overheid? Welnu, alle voorafgaande argumenten kunnen worden samengevat in twee stelregels voor robuust overheidsbeleid ten aanzien van open source software:

1. Selecteer leveranciers van software en van ondersteunende diensten louter op grond van de kwaliteit die ze leveren.
2. Maak alleen gebruik van gesloten standaarden als het echt niet anders kan.

Niet meer en niet minder.

¹⁷ Het zelf inhuren van leveranciers en bemiddelaars is een impliciete vorm van certificering. De overheid dient zich terdege rekening te geven van dit effect.