

CPB Document

No 181

Juli 2009

**Concurrentie, innovatie en intellectuele
eigendomsrechten in softwaremarkten**

Michiel Bijlsma, Paul de Bijl en Viktória Kocsis

Centraal Planbureau
Van Stolkweg 14
Postbus 80510
2508 GM Den Haag, Nederland

Telefoon +31 70 338 33 80
Telefax +31 70 338 33 50
Internet www.cpb.nl

ISBN 978-90-5833-397-1

English summary

This study analyzes under which circumstances it may be desirable for the government to stimulate open source software as a response to market failures in software markets. To consider whether policy intervention can increase dynamic efficiency, we discuss the differences between proprietary software and open source software with respect to the incentives to innovate and market failures that may occur. The document proposes guidelines to determine which types of policy intervention may be suitable. Our most important finding is that directly stimulating open source software, e.g. by acting as a lead customer, can improve dynamic efficiency if (i) there is a serious customer lock-in problem, while (ii) to develop the software, there is no need to purchase specific, complementary inputs at a substantial cost, and (iii) follow-on innovations are socially valuable but there are impediments to contractual agreements between developers that aim at realizing such innovations.

Key words: Software markets, Intellectual property rights, Open source software, Public policy
JEL code: L17, L52, L86, O34

Nederlandse samenvatting

Deze studie analyseert wanneer het wenselijk is dat de overheid open source software stimuleert in reactie op marktfalen in softwaremarkten. Om te beoordelen of beleidsinterventie de dynamische efficiëntie kan verhogen, bespreken we de verschillen tussen ‘propriëtaire’ en open source software met betrekking tot de prikkels om te innoveren en marktfalen dat kan optreden. Deze studie stelt richtlijnen voor om te bepalen welke vormen van ingrijpen geschikt kunnen zijn. De belangrijkste bevinding is dat het direct stimuleren van open source software, bijvoorbeeld door als aanjagende klant op te treden, de dynamische efficiëntie kan vergroten als (i) de leveranciersafhankelijkheid groot is, terwijl (ii) er in de ontwikkelingsfase geen specifieke, complementaire input tegen substantiële kosten ingekocht hoeft te worden, en (iii) vervolginnovaties sociaal wenselijk zijn maar er beperkingen zijn om die door middel van contracteren tussen ontwikkelaars tot stand te laten komen.

Steekwoorden: Softwaremarkten, Intellectuele eigendomsrechten, Open source software, Overheidsbeleid

Inhoud

Voorwoord	7
Samenvatting	9
Voorwoord	6
Summary (Engelse samenvatting)	7
Nederlandse samenvatting	10
1 Inleiding	13
2 Definities en beschrijvingen	17
3 Kenmerken van open source en propriëtaire software	21
3.1 Prikkel om deel te nemen in de ontwikkeling van open source software	21
3.2 Marktfalen met betrekking tot software	28
3.3 Effecten van innovatieprikkel marktfalen op software	33
4 De mogelijke rol van de overheid	37
4.1 Beleidsimplicaties	37
4.2 Overheidsfalen	48
4.3 Samenvatting	49
5 Conclusies en discussie	51
Literatuur	53

Voorwoord

Het Nederlandse parlement en de Nederlandse regering hebben de intentie efficiënte elektronische informatie-uitwisseling te stimuleren en zien daarbij een belangrijke rol weggelegd voor innovaties, met name gericht op interoperabiliteit en concurrentie in softwaremarkten. Er zijn daarom diverse maatregelen voorgesteld om de toepassing van open standaarden en open source software te stimuleren in de publieke en de semipublieke sector. Op verzoek van het parlement in 2007 heeft het ministerie van Economische Zaken het actieplan *Nederland Open in Verbinding* gepubliceerd over de manier waarop deze ideeën moeten worden voorbereid en geïmplementeerd. Het ministerie van Economische Zaken heeft het CPB gevraagd de economische effecten op concurrentie en innovatie te analyseren van het stimuleren van open source software. Gebaseerd op bestaande inzichten uit de economieliteratuur analyseert deze studie beleidsopties in het licht van de kenmerken van de softwaremarkt, zoals stimuleringsmaatregelen voor innovaties onder verschillende typen softwarelicenties en soorten marktfalen die in deze markten kunnen optreden. Deze studie stelt richtlijnen voor om te bepalen onder welke omstandigheden de diverse soorten beleidsinterventies wenselijk kunnen zijn om de dynamische efficiëntie te verhogen.

Dit document is geschreven door Michiel Bijlsma, Paul de Bijl en Viktória Kocsis. Viktória Kocsis was hierbij de projectleider. De auteurs zijn zeer erkentelijk voor de uitgebreide opmerkingen van en de discussies met de adviescommissie van het ministerie van Economische Zaken. Deze commissie bestond uit Jan Julianus, die het initiatief nam voor het contact met het CPB dat aanleiding was tot dit project en die de bijeenkomsten voorzat, Margreet Groenenboom, Yvonne van Santen, Hendrik Brandwijk en Tjade Stroband. De auteurs willen ook hun dank betuigen aan Rob Aalbers (CPB), Jan Boone (UvT), Madeleine de Cock Buning (CIER/UU), Casper van Ewijk (CPB), Marco Haan (RUG), Berend Hasselman (CPB), Pierre Larouche (TILEC), Hans Lunsing (CPB), Bas Straathof (CPB), Erwin Zijleman (CPB) en deelnemers aan onze interne seminars voor nuttige suggesties en discussies. De verantwoordelijkheid voor de inhoud en de conclusies van dit rapport ligt volledig bij het CPB.

Coen Teulings

Directeur CPB

Samenvatting

Met het Actieplan *Nederland Open in Verbinding* beoogt het ministerie van Economische Zaken het gebruik van open source software (OSS) te stimuleren, ten behoeve van interoperabiliteit, innovatie en terugdringen van afhankelijkheid van softwareleveranciers. Het ministerie heeft het CPB verzocht onderzoek te doen naar de relatie tussen het stimuleren van OSS en innovatie en concurrentie in de softwaremarkten. Dit document gaat daarom in op de volgende vragen:

1. Hoe beïnvloeden verschillende licentiemodellen de prikkels om software te ontwikkelen?
2. Wat zijn de soorten marktfalen in softwaremarkten?
3. Hoe wordt welvaart beïnvloed door innovatieprikkels en marktfalen?
4. Wat kunnen beleidsmakers doen om softwaremarkten beter te laten werken?

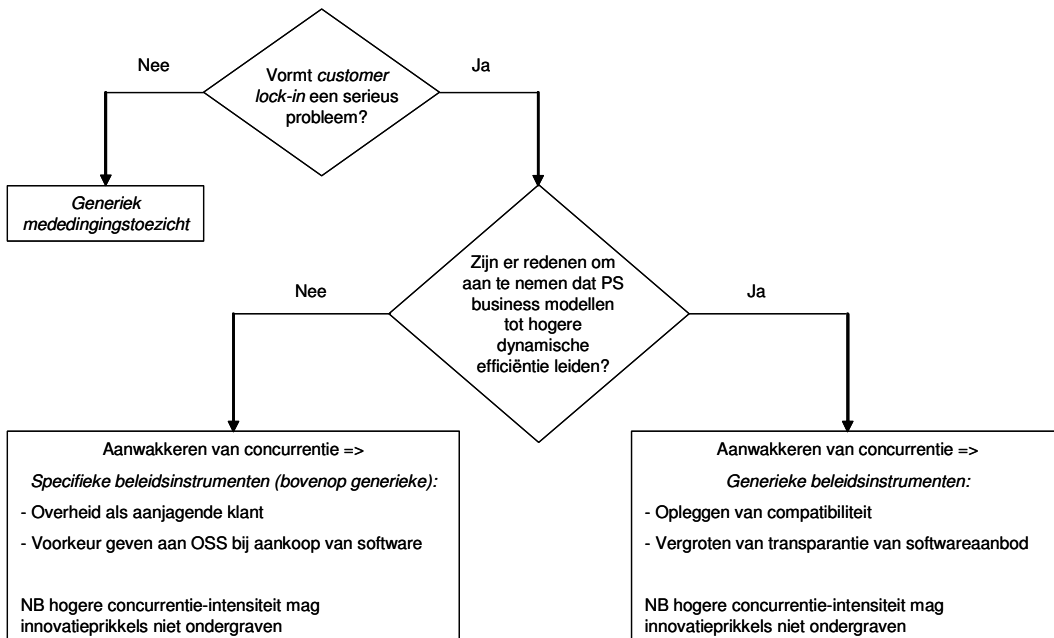
Op software rusten in Europa intellectuele eigendomsrechten in de vorm van auteursrechten. Verschillende softwarevormen gebruiken auteursrechten op een verschillende manier. Ten eerste is er *propriétaire software* (PS). Bij PS maken auteursrechten een tijdelijk monopolie mogelijk ten aanzien van reproductie, distributie en ontwikkeling van afgeleide werken. Dit beschermt ontwikkelaars tegen directe imitatie en stelt hen daardoor in staat consumenten of bedrijven te laten betalen voor gebruik van het product. Ten tweede is er OSS, dat veelal gratis beschikbaar is en biedt de licentie toegang tot de broncode, zodanig dat anderen deze mogen wijzigen, verbeteren en distribueren zonder daarvoor te hoeven betalen aan de houder van het auteursrecht. De licentievoorwaarden van PS creëren vooruitzichten op winst uit directe verkopen vanwege de mogelijkheid om een mark-up in de prijs op te nemen. Dit waarborgt, vooral bij forse ontwikkelkosten, de ex ante prikkels om te innoveren. De licentievoorwaarden van OSS beperken de mogelijkheid marktmacht uit te oefenen, omdat iedereen op basis van de broncode zijn eigen variant kan ontwikkelen en verspreiden. Innovatieprikkels komen dus uit een andere hoek, zoals de winst door verkoop van complementaire producten (bijvoorbeeld ondersteunende diensten, advies, en complementaire PS) en prikkels van individuele ontwikkelaars. Dat kunnen zowel intrinsieke prikkels zijn (motief uit eigen beweging) als extrinsieke (bijvoorbeeld kwaliteiten demonstreren op de arbeidsmarkt, leereffecten, eigen gebruik).

Zowel PS als OSS heeft voor- en nadelen die gerelateerd zijn aan soorten marktfalen die kunnen optreden in softwaremarkten. De belangrijkste zijn *marktmacht* en *kennisspillovers*. Marktmacht kan ten eerste ontstaan door netwerkexternaliteiten (gebruikers ontlene meer nut naarmate er een groter gebruikersnetwerk is) en overstapkosten (materiële of immateriële kosten van overschakelen op andere software). In beide gevallen kan leveranciersafhankelijkheid (*customer lock-in*) ontstaan. Een tweede oorzaak betreft schaaleffecten die vooral belangrijk worden wanneer er specifieke, complementaire expertise

ingekocht dient te worden ten behoeve van de ontwikkeling van software, aanvullend op de expertise van programmeurs. Denk bijvoorbeeld aan grafische expertise bij *games*. Kennisspillovers zijn vooral problematisch wanneer het vooruitzicht op *free riding* het minder aantrekkelijk maakt om R&D te verrichten. PS-licenties corrigeren daar in beginsel voor, omdat zij *free riding* tegengaan. Een nadeel echter is dat zij daarmee (te) beperkend kunnen zijn voor innovaties die voortbouwen op bestaande kennis. Beide softwaretypen hebben in dit opzicht dus voor- en nadelen: OSS kan sneller vervolginnovaties aanwakkeren, maar PS voorkomt dat anderen gratis meeliften op bestaande innovaties (wat gunstig is voor ex ante prikkels).

Deze studie doet een concreet voorstel om de verschillende vormen van mogelijk marktfalen in softwaremarkten te beoordelen, met als doel te bepalen of een beleidsinterventie wenselijk is. De volgende figuur illustreert het centrale idee.

Figuur 1 Beslissingsboom voor beleidsmakers



Het schema werkt als volgt:

1. De eerste stap geeft aan dat wanneer *customer lock-in* geen serieus probleem vormt – en er dus geen grote toetredingsbarrières zijn – generiek mededingingstoezicht volstaat om softwaremarkten goed te laten functioneren.
2. Wanneer de leveranciersafhankelijkheid in een bepaalde softwaremarkt groot is, is het belangrijk om eerst te bepalen of het wenselijk is om beleid specifiek te richten op OSS of om het generiek te richten op alle software. Ten eerste, wanneer in de ontwikkelingsfase dure expertise ingekocht moet worden, biedt PS betere mogelijkheden om de vooraf gemaakte kosten terug te verdienen. Ten tweede kan

OSS kan sneller vervolginnovaties aanwakkeren, maar PS voorkomt dat anderen gratis meeliften op bestaande innovaties. Afhankelijk van deze factoren kan bepaald worden welk type beleid het meest geschikt is om afhankelijkheid van leveranciers terug te dringen.

Het is belangrijk om na te gaan of het aanwakkeren van concurrentie via OSS innovatieprikkelers zou kunnen schaden, wat in theorie zou kunnen optreden wanneer er al een hoge mate van concurrentie is om mee te beginnen. Echter, wanneer de markt op slot zit door *customer lock-in* en de gevestigde aanbieder daardoor weinig concurrentiedruk ervaart, is het waarschijnlijk dat meer concurrentie innovatie juist aanwakkert in plaats van verder afzwakt.

De beslissingsboom geeft aan wanneer beleid specifiek gericht op OSS, dan wel generiek beleid, wenselijk kan zijn om toetredingsbarrières te slechten en daarmee de welvaart op lange termijn te vergroten. Voorbeelden van specifiek beleid zijn dat de overheid *lead customer* (aanjagende klant) wordt of dat de overheid oplegt dat voor eigen inkoop voorkeur gegeven dient te worden aan OSS boven PS. Een voorbeeld van generiek beleid is het opleggen van volledige compatibiliteit, zodat bestanden naadloos uitgewisseld kunnen worden tussen applicaties van verschillende leveranciers.

Vanwege het risico van overheidsfalen is bij dergelijk beleid zorgvuldigheid geboden. De overheid heeft immers een informatieachterstand ten opzichte van marktpartijen, waardoor het in de regel verstandig is om geen beleid in gang te zetten dat bij voorbaat een winnaar selecteert. Ook geeft specifiek beleid vaak aanleiding tot lobbyen om het beleid in een vroeg stadium te beïnvloeden. Vanwege de informatieachterstand van de overheid kan het beleidsontwerpproces verstoord worden, nog los van de inefficiënties van het lobbyproces zelf. Dat speelt overigens minder naarmate beleid generieker is, dus minder gericht op een bepaald type software.

1 Inleiding

Op software rusten in Europa eigendomsrechten in de vorm van auteursrechten.¹

Auteursrechten, en meer algemeen intellectuele eigendomsrechten (IER), kennen ontwikkelaars tijdelijke, exclusieve rechten toe voor het reproduceren en distribueren van de onderliggende code, alsmede voor het ontwikkelen van afgeleide werken.

Er zijn twee hoofdvormen voor het uitgeven van licenties voor software: licenties voor propriëtaire software en voor open source.² Bij 'traditionele' propriëtaire software (PS) wordt de onderliggende code niet openbaar gemaakt zodat er niet door anderen aan de software gesleuteld kan worden, wat de bedoeling ook niet is.³ Consumenten of bedrijven kunnen een bepaald programma alleen gebruiken met toestemming van de licentiehouder. Om toestemming te krijgen moeten gebruikers de licentievoorwaarden accepteren en in veel gevallen betalen voor het gebruik van de software.⁴ Als de software wordt gedistribueerd of gebruikt zonder toestemming van de auteursrechtenhouder is dit een overtreding van de wet op de auteursrechten.

Het laatste decennium is ook een andere vorm van het uitgeven van licenties ontstaan in softwaremarkten, die in het algemeen open-source licenties worden genoemd. In het geval van open source software (OSS) wordt de broncode beschikbaar gesteld aan andere ontwikkelaars onder voorwaarden zoals gespecificeerd in de licentie. OSS-licenties staan in het algemeen toe dat anderen de broncode wijzigen en verbeteren en de software distribueren zonder betaling aan de licentiehouder. Klanten kunnen de broncode vervolgens aanpassen, verbeteren of distribueren en ook het uiteindelijke (aangepaste) programma distribueren, onder de voorwaarden zoals vastgelegd in de oorspronkelijke open-source licentie. Met andere woorden, OSS-licenties zijn uitgekilde versies van auteursrechten, in die zin dat de uitgever van de licentie minder rechten claimt dan hij mogelijk als eigenaar van de auteursrechten zou kunnen doen gelden. Net als bij PS, worden open-source licenties afgedwongen. Niet naleven van de voorwaarden ervan is een schending van de wet op de auteursrechten.

Het Nederlandse parlement en de Nederlandse regering hebben de intentie efficiënte elektronische informatie-uitwisseling te stimuleren. De regering ziet een belangrijke rol weggelegd voor innovaties, met name gericht op interoperabiliteit en concurrentie in softwaremarkten en heeft daarom diverse maatregelen voorgesteld om de toepassing van open standaarden en open-source software te stimuleren in de publieke en de semi-publieke sector. Op verzoek van het parlement in 2007 heeft het ministerie van Economische Zaken het actieplan *Nederland Open in Verbinding* gepubliceerd over de manier waarop deze ideeën

¹ In de VS worden patenten gebruikt om bedrijfsprocessen te beschermen die zijn vertaald naar software. Denk bijvoorbeeld aan een vertaald proces zoals het 'one-click'-inkoopformat.

² Merk op dat beide soorten licenties zijn gebaseerd op IER.

³ Als alternatief kan de broncode privé worden vrijgegeven, onder zeer strenge geheimhoudingsverplichtingen (de licentievoorwaarden kunnen in dat geval ook vertrouwelijk zijn).

⁴ Een voorbeeld van wanneer consumenten niet hoeven te betalen voor gebruik is freeware.

moeten worden voorbereid en geïmplementeerd. Het ministerie van Economische Zaken heeft het CPB gevraagd de economische effecten op concurrentie en innovatie te analyseren van het stimuleren van open-source-software:⁵

“17. Onderzoek naar de economische effecten van open-source software

Het Centraal Planbureau (CPB) heeft de opdracht gekregen een economisch onderzoek uit te voeren naar de relaties tussen de promotie van open-source software en de effecten ervan op innovatie en concurrentie in de ICT-sector. De wetenschappelijke belangstelling voor de economische effecten van de introductie van open-source software is groeiende. Het is verstandig te onderzoeken wat deze effecten zijn, met name op het functioneren van softwaremarkten (bijvoorbeeld in welke mate klanten ervan profiteren).”

In reactie hierop hebben we in deze studie de volgende vragen geformuleerd:

1. Welke prikkels zijn er om software te ontwikkelen onder de verschillende licentiemodellen?
2. Wat zijn de soorten marktfalen in softwaremarkten?
3. Hoe beïnvloeden innovatieprikkels en marktfalen de welvaart?
4. Wat kunnen beleidsmakers doen om marktfalen te verzachten en de welvaart op softwaremarkten te verhogen?

De laatste drie vragen spreken voor zich; de eerste vraag vereist echter enige uitleg.

Innovatieprikkels krijgen speciale aandacht omdat zij belangrijk zijn voor softwaremarkten. Bovendien zijn IER's sterk gerelateerd aan innovatieprikkels, hetgeen ook de bedoeling is. Met betrekking tot de overige vragen nemen we bij het analyseren van de rol van de overheid aan dat zij in het algemeen belang handelt, dat wil zeggen dat zij de welvaart stimuleert. We onderzoeken hoe het stimuleren van OSS de welvaart op de korte termijn (statische efficiëntie) en op de lange termijn (dynamische efficiëntie) beïnvloedt.

Er zijn diverse andere verhandelingen en publicaties die een hieraan gerelateerde of aanvullende beleidsanalyse bevatten over de relatieve waarden van OSS en PS. Enkele referenties zijn Knubben (2001), Schmidt en Schnitzer (2003) en Lee (2006). Zoals de laatste auteur aangeeft, zijn 'beleidsoverwegingen die aanleiding zijn tot overheidsbesluiten, extreem gecompliceerd en soms onderling afhankelijk van elkaar' (p. 112). Deze complexiteit komt voort uit de technologische aspecten in relatie tot diverse vormen en combinaties van marktfalen die zich kunnen voordoen in specifieke softwaremarkten; ze zijn niet uniform voor de softwaremarkt als geheel. Een andere studie, Mendys-Kamphorst (2002), analyseert de effecten van open source-projecten op concurrentie in softwaremarkten.

Ten behoeve van de beknoptheid ervan concentreert deze studie zich uitsluitend op open source- versus propriëtaire licenties in softwaremarkten en behandelt ze niet de problemen en

⁵ Ministerie van Economische Zaken (2007), p.23.

kwesties die betrekking hebben op open versus gesloten standaarden. Gezien het belang en de complexiteit van de aspecten die te maken hebben met het opstellen van standaarden is een afzonderlijke studie die gericht is op standaarden gerechtvaardigd. Niettemin is het van belang op te merken (tevens benadrukt door Lee, 2006) dat beleid dat specifiek OSS ondersteunt, effecten kan hebben waarvoor open standaarden tekortschieten.

Het resterende deel van dit document is als volgt samengesteld: In hoofdstuk 2 bespreken we de concepten die betrekking hebben op de bescherming van intellectuele eigendomsrechten in softwaremarkten. Hoofdstuk 3 presenteert de stimuleringsmaatregelen van deelnemers, de vormen van marktfalen en hun effecten op markttuitkomsten en welvaart. De rol van de overheid wordt besproken in hoofdstuk 4. Hoofdstuk 5 toont de conclusie van het rapport.

2 Definities en afbakening

De rol van intellectuele eigendomsrechten in het innovatieproces

Gesteld kan worden dat de bescherming van intellectuele eigendomsrechten (IER) het exclusieve recht garandeert om een product gedurende een bepaalde periode op de markt te brengen.⁶ Deze exclusieve rechten creëren in feite de mogelijkheid tot een monopolypositie met betrekking tot het betreffende product, waardoor achteraf de investeringskosten dus (deels) kunnen worden terugverdiend door de opbrengsten die hieruit voortvloeien. Twee bekende vormen van IER zijn patenten en auteursrechten. Patenten hebben betrekking op ideeën en komen tot stand na toewijzing van een ingediende aanvraag bij het Octrooibureau. Auteursrechten zijn gericht op het beschermen van de *verwezenlijking* van ideeën en zijn automatisch van toepassing. Vanuit economisch oogpunt moet een IER-model werken als een mechanisme dat prikkels voor innovaties biedt en beschermt. Een dergelijk prikkelmechanisme kan diverse vormen hebben. Daarom heeft een fundamentele kwestie betrekking op de optimale vorm van IER, in die zin dat deze het verlies in welvaart als gevolg van monopolieprijsstelling minimaliseert, onder de voorwaarde dat de prikkels voor innovatie niet worden ondermijnd.⁷ In Europa valt software onder auteursrechten, daarom concentreert deze studie zich meer op auteursrechten dan op patenten.⁸ Aangevoerd kan worden dat auteursrechten niet de perfecte manier zijn om software te beschermen, omdat het de software beschermt als een *kunstwerk* of *literair werk* en derhalve niet voldoende bescherming biedt tegen omzetting van de broncode naar modellen (*reverse engineering*). Dit omvat het onderzoeken van een product (meestal een stuk software of een communicatieprotocol) om daaruit af te leiden wat de eisen zijn waaraan het product probeert te voldoen, of om de precieze interne werking ervan te achterhalen. Dit doet men mogelijk (maar zeker niet uitsluitend) met het doel een concurrerend product te ontwerpen.⁹ Softwareontwikkelaars kunnen ook gebruik maken van een andere optie om de vruchten van hun intellectuele inspanningen te beschermen, namelijk handelsgeheimen.¹⁰

Verschillende soorten licenties voor software

Software is een algemene term voor diverse programma's die reeksen instructies bevatten die door het logische circuit in computers en andere apparaten worden gevolgd. Belangrijke soorten software zijn applicatiesoftware voor computergebruikers (bijvoorbeeld computerspelletjes en

⁶ Deze sectie is voor het overgrote deel gebaseerd op Gallini en Scotchmer (2002) en Guibault en Van Daalen (2005). Zie ook Stiglitz (verschijnt binnenkort).

⁷ Een meer algemene vraag is of er mechanismen zijn naast IER, die gericht zijn op het maximaliseren van de meerwaarde die door innovatie wordt gecreëerd. Voorbeelden van dergelijke mechanismes zijn prijzen en inkoop.

⁸ Hoewel we ons hier concentreren op auteursrechten, heeft de richting van ons economisch betoog met betrekking tot innovatieprikkels en de markt *mutatis mutandis* ook betrekking op patenten.

⁹ De *Auteurswet* staat geen nabouw toe, met uitzondering van bijzondere omstandigheden (art. 45m). Voor meer informatie over nabouwen, zie bijvoorbeeld De Cock Buning (1993).

¹⁰ Handelsgeheimen bestaan uit het niet vrijgeven van de onderliggende broncode, behalve onder zeer strikte voorwaarden in combinatie met zware straffen in geval van overtreding van de geheimhoudingsovereenkomst. Ze worden niet (bijvoorbeeld in internationale verdragen) niet in dezelfde mate als patenten of auteursrechten erkend als IER

tekstverwerkers), systeemsoftware voor de interfaces van hardware die de applicatiesoftware ondersteunen (bijvoorbeeld besturingssystemen) en *middleware* die de verbinding vormt tussen softwarecomponenten die samen moeten functioneren en misschien op verschillende besturingssystemen werken. Software die geïntegreerd is in hardware en wordt aangestuurd door die software wordt ingebedde (*embedded*) software genoemd. Dit type software wordt vaak gebruikt in elektronische apparatuur anders dan computers, zoals telefoons of auto-onderdelen.

We onderscheiden open source software (OSS) en propriëtaire software (PS). OSS is 'software die een vrij beschikbare broncode heeft die de licentienemer in staat stelt de broncode te bekijken, te gebruiken, te verbeteren, uit te breiden en te distribueren' (ministerie van Economische Zaken, 2007, p. 6).¹¹ Deze omschrijving behoeft enige toelichting. Softwareprogramma's kunnen worden gedistribueerd als broncode of als objectcode. De broncode hebben gebruikers nodig als ze het programma willen wijzigen. Een programma in broncode moet worden gecompileerd naar objectcode voordat het kan worden uitgevoerd. In het geval van OSS is de onderliggende kennis van een programma zoals besloten in de broncode publiekelijk beschikbaar voor imitatie, verbeteringen en aanpassingen, zij het onder bepaalde voorwaarden zoals gespecificeerd in de bijbehorende licentie. Dit is meestal niet het geval voor PS, omdat bijvoorbeeld de broncode is achtergehouden of exclusieve juridische rechten de imitatie en het sleutelen aan de code uitsluiten. Niets belet een ontwikkelaar van software met een propriëtaire licentie echter om de broncode openbaar te maken, waarvan akte.

Een auteursrechtenmodel kent ontwikkelaars een exclusief recht toe de onderliggende code te reproduceren en te distribueren, alsmede afgeleide werken te ontwikkelen.¹² OSS-licenties kennen bepaalde voorwaardelijke goedkeuringen toe. OSS-licenties gebruiken het auteursrechtenmodel om ervoor te zorgen dat de broncode of de software gratis beschikbaar is en dat de gebruiker van de software bepaalde rechten heeft om het oorspronkelijke programma te distribueren en wijzigingen en verbeteringen aan te brengen. Vergeet echter niet dat ongeacht of de software een propriëtaire of open-source licentie heeft, het niet naleven van de licentie een overtreding van de wet op de auteursrechten inhoudt.

Recente inzichten uit de economieliteratuur

Kort samengevat is het traditionele idee achter IER, vanuit economisch oogpunt, het beschermen van de prikkels voor innovatie door kopiëren en imitatie uit te sluiten. Dit kan leiden tot de indruk dat OSS ervoor kan zorgen dat innovatieprocessen vastlopen. Zoals we hier zullen beargumenteren, kan dat in sommige situaties het geval zijn. Het plaatje kan er echter ook volledig anders uitzien. Wij wijzen u in dit verband op het feit dat economen, onder wie Nobelprijswinnaars Joseph Stiglitz en Eric Maskin, nieuwe inzichten toevoegen aan de

¹¹ Voor een meer formele definitie, zie de definitie zoals voorgesteld door Bruce Perens op <http://www.opensource.org/docs/osd>. Zie ook Groenenboom (2002).

¹² McGowan (2005).

traditionele opvattingen. Zoals Stiglitz (te verschijnen) beschrijft in een onderzoeksartikel over de economische basis van intellectuele eigendomsrechten: “De [...] intentie [van het intellectuele eigendommodel] is het bieden van prikkels voor innovatie door innovatoren in de gelegenheid te stellen het gebruik van de kennis die zij produceren te beperken door het in rekening brengen van kosten voor gebruik van die kennis toe te staan, waardoor zij rendement uit hun investering kunnen behalen” (p. 104). Desondanks kan men beargumenteren dat “[...] vele van de belangrijkste intellectuele ontwikkelingen in het geheel niet worden gedekt door het patentsysteem” (p. 105). Bessen en Maskin (2004) doen een gelijksoortige observatie met betrekking tot de groei van het internet: “[...] het economische model dat ten grondslag ligt aan deze traditionele redenering is verrassend beperkt. [Veel] creatieve activiteit is niet het werk van eenzame creatievelingen. Het is eerder interactief en bevat bijdragen van vele verschillende partijen. Innovatie is zelfs vaak sequentieel van aard, waarbij iedere ontwikkelaar voortbouwt op het werk van de voorgaande iteratie” (p. 2). Het basisprincipe van deze observatie is dat in het standaardmodel kopiëren en imiteren even “slecht” zijn. Wanneer het innovatieproces echter interactief en sequentieel is, is imitatie niet hetzelfde als kopiëren, maar helpt het uiteindelijk waarde voor de samenleving te creëren.

3 Kenmerken van open source- en propriëtaire software

Om te kunnen beoordelen of er een interveniërende rol is weggelegd voor de overheid beschrijven we in dit gedeelte de belangrijkste kenmerken van softwaremarkten die van invloed kunnen zijn op het concurrentie- en innovatieniveau. We vergelijken bedrijfsmodellen voor propriëtaire en open source software om te begrijpen (1) wat de sturende factor is achter softwareontwikkelaars, (2) welke soorten marktfalen er kunnen zijn en (3) hoe deze twee kenmerken van invloed zijn op de welvaart bij afwezigheid van overheidsinterventie.

3.1 Prikkel voor ontwikkeling open source software

In deze paragraaf zoeken we naar antwoord op de eerste vraag: welke prikkels bestaan er onder de verschillende licentiemodellen om software te ontwikkelen? We beschrijven eerst de prikkels voor individuele ontwikkelaars om deel te nemen aan het ontwikkelen van OSS zonder daarvoor te worden betaald. Daarna bespreken we waarom op winst georiënteerde bedrijven OSS zouden willen ontwikkelen.¹³

Propriëtaire software wordt soms gezien als het gebied waarop particuliere bedrijven proberen winst te maken, terwijl open source software soms wordt gezien als het werk van afzonderlijke programmeurs, of van teams van individuen die vrijwillig deelnemen in open source-communities zonder daarvoor te worden betaald. Voorbeelden van particuliere bedrijven die overbekende propriëtaire programma's ontwikkelen en verkopen zijn: Microsoft, Oracle, Sun Micro Systems en Apple. Een zeer bekende ontwikkelaar van open source software is Linus Torvalds, de initiatiefnemer van het Linux-besturingsplatform. Dit platform is een voorbeeld van een community van softwareontwikkelaars waarin individuen bijdragen aan de ontwikkeling van een bepaald programma (zoals Apache en Linux).¹⁴

Particuliere bedrijven maken de broncode soms door middel van een open source licentie openbaar.¹⁵ Het is zelfs zo dat ongeveer de helft van alle mensen die aan open source software heeft gewerkt, betaald werd en voor hun bijdrage aan deze projecten (Ghosh et al., 2002). Dat impliceert dat bedrijven hen direct of indirect ondersteunen. Soms initiëren bedrijven actief de ontwikkeling van OSS, zoals Sun Microsystems met Open Office. Natuurlijk kan een afzonderlijke softwareontwikkelaar ook primair worden gedreven door winstmotieven (bijvoorbeeld als ondernemer die zich richt op het ontwikkelen en verkopen van software onder de bescherming van propriëtaire auteursrechten).

¹³ Dit gedeelte is gebaseerd op Lerner en Tirole (2005a en 2005b), Maurer en Scotchmer (2006) en Mendys-Kamphorst (2002).

¹⁴ Zie Von Hippel en Von Krogh (2003).

¹⁵ Zie Lerner et al. (2006).

3.1.1 **Waarom dragen afzonderlijke softwareontwikkelaars vrijwillig bij aan OSS-projecten?**

Waarom dragen afzonderlijke softwareontwikkelaars vrijwillig bij aan OSS-projecten, zonder daarvoor direct te worden betaald? Klaarblijkelijk spelen ook andere motieven dan geld een belangrijke rol. Om het verband tussen OSS en innovatieprikkel te begrijpen, is het belangrijk nauwkeurig aan te kunnen wijzen wat individuele ontwikkelaars, die niet bij bedrijven in dienst zijn, motiveert.

In dit gehele document nemen we het traditionele economische standpunt in dat individuele ontwikkelaars rationeel zijn en gericht op het maximaliseren van hun nut. Dit is ook van toepassing op hun beslissing om software te ontwikkelen. Wanneer een programmeur deelneemt aan de ontwikkeling van software krijgt hij te maken met verschillende voordelen en kosten.¹⁶ Hij besluit om aan een project deel te nemen als de voordelen opwegen tegen de kosten. Deze aanname impliceert overigens niet dat alleen financiële of materiële motieven van belang zijn: nut kan zowel extrinsieke als intrinsieke componenten hebben. Wanneer iemands acties door extrinsieke motieven worden gemotiveerd, dan heeft hij of zij vaak een externe prikkel nodig, meestal in de vorm van een financiële vergoeding, om deze activiteiten uit te voeren. Een persoon die intrinsiek wordt gemotiveerd, ontleent dit nut echter aan de activiteit zelf en heeft een dergelijke externe prikkel niet nodig.

Softwareontwikkelaars van propriëtaire projecten ontvangen *directe betalingen* in de vorm van salaris of beloningen, afhankelijk van voltooiing of prestaties. Echter, programmeurs die aan open source projecten werken, ontvangen noodzakelijkerwijs zulke directe betalingen niet. Bovendien beschermen ze hun bijdragen normaal gesproken onder een licentie die het moeilijk maakt door verkoop voordeel te halen uit het resulterende product. Daarom moeten OSS-ontwikkelaars door andere prikkels dan financiële vergoedingen worden gemotiveerd om zich in te spannen. Deze prikkels kunnen zowel intrinsiek als extrinsiek van aard zijn.

Er zijn diverse empirische onderzoeken (Alexy en Leitner, 2008, Hann et al., 2004 en Mockus et al., 2002 over het Apache-project, Ghosh et al., 2002, Lakhani en Wolf, 2005, Henkel, 2005 en Van Wendel, 2005) en theoretische verhandelingen (bijvoorbeeld Lerner en Tirole, 2005a) waarin wordt onderzocht wat softwareontwikkelaars motiveert om aan open source-projecten bij te dragen. Deze onderzoeken laten zien dat ontwikkelaars voornamelijk deelnemen aan open source-projecten als gevolg van extrinsieke motieven. Een significant deel van de deelnemers gaf echter enkele intrinsieke elementen aan als motief. Er is tot dusver nog weinig onderzoek gedaan naar het relatieve belang van intrinsieke motieven in vergelijking met extrinsieke motieven.

De belangrijkste *extrinsieke motieven* die in deze studies zijn besproken, zijn: (i) voordelen voor eigen gebruik, (ii) demonstreren van de eigen kwaliteiten op de arbeidsmarkt ('erkenning door gelijken' en 'verbetering professionele status') en (iii) zelfeducatie ('verbetering vaardigheden').

¹⁶ De kosten voor de programmeur bevatten o.a. de alternatieve kosten voor tijd, d.w.z. de waarde van het beste alternatief. Denk hierbij bijvoorbeeld aan gemiste financiële compensatie uit een andere baan of de waarde van vrije tijd.

Hierbij dient opgemerkt dat (ii) en (iii) ook motieven kunnen zijn voor programmeurs die aan propriëtaire software werken en algemeen betrekking hebben op toekomstige financiële beloning. Lerner en Tirole (2002, 2005a) stellen dat hoe sterker deze prikkels voor de lange termijn zijn, (a) des te zichtbaarder het product is voor desbetreffende doelgroep (gelijken, potentiële werkgevers en gebruikers), (b) des te hoger de invloed van de inspanningen is op de prestatie, (c) des te meer de prestatie iets zegt over het talent en (d) des te groter de netwerkeffecten van programmeurs zijn.

Intrinsieke motieven zijn sociaal-psychologisch van aard en hebben geen betrekking op financiële beloningen. Altruïsme, plezier, meritocratie en ideologie (wederkerigheid, voordelen van schenken) spelen vaak een belangrijke rol wanneer een programmeur besluit deel te nemen aan de ontwikkeling van open source software. Ook betaalde softwareontwikkelaars hebben intrinsieke motieven kunnen hebben, maar aantoonbaar minder dan onbetaalde ontwikkelaars van OSS.

3.1.2 Waarom ontwikkelen bedrijven OSS?

In onderstaande uiteenzetting (en in dit gehele document) nemen we aan dat particuliere bedrijven hun winst willen maximaliseren. Algemeen kan een bedrijf dat software ontwikkelt, inkomsten genereren door verkoop van de software of daaraan gerelateerde complementaire producten en diensten. Een bedrijf krijgt te maken met kosten voor softwareontwikkeling, die hoog of laag kunnen zijn. Hoge ontwikkelingskosten kunnen bijvoorbeeld ontstaan als voor de ontwikkeling, naast programmeervaardigheden, speciale inbreng vereist is, zoals hoogkwalitatief menselijk kapitaal dat tegen een hoge prijs moet worden ingekocht. Nadat de initiële kosten zijn gemaakt, zijn de incrementele kosten voor productie en verkoop van kopieën van een programma te verwaarlozen.

Er zijn vele voorbeelden van winstgeoriënteerde bedrijven die OSS ontwikkelen. Een bedrijf zal uiteraard alleen een open source project initiëren of eraan deelnemen als deze verwacht daaruit winst te maken. Daarom moeten de (verwachte) financiële voordelen van het ontwikkelen van de broncode onder een open source licentie opwegen tegen de kosten van het afzien van inkomsten die anders zouden worden gegenereerd uit de verkoop van software onder een propriëtaire licentie. Om deze voorwaarde te kunnen nakomen, moeten door bedrijven geïnitieerde open source projecten aan twee vereisten voldoen. Allereerst moeten de kansen op succesvolle implementatie van open source groot genoeg zijn. Het succes van een open source project hangt af van het vermogen van het project contribuanten aan te trekken. Niet alle soorten software zijn even geschikt voor implementatie als open source. Het is dus niet altijd verstandig om 'open source te gaan'. Ten tweede moet een bedrijf er geld mee kunnen verdienen. Twee belangrijke vragen zijn daarom:

1. Welke soorten projecten zijn geschikt voor implementatie als open source?
2. Hoe kan OSS inkomsten genereren?

Welke soorten projecten zijn geschikt voor implementatie als open source?

Particuliere bedrijven die open source software ontwikkelen, lopen tegen diverse problemen aan met betrekking tot succesvolle implementatie.

1. Een eerste vereiste is dat het mogelijk moet zijn om het project als open source project te implementeren. Zoals uitgelegd in het kader 'Consequenties van de aard van de motieven van ontwikkelaars voor de organisatie van OSS-projecten' zijn algemene kenmerken van succesvolle open source-projecten modulariteit en leiderschap. Merk op dat binnen de setting van een bedrijf de leiderschapsvoorwaarde een minder groot probleem zal zijn dan in meer 'organische' omgevingen. Een andere opmerking betreft de kwaliteit van software. Het is moeilijk ondubbelzinnige uitspraken te doen over de kwaliteit van OSS in vergelijking met PS, omdat er geen consensus bestaat over deze kwestie. Aan de ene kant kan de kwaliteit van software hoger zijn als deze is ontwikkeld onder centrale leiding, zoals bij PS het geval is. Dit soort leiding ontbreekt soms in het geval van OSS, wanneer een ontwikkelaar zelf de verantwoordelijkheid neemt voor zijn eigen subproject.¹⁷ Aan de andere kant kan worden gesteld dat een hiërarchische organisatie het nooit beter kan doen dan een netwerk van ontwikkelaars.¹⁸
2. Ten tweede moeten bedrijven in staat zijn te garanderen dat de broncode openbaar blijft, in plaats van deze om te zetten naar een propriëitair systeem als het project een succes is en de open source software door een groot aantal gebruikers wordt overgenomen. Eén mogelijke oplossing is dat het bedrijf het bestuur over het open source-project overdraagt aan externe leiders (Maurer en Scotchmer, 2006). Daardoor kan, als het particuliere bedrijf uit het project stapt, dezelfde of een andere open source-community in principe doorgaan met de ontwikkeling van de software. Dit garantieprobleem kan ook worden omzeild door de juiste keuze van een open source licentie, bijvoorbeeld als deze inhoudt dat wanneer een project open source, is het niet meer uit het open source domein gehaald kan worden (zoals het geval is bij GPL, General Public License). Een voorbeeld hiervan is Netscape's initiatief voor het Mozilla-project,¹⁹ door een deel van de broncode van haar browser (die in eerste instantie in eigendom was) vrij beschikbaar te stellen aan de Mozilla End-User License Agreement. Netscape initieerde dit project toen duidelijk werd dat het de strijd met Microsofts webbrowser had verloren. Dit project was vanaf het begin al een probleem, wat, zoals Lerner en Tirole (2002) beargumenteren, het gevolg kan zijn geweest van twijfels over de betrokkenheid van Netscape bij het open source karakter van het project (de oorspronkelijke licentie stond afgeleide propriëtaire werken toe) en de onvoldoende modulaire aard van het project. Mozilla gaf de broncode uiteindelijk

¹⁷ Voor meer informatie over de onderliggende complexiteit, zie bijvoorbeeld Lerner en Tirole (2002). Er zijn ook voorbeelden van het tegendeel, waarbij software van hoge kwaliteit is ontwikkeld onder centrale leiding, zoals het geval was bij Linux, Mozilla's Firefox of Open Office van Sun Microsystems.

¹⁸ Een voorbeeld hiervan is te vinden bij Raymond (1999) die beargumenteert: "als velen meekijken, is iedere programmeerfout triviaal".

¹⁹ Mozilla was vroeger de codenaam van Netscape. Het is nu de merknaam van de Mozilla Corporation die diverse toepassingen uitgeeft onder open source-licenties.

vrij onder licenties die iedereen toestonden om er wijzigingen in aan te brengen. Uiteindelijk leidde het project tot de open source webbrowser Mozilla Firefox, die op dit moment de op één na populairste internetbrowser is.

Consequenties van de aard van de motieven van ontwikkelaars voor de organisatie van OSS-projecten

Het succes van een open source project zoals Linux hangt af van het vermogen van het project contribuanten aan te trekken en uiteindelijk het product op de markt te brengen. De organisatorische structuur van een open source-project wordt daarom bepaald door de motieven van de individuen om deel te nemen en bij te dragen aan het project. Twee belangrijke kenmerken van de organisatie van open source projecten die vaak worden genoemd, zijn modulariteit en leiderschap.

Modulariteit heeft betrekking op de manier waarop een softwareontwikkelingsproject is georganiseerd. Het vereist dat het project kan worden onderverdeeld in afzonderlijke modules, waaraan programmeurs onafhankelijk van elkaar kunnen werken. Een modulaire organisatie heeft een tweeledig effect. Ten eerste geeft het individuele contribuanten de kans om hun capaciteiten effectiever te demonstreren. Door de modulaire organisatie van een project is het makkelijker voor buitenstaanders om te herkennen wie wat heeft bijgedragen aan een project en hoe belangrijk die bijdrage was. Ten tweede zijn er bij een modulaire projectopzet minder coördinatieproblemen, die wel optreden bij de ontwikkeling van open source. Gebrek aan coördinatie tussen programmeurs kan bijvoorbeeld leiden tot verspilling van tijd door dubbel werk. Het definiëren van modules is een effectieve manier om de verschillende bijdragen aan een OSS-project te coördineren, door transparant te maken wie aan welke module werkt. Zoals Giuri et al. (2008) aanvoeren, moet een hogere mate van modulariteit echter samengaan met meer en effectievere leiders die het project aansturen.

Leiderschap is relevant aan het begin van een open source project en wanneer het oorspronkelijke idee op het punt staat te worden gelanceerd. Een leider moet ten eerste softwareontwikkelaars overhalen om deel te nemen aan het project. Er wordt wel gezegd dat, wil een open source-project succesvol zijn, het voldoende uitdagende programmeerproblemen moet bieden waarmee het softwareontwikkelaars aantrekt (Lerner en Tirole, 2002). Ten tweede kan het bestuur van de community, zelfs wanneer open source-projecten zijn gemodulariseerd, even complex zijn als van een commercieel bedrijf. Ten slotte is een leider belangrijk bij het op de markt brengen van software, met name omdat OSS-ontwikkelaars niet noodzakelijkerwijs worden gemotiveerd door de behoeften van de gemiddelde consument. De kans dat projecten succesvol zijn, is daarom groter als iemand de verantwoordelijkheid neemt voor het verzamelen van informatie, beslist welke bijdrage relevant is en hoe het eindproduct op de markt moet worden gebracht.

Leiderschap kan op verschillende manieren worden gerealiseerd. Linux had bijvoorbeeld baat bij een onbetwiste leider die het project initieerde en die zijn autoriteit sterk gecentraliseerd hield. In andere gevallen, zoals bij Apache, neemt een commissie de beslissingen op basis van een consensusmechanisme (Lerner en Tirole, 2002). Debian, een besturingssysteem dat gebaseerd is op Linux-code en één van de oudste open source-projecten is, werd oorspronkelijk bestuurd door een sterk 'visionair' leider en ondersteund door een community. Daarna verschoof de organisatiestructuur ervan naar een informeel controlemechanisme gebaseerd op statuten, een ieder jaar opnieuw gekozen projectleider, ontwikkelaars (die samen de 'kern' vormen die verantwoordelijk is voor exploratie gericht op innovatie en ontwikkeling) en de onderhouders (de 'periferie', verantwoordelijk voor de exploitatie van de bestaande vaardigheden) (Mateos-Garcia en Steinmueller, 2008).

3. Ten derde zijn voor bepaalde soorten software complementaire vaardigheden nodig, die niet direct of helemaal niet aanwezig zijn bij softwareontwikkelaars. Als het kostbaar is om deze vaardigheden in te kopen, is een open source licentie misschien niet geschikt vanwege het van tevoren benodigde kapitaal. Voorbeelden van dergelijke software zijn games (waarvoor geavanceerd grafisch werk nodig is) en gespecialiseerde programma's, zoals ERP-programma's (bijvoorbeeld SAP, dat gebruik maakt van specifieke kennis op het gebied van accountancy en bedrijfsadministratie en tevens gebruikersondersteuning van goed opgeleide professionals). Als software een dergelijke specifieke component bevat, is de kans groot dat de ontwikkelingskosten aanzienlijk zijn en financiële compensatie onvermijdelijk is. In dat geval is een propriëtaire licentie doeltreffender om een project van de grond te krijgen.

We merken op dat dergelijke problemen meestal worden opgelost door marktspelers die diverse organisatorische methodes en licentievormen ter beschikking hebben die open source ontwikkeling ondersteunen. Bovendien speelt het derde probleem, namelijk dat van tevoren kapitaal benodigd is wanneer kostbare, specifieke inbreng vereist is voor de ontwikkeling van een programma, een grote rol. Dat komt aan de orde wanneer we op een later tijdstip de beleidsimplicaties bespreken.

Hoe kan OSS inkomsten genereren?

Bedrijven die software ontwikkelen onder een propriëtaire licentie, ontlene 'marktmacht' aan de wet op de auteursrechten, omdat ze het exclusieve recht hebben hun producten op de markt te brengen door hun auteursrechten te verkopen of hun producten onder licentie uit te brengen voor gebruik door anderen.^{20 21} In tegenstelling hiermee ontlene bedrijven onder open source licenties geen marktmacht aan hun licentie omdat iedereen het product kosteloos mag gebruiken, verbeteren en distribueren. Bedrijven met winstoogmerk zullen daarom alleen in OSS investeren als dit om andere redenen winstgevend is of als ze marktmacht ontlene aan andere bronnen.

Een mogelijk bedrijfsmodel voor commerciële bedrijven is wat Lerner en Tirole (2005a) 'symbiotisch leven' noemen. Laten we aannemen dat het open source product inkomsten genereert uit bepaalde complementaire propriëtaire activiteiten (producten of diensten). Als de extra voordelen van deze complementaire activiteiten opwegen tegen de kosten van het uitbrengen van de software als open source, dan kan het vrijgeven van de broncode een aantrekkelijke strategie zijn. Complementaire activiteiten kunnen bijvoorbeeld technische ondersteuning, opleiding, training, als wel de verhoogde verkoop van complementaire en

²⁰ Ten behoeve van dit document gebruiken we de economische definitie van marktmacht, dat wil zeggen dat marktmacht wordt gedefinieerd als het vermogen van bedrijven de prijs aan te passen boven de marginale kosten (of de mark-up te verhogen). Let op het verschil met de juridische interpretaties van marktmacht.

²¹ De feitelijke omvang van marktmacht hangt af van diverse marktkenmerken, zoals netwerkexternaliteiten, overstapkosten en schaalvoordelen. Zie gedeelte 3.2.

propriëtaire software zijn.²² Particuliere bedrijven moeten vrijwillige contribuanten overtuigen dat het project niet eenvoudigweg wordt gedumpt in het open source-domein als de aanvullende activiteiten een commerciële mislukking blijken te zijn. Een manier om deelnemers daarvan te overtuigen is door voldoende investeringen te doen in de ontwikkeling van open source.

Soms schakelen bedrijven over van de ontwikkeling van propriëtaire software naar open source software. Bedrijven kunnen tot zo'n omschakeling besluiten als de kosten van omschakeling laag zijn, maar de mogelijke opbrengsten hoog. Dit is het geval wanneer een bedrijf achterloopt op een marktleider van propriëtaire software, maar de mogelijkheid ziet dat het product als OSS succesvol zal zijn en daarbij verhoogde opbrengsten verwacht in de complementaire segmenten (Lerner en Tirole, 2005a). Open source ontwikkeling kan opwegen tegen een dominante positie van een concurrent (concurrentie-effecten), of kan de opbrengsten uit verbeterde reputatie en aanwezigheid op de lange termijn verhogen (Henkel, 2005). Het geven van een garantiesignaal stelt mogelijke gebruikers ook gerust dat het product niet zal worden teruggetrokken. Een voorbeeld is het hiervoor genoemde Mozilla-project. Een andere reden is dat de eerste ontwikkelingskosten van OSS ook lager kunnen zijn door de niet-financiële motieven van werknemers.

Een andere bron van inkomsten kan voortkomen uit de verminderde ontwikkelingskosten, bijvoorbeeld door externe ontwikkelingsondersteuning, zoals onderhoud van de software door OSS-communities (kennispillovers, zie in sectie 3.2) en debugging.

3.1.3 Samenvatting

In dit gedeelte hebben we onze eerste vraag beantwoord, namelijk wat softwareontwikkelaars motiveert onder verschillende licentiemodellen. De motieven om software te ontwikkelen onder een propriëtaire licentie, verschillen over het algemeen van die onder een open source-licentie. Propriëtaire licenties creëren voor softwareontwikkelaars de mogelijkheid om direct marktmacht te krijgen. Hun motieven voor het ontwikkelen van software komen hoofdzakelijk voort uit de verwachte opbrengsten uit de verkoop van hun software. Onder open source licenties blijven ontwikkelaars echter verstoken van de mogelijkheden voor het verwerven van marktmacht die IER-systemen bieden. Als gevolg daarvan komen de verwachte opbrengsten niet zozeer (of helemaal niet, in het geval van gratis software) uit de verkoopopbrengsten, maar ergens anders vandaan. Bedrijven kunnen alternatieve financiële opbrengsten genereren door complementaire producten en diensten, zoals ondersteuning, premiumversies, adviesdiensten of het aantrekken en vasthouden van waardevolle werknemers. Bovendien worden ontwikkelaars die vrijwillig bijdragen aan open source projecten, gedreven door intrinsieke, niet-financiële motieven zoals altruïsme, plezier, meritocratie en ideologie, of door extrinsieke, vaak financiële motieven op de lange termijn, zoals eigen gebruik, zelfeducatie en het demonstreren van de eigen kwaliteiten op de arbeidsmarkt.

²² Verhoogde verkoop van complementaire en propriëtaire producten kan worden veroorzaakt door netwerkeffecten (zie sectie 3.2.1), maar ook door een grotere bereidheid van de consument om de propriëtaire software toe te passen.

3.2 Software-gerelateerde vormen van marktfalen

Onze tweede vraag was welke soorten marktfalen aanwezig zijn in softwaremarkten. Als er vormen van marktfalen bestaan die niet worden gecorrigeerd, zullen marktprocessen normaal gesproken niet resulteren in voor de welvaart optimale keuzes van licentietype, prijzen, kwaliteit en innovatieniveau. Voor beleidsmakers is het daarom belangrijk de mogelijke vormen van marktfalen in softwaremarkten, en de manieren waarop ze op de korte en lange termijn negatief uitwerken op de welvaart, te begrijpen. Softwaremarkten vertonen over het algemeen de volgende vormen van marktfalen: marktmacht als gevolg van *customer lock-in* en marktmacht als gevolg van schaalvoordelen en kennisspillovers.²³ Hierbij moet worden opgemerkt dat wanneer een product naam heeft gemaakt, het bedrijf dat de propriëtaire licentie in bezit heeft, marktmacht heeft die hem wordt ‘verleend’ door het IER-systeem.^{24 25}

3.2.1 Marktmacht als gevolg van *customer lock-in*

Als eerste bekijken we marktmacht die wordt gecreëerd door *customer lock-in*. In softwaremarkten komt *customer lock-in* met name voort uit netwerkeexternaliteiten en overstapkosten.

Netwerkeexternaliteiten

Customer lock-in kan worden veroorzaakt door netwerkeexternaliteiten. Netwerkeffecten ontstaan wanneer de baat die klanten hebben bij het gebruik van de software, hoger wordt naarmate het aantal andere klanten toeneemt die dezelfde of compatibele software gebruiken. Bovendien kan de baat die een klant hierbij heeft, toenemen in verhouding met het aantal complementaire toepassingen. Netwerkeexternaliteiten bestaan als klanten, wanneer ze kiezen tussen verschillende technologieën, geen rekening houden met deze netwerkeffecten. Dat wil zeggen dat deze netwerkeffecten niet worden geïnternaliseerd.

Netwerkeffecten kunnen zowel aan de vraagzijde aanwezig zijn als aan de aanbodzijde van de softwaremarkt.

- Aan de *vraagzijde* van softwaremarkten kunnen we onderscheid maken tussen twee soorten netwerkeffecten. Ten eerste ontstaan *directe netwerkeffecten* wanneer klanten een bepaald softwareprogramma meer waarderen naarmate het aantal andere gebruikers groter is. De baat

²³ Vormen van marktfalen beschreven in dit gedeelte zijn gebaseerd op Farrell en Klemperer (2007), Schmidt en Schnitzer (2003) en Geist et al. (2000).

²⁴ Zie bijvoorbeeld Maurer en Scotchmer (2006).

²⁵ Er kan nog een andere oorzaak van marktmacht worden toegevoegd, namelijk de aanwezigheid van ‘toetredingsbarrières’. Er is bijvoorbeeld sprake van dergelijke toetredingsbarrières als niemand kan concurreren met een bestaand besturingssysteem zonder dat er voldoende toepassingen op draaien. Tot bepaalde hoogte kan deze situatie worden gezien als een geval van indirecte netwerkeffecten, of ook als tweezijdige markt (zie bijvoorbeeld Rochet en Tirole, 2006). Dit kan nog verder worden doorgetrokken door op te merken dat een besturingssysteem in staat moet zijn zo goed als ieder softwareprogramma op ieder stuk hardware te draaien. De analyse van dit probleem valt buiten het kader van deze studie; zie Larouche (2008) voor meer informatie over de Europese rechtzaak van Microsoft.

die klanten hebben bij het gebruik van een bepaald softwareprogramma, wordt hoger naarmate het aantal andere consumenten die dezelfde of compatibele software gebruiken toeneemt.

Voorbeelden zijn tekstverwerkers zoals Word. Ten tweede kunnen consumenten een bepaald softwareprogramma hoger waarderen wanneer een groot aantal complementaire toepassingen beschikbaar is. Deze effecten worden over het algemeen *indirecte netwerkeffecten* genoemd.

Voorbeelden van indirecte netwerkeffecten zijn besturingsplatforms zoals Windows en Linux: hoe meer complementaire software er beschikbaar is voor een platform, des te aantrekkelijker het voor consumenten wordt om dat platform te gebruiken. De aanwezigheid van netwerkeffecten op de vraagzijde moedigt bestaande consumenten aan hetzelfde product te blijven gebruiken, of nieuwe consumenten producten te kiezen met een groter netwerk (*customer lock-in*), waarmee toetredingsbarrières worden gecreëerd voor nieuwe en mogelijk superieure producten.

- De *aanbodzijde* van softwaremarkten kan ook directe en indirecte netwerkeffecten vertonen. In de ontwikkeling van OSS kunnen *directe netwerkeffecten* ontstaan omdat de aantrekkelijkheid van een open source project voor softwareontwikkelaars afhangt van het aantal andere ontwikkelaars dat hieraan bijdraagt. Intuïtief impliceert een groter aantal contribuanten een duidelijker signaal voor andere potentiële programmeurs. Een groter aantal actieve contribuanten biedt een ontwikkelaar de mogelijkheid om te laten zien dat hij beter presteert dan een grotere groep medeontwikkelaars. Aan de andere kant kan een groter aantal ontwikkelaars een persoon ontmoedigen om een bijdrage te leveren, omdat zijn bijdrage misschien marginaal is. *Indirecte netwerkeffecten* treden ook op aan de productiezijde van de softwaremarkt. De potentiële waarde (voor ontwikkelaars) van het ontwikkelen van nieuwe software die complementair is ten opzichte van andere software, neemt toe met het aantal gebruikers. Hoe meer consumenten bijvoorbeeld een bepaald platform gebruiken, des te voordeliger het is om software te ontwikkelen voor dat platform.

Inefficiënties veroorzaakt door netwerkexternaliteiten

Naast het versterken van marktmacht kunnen netwerkexternaliteiten ook leiden tot verschillende andere inefficiënties (gebaseerd op Farrell en Klemperer, 2007).

De belangrijkste hiervan is *onderaanvaarding* die ontstaat wanneer netwerkeffecten niet worden geïnternaliseerd: wanneer consumenten het feit negeren dat anderen baat hebben bij hun aanvaardingsbeslissingen kan vanuit welvaartsoogpunt het aantal consumenten dat een bepaalde technologie aanvaardt (overneemt) te laag zijn. Overaanvaarding kan ook voorkomen, zie *market tipping* hieronder.

Als indirecte netwerkeffecten niet worden geïnternaliseerd, leidt dit tot *een te laag aanbod van de ontwikkeling van complementaire software*. Een monopolist met een propriëtaire licentie kan overigens netwerkeffecten met betrekking tot complementaire producten tot op zekere hoogte internaliseren. Zijn exclusieve rechten stellen het bedrijf in staat de prijs boven de kostprijs te verhogen in verhouding tot het aantal complementaire producten. Op deze manier kan de monopolist zich de voordelen voor de gebruikers van het verhoogde aantal complementaire toepassingen of verhoogde kwaliteit toe-eigenen.

Netwerkexternaliteiten kunnen *de markt laten overhellen* in de richting van een technologie of standaard, niet noodzakelijkerwijs degene die de grootste voordelen biedt voor gebruikers (web van netwerkeffecten) en een monopoliepositie creëren voor de verkoper van die technologie. De hieruit voortvloeiende hoge monopolieprijs creëert een permanent verlies van welvaart (*deadweight loss of excess burden*), dat is een welvaartsverlies dat ontstaat wanneer de vraag lager wordt als gevolg van een verhoging van de prijs. Het overhellen van de markt kan ook het gevolg zijn van coördinatieproblemen tussen aanvaarders. Welk netwerkproduct consumenten aanvaarden hangt af van welk product zij verwachten dat aanvaard zal worden. Als netwerkproduct A superieur is aan netwerkproduct B, maar iedereen verwacht dat netwerkproduct B zal worden aanvaard, zal het product van mindere kwaliteit worden aanvaard. Wanneer de markt eenmaal overhelt, neigt de vraag in de richting van het toonaangevende product in termen van netwerkeffecten, ongeacht de relatieve productkwaliteit in vergelijking met het aanbod van de concurrent.

Netwerkexternaliteiten kunnen leiden tot *coördinatiefalen*: consumenten kunnen zich bundelen en een inferieure technologie aanvaarden (denk aan de videostandaarden VHS versus Betamax), of het ontbreken van consumentenbundeling kan versplintering veroorzaken. Dat wil zeggen dat ze meerdere concurrerende en incompatibele producten aanvaarden in een markt waar één product beter zou zijn (Lerner en Tirole, 2005a).

Overstapkosten

Consumenten kunnen te maken krijgen met substantiële kosten wanneer ze besluiten over te stappen op een andere technologie, product of leverancier. Overstapkosten kunnen bijvoorbeeld ontstaan door de noodzaak te leren hoe een ander product moet worden gebruikt, door de aanschafkosten van nieuwe compatibele softwareprogramma's of de kosten van het zoeken naar een alternatief product.^{26 27} In sommige gevallen zijn de overstapkosten immaterieel, bijvoorbeeld ten gevolge van de gewenning aan een bepaalde technologie. Zwiebel (1995) formuleert een principe achter dit soort overstapkosten. Hij redeneert dat managers, voornamelijk om redenen van reputatie en door gebrek aan informatie nalaten om over te

²⁶ Voor een uitgebreide analyse van overstapkosten, zie Farrell en Klemperer (2007).

²⁷ Gesloten standaarden in softwaremarkten beperken de uitwisseling van bestanden tussen gebruikers van verschillende toepassingen, waardoor het overstappen op nieuwe software kostbaar wordt. Het probleem van gesloten standaarden is meer uitgesproken wanneer er sprake is van netwerkeffecten.

stappen naar een alternatieve oplossing die superieur is aan de industriestandaard of het product van een dominant bedrijf ('Er is nog nooit iemand ontslagen omdat hij IBM heeft aangeschaft.').

Hoge overstapkosten veroorzaken *customer lock-in*, hetgeen resulteert in marktmacht voor softwareproducenten die het voordeel hebben 'first-mover' te zijn. Hoge overstapkosten kunnen daarom leiden tot hogere prijzen en toetredingsbarrières voor leveranciers van nieuwe softwareproducten.

3.2.2 Marktmacht als gevolg van schaalvoordelen

Softwaremarkten worden vaak gekenmerkt door *schaalvoordelen* omdat de ontwikkeling van de oorspronkelijke 'eerste kopie' van een product vaste kosten met zich meebrengt, terwijl de kosten voor het produceren van volgende kopieën relatief laag of zelfs bijna nul zijn.²⁸ Het is interessant te zien dat dit voor softwaremarkten niet noodzakelijkerwijs impliceert dat productie door één bedrijf de meest efficiënte oplossing is. Zelfs als de kosten van het produceren van een eerste kopie hoog zijn, kunnen deze worden verspreid over deelnemers in een programmeurscommunity. Bovendien kunnen de eerste ontwikkelingskosten verwaarloosbaar zijn wanneer programmeurs bereid zijn inspanningen te verrichten zonder directe betaling, zoals het geval kan zijn bij open source software.

Er is een uitzondering die beperkingen voor het gebruik, wijzigen en distribueren van software zoals opgenomen in de softwarelicentie rechtvaardigt. De ontwikkeling van bepaalde soorten software vereist de aankoop van specifieke complementaire inbreng tegen substantiële kosten, zoals expertise met inzicht in gecompliceerde econometrische methoden (zoals in het geval van geavanceerde statistische software) of grafische ontwerpvaardigheden (zoals in het geval van games). Wanneer dergelijke specifieke inbreng noodzakelijk is, werken propriëtaire licenties beter om de ontwikkelingskosten terug te kunnen verdienen.

3.2.3 Kennispillovers

Ten slotte kan er sprake zijn van marktfalen wanneer er *kennispillovers* aanwezig zijn.

Kennispillovers van innovaties treden op als R&D door het ene bedrijf innovaties door andere bedrijven mogelijk maakt, die succes behalen door 'op de schouders van de reuzen mee te liften'. Dergelijke spillovers kunnen twee soorten innovaties mogelijk maken:

- 'Imiterende spillover': een ietwat gedifferentieerd product dat gebaseerd is, voornamelijk door imitatie, op de originele technologie, en
- 'Creatieve spillover': een vervolginnovatie die een significante uitbreiding vormt op de oorspronkelijke technologie.

Allereerst kunnen andere bedrijven het product van een innovator imiteren zonder de laatstgenoemde schadeloos te stellen voor de R&D-kosten. Als gevolg van dergelijke

²⁸ Shy (2001).

imiterende spillovers wordt het winstniveau van de innovator achteraf gereduceerd. Dat betekent dus dat zijn motief vooraf om te innoveren ook lager zal zijn als imitatie makkelijk is. Ten tweede is het innovatieproces in softwaremarkten vaak complementair en sequentieel en zijn er veel ontwikkelaars bij betrokken die elkaars bijdragen verbeteren. Daarom kan kennis die is ontwikkeld door één bedrijf, andere bedrijven helpen nieuwe innovaties te ontwikkelen die niet direct concurreren met de oorspronkelijke innovatie, bijvoorbeeld omdat ze een nieuwe nichemarkt creëren. Deze vorm van creatieve spillovers verhoogt de sociale waarde van een innovatie. Als een bedrijf geen rekening houdt met de extra voordelen als gevolg van creatieve spillovers tijdens het maken van investeringsbeslissingen, is de investering vanuit sociaal oogpunt misschien te laag. Dit effect is bijzonder uitgesproken wanneer de ontwikkelingskosten vooraf hoog zijn: geen rekening houden met creatieve spillovers kan dan het effect hebben dat een sociaal wenselijke investering niet wordt gedaan omdat deze 'persoonlijk' niet winstgevend is.

Propriëtaire licenties pakken beide vormen van spillovers aan: (i) ze bieden bescherming tegen imitatie van het idee van de innovator en (ii) de innovator kan door licenties meer opbrengsten realiseren uit vervolginnovaties van derden. Bijgevolg garandeert propriëtaire licentiëring, door middel van het vooruitzicht van terugverdienen, de stimulering van ontwikkelingsinspanningen. Het is echter belangrijk te beseffen dat het argument voor vervolginnovaties ervan uitgaat dat een innovator weet hoe hij de prijs voor andere producenten moet vaststellen. Zoals Bessen en Maskin (2006) stellen, kan dit een probleem zijn door de asymmetrie in informatie tussen de oorspronkelijke innovator en potentiële vervolginnovatoren met betrekking tot de toekomstige winst van de laatste groep. Concurrenten die willen voortbouwen op een bestaande innovatie kunnen met name bruikbare ideeën hebben over hoe dergelijke innovaties kunnen worden gerealiseerd en die bij het oorspronkelijke bedrijf niet bekend zijn. In dat geval kunnen strikte intellectuele eigendomsrechten de innovatiesnelheid vertragen. Hieruit volgt dat wanneer deze vorm van asymmetrie belet dat de optimale licentievergoeding kan worden overeengekomen, het sociaal optimale innovatieniveau niet zal worden gerealiseerd.

Het verlies aan welvaart dat optreedt wanneer een innovator niet in staat is opbrengsten te genereren uit vervolginnovaties, hoeft niet op te treden bij open source licenties. Wanneer een softwarelicentie toestaat dat de broncode door anderen wordt gebruikt, kunnen ongehinderd vervolginnovaties worden gedaan. Omschakeling van PS- naar OSS-ontwikkeling kan het innovatieniveau dus verhogen door kennisspillovers mogelijk te maken, onder de voorwaarde dat het innovatieproces plaatsvindt op een interactieve en sequentiële wijze (zie ook Bessen en Maskin, 2004).

3.2.4 Samenvatting

Dit gedeelte beschrijft de relevante soorten marktfalen in softwaremarkten. Marktfalen kan, indien het niet wordt gecorrigeerd, leiden tot marktresultaten die afwijken van de sociaal

optimale mogelijkheden. Beleidsmakers moeten daarom begrijpen welke soorten marktfalen er zijn en hoe deze de welvaart beïnvloeden, en daar rekening mee houden. Softwaremarkten vertonen diverse belangrijke vormen van marktfalen: *customer lock-in* als gevolg van netwerkeffecten of overstapkosten, schaalvoordelen veroorzaakt door de noodzaak specifieke, complementaire inbreng in te kopen tegen aanzienlijke kosten, en kennisspillovers.

Ten eerste kan *customer lock-in* twee oorzaken hebben. De aanwezigheid van netwerkeffecten stimuleert bestaande klanten om hetzelfde product te blijven gebruiken, of nieuwe klanten om een product te kiezen met een groter netwerk. Daarmee worden toetredingsbarrières gecreëerd voor nieuwe, mogelijk superieure producten. Consumenten kunnen ook te maken krijgen met substantiële kosten wanneer ze besluiten over te stappen op een andere technologie, product of leverancier. Hoge overstapkosten kunnen ook leiden tot *customer lock-in*.

Ten tweede kan voor de ontwikkeling van bepaalde soorten software specifieke complementaire inbreng tegen substantiële kosten nodig zijn. Daarvoor dienen voldoende terugverdienmogelijkheden van de gemaakte ontwikkelingskosten te bestaan. In zulke situaties kunnen propriëtaire licenties geschikter zijn om deze prikkels zeker te stellen en werken die daarom beter om innovatieprikkels te beschermen.

Ten slotte kan er sprake zijn van kennisspillovers van innovaties als R&D bij één bedrijf innovatie bij andere bedrijven mogelijk maakt. Als innovatoren geen rekening houden met kennisspillovers kunnen de innovatieniveaus vanuit welvaartsperspectief te laag zijn. Er kan onderscheid worden gemaakt tussen twee soorten kennisspillovers op basis van de oorspronkelijke technologie: 'imiterende spillovers', die iets gedifferentieerde producten creëren en 'creatieve spillovers' die vervolginnovaties creëren. Propriëtaire licenties bieden bescherming tegen imiterende spillovers en garanderen daarmee de terugverdienmogelijkheid van de kostbare ontwikkelingsinspanningen. Propriëtaire licenties kunnen ook mogelijkheden creëren tot het genereren van een surplus uit creatieve spillovers. Als gevolg van de asymmetrische informatie tussen de innovatoren zijn sociaal optimale overeenkomsten tussen de oorspronkelijke en de vervolginnovatoren misschien niet haalbaar. Open source licenties omzeilen dit probleem, waardoor creatieve spillovers mogelijk worden gemaakt en het niveau van vervolginnovaties wordt verhoogd.

3.3 Effecten van innovatieprikkels en marktfalen op de welvaart

Dit gedeelte onderzoekt hoe marktfalen en prikkels voor het ontwikkelen van software de welvaart beïnvloeden. Met betrekking tot marktmacht is er sprake van een mogelijke wisselwerking in softwaremarkten.

3.3.1 De klassieke wisselwerking

De klassieke wisselwerking met betrekking tot de economische grondslagen van intellectueel eigendomsrecht is als volgt:

- Aan de ene kant kan het vooruitzicht op marktmacht noodzakelijk zijn om prikkels te bieden voor innovaties. Met name wanneer softwareontwikkeling hoge kosten met zich meebrengt, zal innovatie alleen plaatsvinden wanneer deze ontwikkelingskosten kunnen worden terugverdiend. Marktmacht creëert de noodzakelijke opbrengsten.
- Aan de andere kant zouden, op het moment dat de innovatie is gerealiseerd, de prijzen op het niveau van de marginale kosten moeten zijn en de toetredingsbarrières zo laag mogelijk, terwijl kennisspillovers zo hoog mogelijk zouden moeten zijn. Zodra de software is ontwikkeld, hebben gebruikers baat bij de prijsstelling op het niveau van de marginale kosten (dat wil zeggen, bijna nul) en naadloos overstappen op andere software, en andere ontwikkelaars hebben er baat bij als de software ongehinderd kan worden gemodificeerd om vervolginnovaties te kunnen realiseren.

De wisselwerking tussen marktmacht aan de ene kant en lage prijzen en toetredingsbarrières of meer vervolginnovaties aan de andere kant, is de overbekende wisselwerking tussen ex-ante prikkels en ex-post efficiëntie (welvaart).

Tot op zekere hoogte komt de discussie over de vraag of open source- of propriëtaire licenties optimaal zijn neer op de wisselwerking zoals hierboven beschreven. Maar er komt meer bij kijken, zoals het werk van Bessen en Maskin (2004, 2006) over interactieve en sequentiële innovatieprocessen laat zien. Over het algemeen is er geen uitgesproken optimum. Aan het ene uiterste kan worden gesteld dat een bepaald niveau van marktmacht ook aanwezig is onder open source licenties en dat dit genoeg is om de ontwikkelingskosten te compenseren. Anderen²⁹ beweren dat intrinsieke motieven, eigen gebruik, zelfeducatie, reputatie en het kunnen demonstreren van de eigen kwaliteiten op de arbeidsmarkt, voldoende prikkels bieden. Binnen deze manier van denken zou software die is ontwikkeld onder propriëtaire licenties, ook worden ontwikkeld onder open source licenties. In dat geval bieden propriëtaire licenties geen voordelen vergeleken met open source en zou open source vanuit welvaartsperspectief de voorkeur hebben. Aan het andere uiterste kan worden gesteld dat onder propriëtaire licenties de innovator de optimale licentievergoeding in rekening kan brengen bij vervolginnovatoren en zodat winstgevende vervolginnovaties kunnen worden gerealiseerd. In dat geval zouden open source licenties over de gehele linie geen enkel voordeel bieden vergeleken met propriëtaire licenties, maar brengen ze eerder sociale kosten met zich mee als bepaalde soorten software niet langer zouden worden ontwikkeld. In dat geval hebben propriëtaire licenties duidelijk de voorkeur.

²⁹ Zie de referenties in het gedeelte over prikkels.

Gebaseerd op bestaande inzichten (die nogal beperkt zijn) is het verstandig aan te nemen dat in beide gezichtspunten een grond van waarheid zit en dat men afhankelijk van de situatie en de kenmerken van de specifieke softwaremarkt voor het ene of juist voor het andere gezichtspunt zou kunnen kiezen. Of nog nauwkeuriger gezegd:

- Bij de afwezigheid van propriëtaire licenties kan het gebeuren dat sommige soorten software niet worden ontwikkeld omdat de kosten te hoog zijn en niet kunnen worden terugverdiend. Dit is met name het geval als voor de ontwikkeling van een bepaald type software vooraf grote investeringen nodig zijn. Denk bijvoorbeeld aan complementaire, specifieke inbreng, waarvoor andere vaardigheden dan voor het ontwikkelen van software vereist zijn, die zorgvuldig wordt ontwikkeld door specialisten buiten het netwerk van de softwareontwikkelaars en die gemakkelijk kan worden gekopieerd. Voor dergelijke softwaretypen kan sprake zijn van substantiële kosten waarvoor financiële vergoeding (in tegenstelling tot ‘immateriële compensatie’) onvermijdelijk is, zodat de beste optie is het als Propriëtaire Software op de markt te brengen. De ontwikkeling van PS wordt gedreven door de verwachte opbrengst, die wordt beïnvloed door de bereidheid van consumenten om te betalen en derhalve door de baat die consumenten ervan verwachten te hebben.^{30 31} Met name wanneer een bedrijf zich richt op het genereren van inkomsten door het realiseren van marktaandeel, moet het software bieden die aantrekkelijk is voor een grote groep ‘onervaren’ gebruikers (die geen professionele programmeurs zijn) en daardoor rekening houden met de behoeften van de gemiddelde gebruiker. Bij de afwezigheid van propriëtaire licenties zouden innovaties, met name in de eerste fase waarin de behoeften van consumenten geen belangrijke rol spelen, voornamelijk worden aangestuurd door intrinsieke motieven (altruïsme, plezier, ideologie) en extrinsieke motieven (eigen gebruik, educatie, demonstreren van ontwikkelingsvaardigheden) van de individuele softwareontwikkelaars.³² Deze soorten prikkels zijn misschien niet voldoende om ontwikkeling aan te sturen wanneer kostbare, specifieke inbreng van specialisten nodig is.
- In de aanwezigheid van propriëtaire licenties kan marktmacht die wordt verkregen uit IER worden versterkt door netwerkexternaliteiten, overstapkosten en schaalvoordelen. Marktmacht kan de welvaart onnodig verzwakken: (i) marktmacht kan leiden tot een significant, permanent verlies van welvaart als gevolg van verminderde vraag door hoge prijzen, (ii) marktmacht kan concurrentiebeperkend gedrag mogelijk maken, met name wanneer overstapkosten en netwerkexternaliteiten toetredingsbarrières creëren voor nieuwkomers in de markt en (iii) wanneer licenties te beperkend zijn, vinden geen sociaal waardevolle vervolginnovaties plaats onder propriëtaire licenties (die onder open source licenties wel zouden plaatsvinden, zie

³⁰ Von Hippel (2002) stelt echter dat het soms kostbaar is voor bedrijven om in te gaan op de heterogene behoeften van consumenten, bijvoorbeeld omdat het moeilijk is de individuele voorkeuren van de consumenten te weten te komen of omdat het onmogelijk is prijzen te differentiëren.

³¹ Natuurlijk kunnen de prikkels van PS ook aanwezig zijn in combinatie met enkele van de voordelen van OSS, met name wanneer er informele of afgedwongen normen zijn om de interoperabiliteit en de compatibiliteit tussen toepassingen en bestandsformaten zeker te stellen.

³² Zie Maurer en Scotchmer (2006).

Bessen en Maskin, 2006). Daarom zouden open source-licenties resulteren in hogere sociale welvaart in softwaremarkten waar geen noodzaak bestaat tot het inkopen van kostbare specifieke inbreng van specialisten buiten de softwarecommunity. Omdat er echter minder aandacht kan zijn voor de behoeften van klanten onder open source-ontwikkeling, kunnen deze licenties resulteren in onvoldoende documentatie,^{fn} onvoldoende gebruikersvriendelijkheid of slechte integratie tussen softwareproducten.^{fn} Dit kan bijvoorbeeld gebeuren in de beginfase van de ontwikkeling, wanneer programmeurs (nog) niet geïnteresseerd zijn in de commerciële mogelijkheden van hun software. Tegelijkertijd moet worden opgemerkt dat er vele voorbeelden zijn van meer volgroeide OSS-producten die zeer goed presteren met betrekking tot de wensen van klanten.

3.3.2 Samenvatting

In dit gedeelte hebben we geanalyseerd hoe marktkenmerken zoals marktfalen en prikkels de welvaart beïnvloeden. In een markt met marktmacht kan het gebeuren dat nieuwe toetreders als gevolg van *customer lock-in* te weinig kansen hebben om toe te treden, zelfs als ze een superieur product bieden tegen een lagere prijs. Vanuit welvaartsperspectief kan marktmacht echter nodig zijn en kunnen propriëtaire modellen de voorkeur hebben boven open source-modellen als specifieke inbreng (zoals vaardigheden anders dan programmeervaardigheden) vereist is, waardoor de ontwikkelingskosten hoog zijn, of als er geen obstakels (zoals informatieasymetrieën) zijn met betrekking tot het contracteren van vervolginnovaties, of beide. Als er voor softwareontwikkeling echter voornamelijk programmeervaardigheden nodig zijn en het contracteren van afgeleide werken inefficiënt is, terwijl een continue stroom van vervolginnovaties sociaal wenselijk is, dan presteren open source modellen beter dan propriëtaire modellen.

4 De mogelijke rol van de overheid

Over het algemeen leidt marktfalen tot inefficiënte marktresultaten (bijvoorbeeld te hoge prijzen, consumenten die niet kunnen overstappen, een te lage kwaliteit of een suboptimaal innovatieniveau). In sectie 3.2 hebben we een centrale rol toegewezen aan de volgende vormen van marktfalen: *customer lock-in* als gevolg van netwerkexternaliteiten, schaalvoordelen veroorzaakt door de noodzaak specifieke, complementaire inbreng in te kopen tegen aanzienlijke kosten en kennispillovers met betrekking tot de mogelijkheden voor het contracteren van vervolginnovaties.

Om deze vormen van marktfalen te corrigeren kan beleidsinterventie wenselijk zijn. De vraag in dit hoofdstuk is dan ook wat beleidsmakers kunnen doen om markten beter te laten werken en de welvaart in softwaremarkten te verhogen. We nemen aan dat de overheid optreedt in het belang van het publiek, dat wil zeggen dat het gericht is op het maximaliseren van de welvaart. We definiëren welvaart zowel als welvaart op de korte termijn (statische efficiëntie) en als welvaart op de lange termijn (dynamische efficiëntie). We bespreken tevens de verschillende soorten overheidsfalen die kunnen optreden, zowel doordat de overheid fouten maakt en niet in staat is zich vast te leggen of omdat additionele doelstellingen een rol spelen (wat resulteert in *regulatory capture* (innige banden tussen de sector en de overheid) en het nastreven van winst).

Om marktfalen aan te pakken en de welvaart te verhogen zijn diverse beleidsopties beschikbaar: het stimuleren van de vraag, het stimuleren van het aanbod en het verlagen van toetredingsbarrières. Zoals we later in dit gedeelte zullen zien, kan dit op verschillende manieren worden gedaan. Om vast te stellen welke rol de overheid zou kunnen spelen bij de stimulering van softwareontwikkeling onder een open source-licentiemodel, moeten we diverse vragen beantwoorden:

1. Afhankelijk van de conclusie uit hoofdstuk 3 (i) hoe beïnvloedt het stimuleren van OSS de welvaart op de korte en op de lange termijn, (ii) wanneer is het stimuleren van OSS wenselijk en (iii) wat zijn de beleidsopties als interventie nodig is? Welke
2. Welke negatieve neveneffecten kunnen optreden als gevolg van beleidsinterventie (bijvoorbeeld wat zijn de mogelijke vormen van overheidsfalen)?

4.1 Beleidsimplicaties

Of interventie wenselijk is hangt af van welk licentiemodel eventueel de voorkeur heeft vanuit het gezichtspunt van de welvaart en van welke marktresultaten verwacht kan worden dat ze optreden in het geval van afwezigheid van overheidsinterventie.³³

³³ Gebaseerd op Schmidt en Schnitzer (2003) en Lerner en Tirole (2005a).

Zoals we eerder hebben gesteld, is het argument voor overheidsinterventie in de softwaremarkt zeer subtiel. Om beleidsaanbevelingen te kunnen afleiden moeten we overwegen of het stimuleren van OSS de welvaart op de korte en de lange termijn versterkt of verzwakt. Deze welvaartconcepten worden als volgt gedefinieerd.³⁴

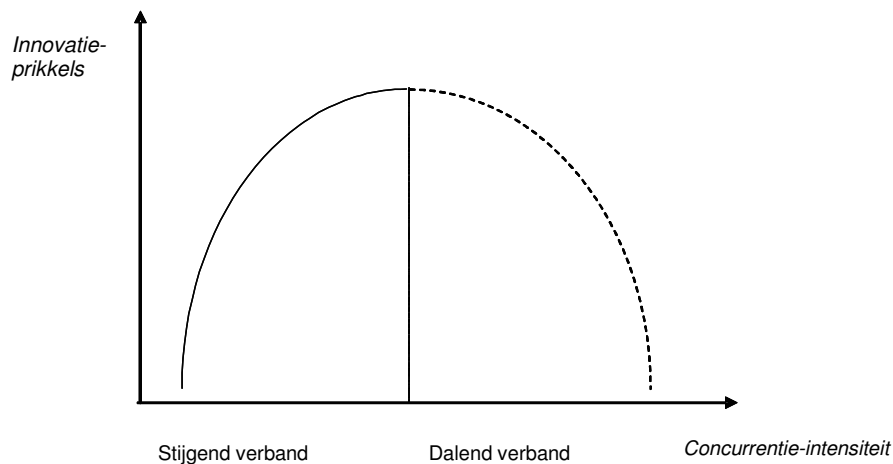
Welvaart op de korte termijn, of *statische efficiëntie*, kan worden gemeten aan de hand van het surplusniveau wanneer technologieën als gegeven feit worden aangenomen, dat wil zeggen door hoe efficiënt inputs worden toegewezen aan een bepaald product (productieve efficiëntie) en de mate waarin consumentenbehoeften zijn bevredigd, gegeven het feit dat producenten voldoende winst maken om in de markt te blijven (allocatie-efficiëntie). Hogere statische efficiëntie kan bijvoorbeeld zichtbaar worden in lagere prijzen als gevolg van intensievere concurrentie en lagere toetredingsbarrières. Welvaart op de lange termijn, of *dynamische efficiëntie*, is het totale surplusniveau na verloop van tijd wanneer technologieën kunnen veranderen, bijvoorbeeld als gevolg van innovaties. Hogere dynamische efficiëntie in softwaremarkten kan voornamelijk worden gerealiseerd door productinnovatie (nieuwe producten), verhoogde keuzemogelijkheden voor consumenten en producten met een betere kwaliteit.

Een wisselwerking kan ontstaan tussen de concurrentie op de korte termijn (d.w.z. statische efficiëntie) versus innovatieprikkels op de lange termijn (d.w.z. dynamische efficiëntie). Deze wisselwerking kan worden geïllustreerd door een omgekeerde U-curve (figuur 4.1).³⁵ Door deze wisselwerking is het in principe mogelijk dat te veel *concurrentie* de dynamische efficiëntie vermindert. Hiervan kan bijvoorbeeld sprake zijn als de markt reeds in grote mate *concurrerend* is, zodat het verder verhogen van de intensiteit van de *concurrentie* zou leiden tot moordende concurrentie waardoor de marges die nodig zijn voor innovatie worden uitgehold (zie de rechterzijde van figuur 4.1). Als echter sprake is van ernstige *customer lock-in* is er over het algemeen in eerste instantie een lage concurrentie-intensiteit (d.w.z. dat de markt zich aan de linkerzijde van figuur 4.1 bevindt). In een dergelijke situatie gaat het verhogen van de *concurrentie* meestal hand in hand met verhoogde prikkels voor innovatie, waardoor ook de dynamische efficiëntie wordt verhoogd.

³⁴ Zie Bennett et al. (2001) voor definities van statische en dynamische efficiëntie.

³⁵ Zie Aghion en Griffith (2005).

Figuur 4.1 Relatie tussen de intensiteit van *concurrentie* en innovatieprikkels in de vorm van een 'omgekeerde U-curve'



Wanneer er een wisselwerking is tussen statische en dynamische efficiëntie nemen we aan dat resultaten die de dynamische efficiëntie verhogen ten koste van de statische efficiëntie over het algemeen beter presteren dan de resultaten die de statische efficiëntie verhogen ten koste van de dynamische efficiëntie. Dit is een heel logische aanname, gezien het feit dat wanneer de toekomst niet overmatig wordt genegeerd, de lange termijn meer belang moet krijgen dan de korte termijn. Ook zal verhoogde innovatie uiteindelijk de statische efficiëntie stimuleren.³⁶

4.1.1 Hoe beïnvloedt stimulering van OSS de welvaart op de korte en lange termijn?

Hieronder bespreken we eerst het effect van het stimuleren van OSS op de statische efficiëntie, hetgeen betrekkelijk eenvoudig is. Daarna bespreken we de effecten op de dynamische efficiëntie, hetgeen ingewikkelder is. Ter wille van het betoog zijn de onderstaande verhandelingen gebaseerd op de aanname dat het startpunt een marktsituatie is waarin PS domineert, met OSS in een ondergeschikte rol. Dit is een logisch uitgangspunt om de vraag te beantwoorden of het zinvol is om OSS te stimuleren.

Statische efficiëntie

Op de korte termijn verhoogt het stimuleren van OSS (zowel aan de vraag- als aan de aanbodzijde) over het algemeen de intensiteit van de concurrentie, omdat het zorgt voor een neerwaartse druk op de prijs van propriëtaire producten.^{37 38} Softwaregebruikers zijn niet bereid

³⁶ De overheid kan niettemin andere doelstellingen hebben. Zij kan bijvoorbeeld in eerste instantie voorkeur hebben voor het verhogen van de welvaart op de korte termijn.

³⁷ Dit hangt af van de mate waarin de aangeboden OSS en PS elkaar kunnen vervangen. Langere tijd werd Linux bijvoorbeeld niet gezien als een volwassen vervanger van Windows.

³⁸ De prijs heeft betrekking op de algehele gebruikskosten, dat wil zeggen inclusief de totale kosten van eigendom.

hoge prijzen te betalen als een vergelijkbaar product kosteloos verkrijgbaar is. Met (+) markeren we de gevallen waarin het stimuleren van OSS alleen invloed heeft op de prijzen (in neerwaartse richting). Bovendien kan, wanneer er sprake is van netwerkexternaliteiten of hoge overstapkosten voor consumenten, het stimuleren van OSS bijvoorbeeld door het optreden als aanjagende klant (zie sectie 4.1.3 voor meer informatie) helpen door toetredingsbarrières veroorzaakt door *customer lock-in*-effecten te verwijderen, onder de voorwaarde dat er voldoende prikkels zijn voor ontwikkelaars om de kwaliteit te verbeteren en compatibiliteit te stimuleren, waarmee de statische efficiëntie verder wordt verhoogd (gemarkeerd met ++). Uiteindelijk is het stimuleren van OSS altijd goed voor statische efficiëntie, omdat het de concurrentie op de korte termijn intensiveert (zie Tabel 4.1).

Tabel 4.1 Effecten van het stimuleren van OSS op statische efficiëntie: verhoogd concurrentieniveau

Aanwezigheid van netwerkexternaliteiten en/of hoge overstapkosten.	Gevolg van het stimuleren van OSS voor de indicatoren (prijs, <i>customer lock-in</i>) van statische efficiëntie	Algemene effecten van het stimuleren van OSS op statische efficiëntie
Nee	<ul style="list-style-type: none"> • Lagere prijzen 	+
Ja	<ul style="list-style-type: none"> • Lagere prijzen • Verlaging van toetredingsbarrières en vermindering van <i>customer lock-in</i> 	++

Dynamische efficiëntie

De vraag of het stimuleren van OSS de dynamische efficiëntie verhoogt of verlaagt, staat centraal bij het rechtvaardigen van overheidsbeleid. De effecten van het stimuleren van OSS op dynamische efficiëntie zijn echter meerduidelijk en moeilijker te beoordelen dan de effecten op statische efficiëntie.

Hoewel alle andere factoren gelijk zijn, genereert het feit dat PS wordt verkocht voor een prijs (die niet gelijk is aan nul) middels financiële indicatoren, zeer directe feedback over de behoeftes van klanten en de wensen van de verkoper. Er wordt daarom meer informatie gegenereerd over de vraagkarakteristieken, met name vergeleken met open source projecten die voornamelijk worden gedreven door prikkels aan de aanbodzijde, zoals intrinsieke motieven en vooruitzichten voor eigen gebruik.³⁹ Met andere woorden slagen PS-projecten beter in het ontwikkelen van producten voor een grote groep 'gemiddelde' of niet zeer ontwikkelde klanten (dat wil zeggen bestaand uit gebruikers anders dan programmeurs) dan open source projecten die (nog) niet de ambitie hebben een commercieel succes te worden. Daarom zouden in een pure OSS-wereld bepaalde type programma's helemaal niet worden ontwikkeld, met name toepassingen die interessant zijn voor gemiddelde gebruikers maar niet voor de programmeurs zelf. Interessant genoeg kan dit argument niet worden omgedraaid tot een argument ten gunste van OSS. Merk op dat zonder commerciële motieven (denk daarbij opnieuw aan door aanbod

³⁹ Dit argument is niet van toepassing op commerciële open source-projecten, bijvoorbeeld projecten die bedoeld zijn winst te genereren uit complementaire producten en diensten.

gedreven OSS-projecten) bepaalde ideeën kunnen worden ontwikkeld die anders geen kans zouden hebben overgenomen te worden in een commerciële omgeving (waarvoor uitzicht nodig is op financiële opbrengsten binnen een redelijke termijn). In een later stadium kan blijken dat een klein deel van deze ideeën wel degelijk geschikt zijn voor commerciële exploitatie. Daarmee kan OSS helpen ‘duizend bloemen te laten bloeien’ en blijken sommige daarvan later een massaproduct te zijn. Om al te globale sponsoring van een breed scala aan persoonlijke hobbyprojecten - met alle inefficiënties van dien - te vermijden, is er pas in een later stadium sprake van de noodzaak tot overheidsinterventie, dat wil zeggen wanneer er een markt voor enkele van deze ‘bloemen’ is ontstaan.

Er kunnen nu meer specifieke beschouwingen worden gedaan met betrekking tot dynamische efficiëntie (zie ook hoofdstuk 3):

1. Voor bepaalde typen software is kostbare inbreng nodig van expertise die complementair is aan programmeervaardigheden (bijvoorbeeld grafisch ontwerp), die zorgvuldig wordt ontwikkeld door specialisten buiten het netwerk van de softwareontwikkelaars en die makkelijk kan worden gekopieerd. Als de ontwikkelingskosten hoog zijn als gevolg van de noodzaak voor dergelijke specifieke complementaire inbreng, dan leidt een bedrijfsmodel dat is gebaseerd op PS eerder naar meer innovatie, omdat het geschikter is om de ontwikkeling van de grond te krijgen. Het stimuleren van OSS zou in dat geval de dynamische efficiëntie kunnen verminderen.
2. Als efficiënte contractering mogelijk is, kunnen softwareontwikkelaars met marktmacht (efficiënte) bedrijven die vervolginnovaties willen ontwikkelen met de broncode toeleveren, in ruil voor (een deel van) de winst die deze nieuwe producten genereren. Als dat het geval is, hoeft het stimuleren van OSS niet te leiden tot meer vervolginnovaties dan het efficiënte niveau. Niettemin is efficiënt contracteren van broncode in eigendom misschien niet haalbaar, bijvoorbeeld omdat het te moeilijk is vast te stellen of contractpartijen de licentievoorwaarden overtreden. In dat geval zal het stimuleren van OSS leiden tot een hoger niveau van vervolginnovatie.

4.1.2 Wanneer wordt het stimuleren van OSS aanbevolen?

Ter samenvatting van sectie 3.2 spelen wat betreft beleidsimplicaties de volgende vormen van marktfalen een centrale rol:

1. *Customer lock-in* veroorzaakt door netwerkexternaliteiten of overstapkosten;
2. Schaalvoordelen met betrekking tot de noodzaak specifieke complementaire inbreng in te kopen tegen substantiële kosten;
3. Kennispillovers met betrekking tot suboptimale mogelijkheden voor het contracteren van vervolginnovaties.

Elk van deze vormen van marktfalen kan al dan niet optreden in een markt voor een specifiek programma, afhankelijk van de kenmerken ervan. Vaststellen welke vormen van marktfalen aanwezig zijn en in welke mate is een noodzakelijke stap om een toepasselijk beleidsontwerp te maken.

We stellen de volgende hiërarchie voor vaststelling van de bovengenoemde vormen van marktfalen, beginnend met het meest problematische type marktfalen dat kan voorkomen in softwaremarkten. Beleidsmakers dienen eerst te overwegen of er sprake is van ernstige *customer lock-in* in de markt (marktfalen 1).⁴⁰ Als dat het geval is, moet worden vastgesteld welk type bedrijfsmodel het meest geschikt is qua dynamische efficiëntie (marktfalen 2 en 3).

1. Laten we eerst aannemen dat er geen ernstig *customer lock-in*-probleem is (marktfalen 1 hierboven). Moet de overheid dan interveniëren om OSS te stimuleren? Aan de vraagzijde hebben klanten niet te maken met ernstige obstakels om over te stappen op nieuwe toetreders zodra deze ten tonele verschijnen. Aan de aanbodzijde hebben nieuwe toetreders (of ze nu PS of OSS aanbieden) niet te maken met substantiële toetredingsbarrières die worden opgeworpen omdat ze moeten concurreren met een bestaande softwareleverancier. Wanneer deze toetreders meer innovatief zijn, zullen ze in staat zijn snel een marktaandeel te verwerven als er vraag is naar een nieuwe productvariëteit. Het ligt daarom voor de hand dat marktprocessen resulteren in een efficiënte constellatie van licentietypes. In principe is er dan geen noodzaak voor bemoeienis met de door de markt aangestuurde opkomst van innovaties. Zich weerhouden van ex ante interventie is in dat geval de beste optie, om te voorkomen dat marktprocessen worden verstoord (zie ook onder sectie 4.2).
2. In het tweede geval nemen we aan dat er sprake is van een ernstig *customer lock-in*-probleem. Nieuwe toetreders kunnen dan in een markt die wordt gedomineerd door een gevestigde aanbieder van PS weinig kansen hebben om toe te treden, zelfs als ze een superieur product aanbieden tegen een lagere prijs. De vraag is nu of er indicaties zijn dat, vanuit welvaartspectief, PS als bedrijfsmodel wellicht de voorkeur heeft boven OSS. Nauwkeuriger gezegd, de vraag is of het zinvol is OSS op de één of andere wijze te steunen. Het centrale punt hierbij is dat vanuit welvaartspectief OSS niet in alle omstandigheden het beste bedrijfsmodel is (dit heeft betrekking op marktfalen 2 en 3 hierboven, tevens zoals is besproken in het voorgaande gedeelte). Als (i) er voor softwareontwikkeling een specifieke complementaire inbreng vereist is, terwijl (ii) er geen obstakels zijn (zoals informatieasymmetrieën) met betrekking tot het contracteren van vervolginnovaties, dan is er geen belangrijke reden om aan te nemen dat een open source-model beter presteert, als bedrijfsmodel om ontwikkeling van de grond te krijgen, dan PS. Wat impliceren deze voorwaarden?

⁴⁰ Om vast te stellen hoe ernstig het *customer lock-in*-probleem is, kan men bijvoorbeeld onderzoeken of de overstapkosten hoog zijn. Econometrische analyse of marktonderzoek kan worden uitgevoerd om deze kosten te bepalen. Daarvoor zijn een analytisch kader en casusonderzoeken beschikbaar in Pomp et al. (2005).

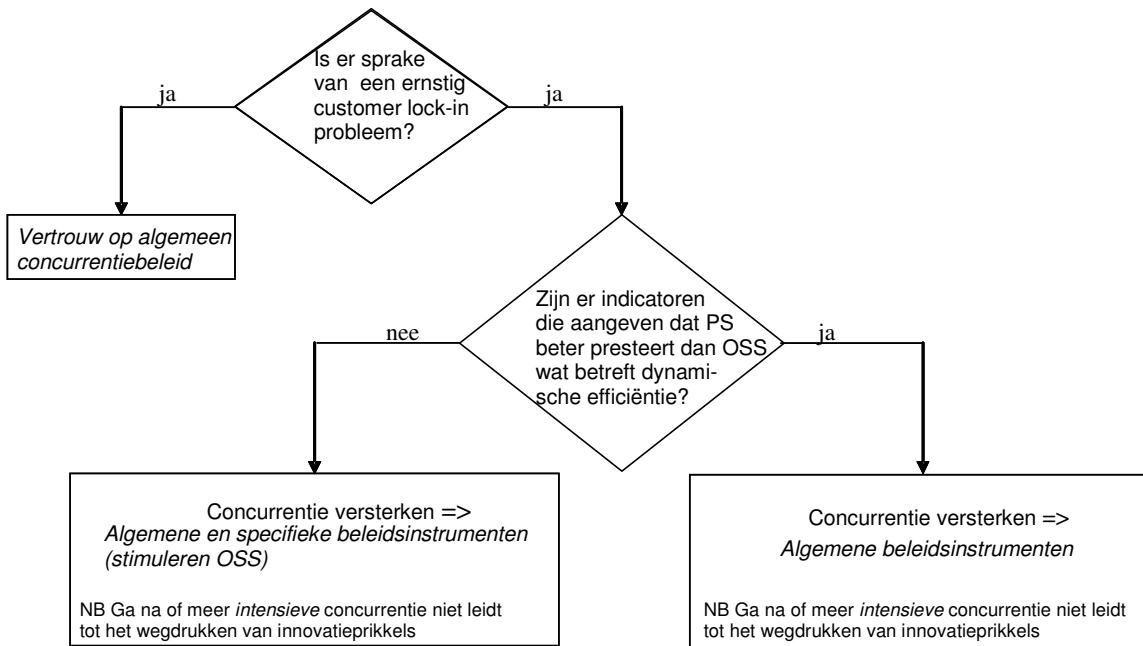
- Als aan de voorwaarden (i) of (ii), of beide, in aanzienlijke mate wordt voldaan, is het wellicht *niet* verstandig om beleid te introduceren dat specifiek gericht is op het stimuleren van OSS. Het actief stimuleren van OSS conflicteert met de effectiviteit van op PS gebaseerde bedrijfsmodellen en kan leiden tot verstoringen van de dynamische efficiëntie. Hier moet echter worden opgemerkt dat *algemeen*⁴¹ beleid voor het reduceren van ernstige *customer lock-in* zeer heilzaam kan zijn, onder de voorwaarde dat meer *concurrentie* niet ten koste gaat van minder innovatie en dynamische efficiëntie.
- Als aan voorwaarde (i) noch aan voorwaarde (ii) wordt voldaan kan het direct stimuleren van OSS, (met *specifiek* beleid) naast het bevorderen van *concurrentie* door algemene instrumenten, zeer behulpzaam zijn bij het overwinnen van *customer lock-in* in die markt.

Het is ook belangrijk om na te gaan of het stimuleren van concurrentie via OSS innovatieprikkelers kan schaden (denk aan de discussie over de mogelijke wisselwerking tussen statische en dynamische efficiëntie, geïllustreerd door figuur 4.1). In theorie zou dit kunnen gebeuren als reeds in het begin een hoge concurrentie-intensiteit is. Echter, wanneer de markt op slot zit door *customer lock-in* en de gevestigde aanbieder daardoor weinig concurrentiedruk ervaart, is het waarschijnlijk dat meer concurrentie innovatie juist aanwakkert in plaats van verder afzwakt.

Gebaseerd op de hierboven genoemde argumenten kan de volgende beslissingsboom voor beleidsmakers worden geconstrueerd.

⁴¹ Dat wil zeggen dat het terugbrengen van toetredingsbarrières voor alle soorten software (niet specifiek OSS), zoals het verplichte gebruik van open standaarden bij alle overheidsaankopen.

Figuur 4.2 Beslissingsboom voor beleidsmakers



Tabel 4.2 illustreert het stroomschema met enkele voorzichtige voorbeelden. We beklemtonen dat deze voorbeelden bedoeld zijn om een eerste idee te geven over hoe ons kader moet worden gebruikt, en niet als oproep voor beleidsmaatregelen voor deze markten. In specifieke gevallen is verdere analyse noodzakelijk.

Tabel 4.2 Voorzichtige voorbeelden van de beslissingsboom

		Zijn er aanwijzingen dat PS beter presteert dan OSS wat betreft dynamische efficiëntie?	
		Ja	Nee
Is er sprake van een ernstig customer lock-in-probleem?	Nee	Geavanceerde statistische software	Mathematische software, internetbrowsers, antivirussoftware, web content management-software
	Ja	'High-end' interactieve games, ESP-software	Besturingssystemen voor pc, standaardkantoortoepassingen, enterprise content management-software

We leggen hieronder de voorbeelden wat gedetailleerder uit. In de markt van geavanceerde statistische software (bijvoorbeeld STATA, Eviews en SPSS) en mathematische software (bijvoorbeeld Mathematica en haar open source-concurrenten Maxima en Sage), zijn lock-in-problemen minder substantieel dan in markten waar klanten continu bestanden en toepassingen uitwisselen (bijvoorbeeld besturingssystemen en kantoortoepassingen). Daarom vormen

netwerkexternaliteiten geen substantiële barrières voor toetreding tot de markt. Er kunnen natuurlijk andere barrières zijn, maar de kans is groot dat deze minder problematisch zijn.

In het geval van geavanceerde statistische software zijn de ontwikkelingskosten hoog als gevolg van de noodzaak tot specifieke complementaire inbreng, dat wil zeggen experts die inzicht hebben in gecompliceerde econometrische methodes. Daarom kan PS geschikter zijn om mogelijkheden te bieden de kosten terug te verdienen. Toch is voor een toetreder die een superieur alternatief aanbiedt de positie van de gevestigde leverancier overkomelijk en kan hij de noodzakelijke investeringen terugverdienen.

Voor de ontwikkeling van mathematische software is geen (of in mindere mate) dergelijke complementaire specifieke inbreng nodig. Daarom zijn er voldoende prikkels om zowel OSS (Maxima) als PS (Mathematica) te ontwikkelen, en als gevolg van de relatief lage toetredingsbarrières kan concurrentie plaatsvinden tussen deze softwareopties. OSS kan echter tot meer vervolginnovaties leiden. Andere voorbeelden zijn internetbrowsers (Internet Explorer en Firefox), antivirussoftware en web content management-software (bijvoorbeeld PS: Interwoven, FatWire en OSS: WordPress, Joomla!).

In de tweede rij zijn twee groepen voorbeelden gegeven met ernstige *customer lock-in*-problemen. Ten eerste van markten waar PS het beter doet als het gaat om verhoging van het innovatieniveau. De markt van high-end interactieve games, waarin spelers een significante hoeveelheid tijd doorbrengen met het spelen van games met elkaar, zijn de netwerkeffecten zeer sterk, hetgeen toetredingsbarrières opwerpt voor nieuwe games. Bovendien worden de ontwikkelingskosten hoogdoor de noodzaak geavanceerde grafische vaardigheden in te kopen. Evenzo kunnen er hoge overstapkosten zijn in de markt van ERP-software (bijvoorbeeld SAP en haar open source-concurrent Compiere). Daarnaast is voor de ontwikkeling van de software specifieke inbreng nodig, zoals specifieke kennis op het gebied van accountancy en bedrijfsadministratie, hetgeen de ontwikkelingskosten aanzienlijk verhoogt. In deze gevallen kunnen op PS gebaseerde bedrijfsmodellen beter in staat zijn de kosten terug te verdienen.

Ten tweede is er bij besturingssystemen voor pc, zoals Windows en Linux, ook sprake van substantiële netwerkexternaliteiten en overstapkosten. Er is echter in verhouding minder noodzaak voor specifieke complementaire expertise (naast de juiste programmeervaardigheden) om dergelijke software te ontwikkelen. Daarom is er geen indicatie dat PS beter presteert dan OSS in het stimuleren van innovatie. Door stimulering van OSS op een manier dat het een significant aantal geïnstalleerde systemen biedt voor concurrerende aanbieders, kunnen de barrières voor consumenten om over te stappen aanzienlijk worden verlaagd. Andere voorbeelden waar deze redenering van toepassing is, is kantoorsoftware (tekstverwerkers, zetwerktoepassingen, spreadsheets en presentatieverwerkers) en enterprise content management-software (bijvoorbeeld Hummingbird en haar open source-concurrent Alfresco).

Zoals eerder is aangegeven is de bespreking van deze voorbeelden voorzichtig. Een completere analyse van de afzonderlijke gevallen valt buiten de reikwijdte van deze studie.

4.1.3 Beleidsopties als interventie wenselijk is

Verschillende beleidsopties kunnen worden onderscheiden, afhankelijk van de vraag of deze gericht zijn op software in het algemeen, of op OSS in het bijzonder. Een beleidsoptie is *algemeen* als deze gericht is op het reduceren van toetredingsbarrières in softwaremarkten, onafhankelijk van het type software. Wanneer beleidsinterventie direct gericht is op het stimuleren van OSS, noemen we deze optie *specifiek*.

We bespreken eerst de algemene beleidsinstrumenten. Ten eerste kan het verplicht stellen van interoperabiliteit (in de betekenis van compatibiliteit) tussen concurrerende producten helpen de netwerkeffecten te internaliseren door gebruikers van verschillende softwareprogramma's in staat te stellen zonder problemen bestanden uit te wisselen.⁴² Dit type interventie kan met name belangrijk zijn wanneer PS-leveranciers reden hebben om interoperabiliteit te belemmeren, bijvoorbeeld door het implementeren van gesloten standaarden om hun marktmacht te behouden. De overheid kan interoperabiliteit ook mogelijk maken als zij optreedt als coördinerend intermediair. Ten tweede, om de vermeende (niet-financiële) overstapkosten voor consumenten te verlagen, kan de overheid de transparantie van softwareproducten verbeteren door het verschaffen van informatie over de beschikbaarheid, prijzen, en locatie van producten, en gerelateerde diensten te bevorderen. Deze soorten ex-ante beleidsinterventies kunnen tot op zekere hoogte ook het toepassen van concurrentiewetgeving inhouden (ex post interventie), met de bedoeling de markt open te houden voor innovatie en ervoor te zorgen dat consumenten uiteindelijk bepalen welke standaard, technologie of product zal winnen.⁴³

We richten ons nu op specifieke beleidsinstrumenten. De overheid kan OSS stimuleren door als aanjagende klant op te treden, bijvoorbeeld door het stimuleren of zelfs verplicht stellen van OSS in de publieke sector en semipublieke sector door OSS als voorwaarde bij de inkoop op te leggen. Ook grote bedrijven kunnen gemakkelijk OSS toepassen, indien inkoopmanagers hun keuze voor OSS kunnen motiveren door te wijzen op het feit dat gevestigde organisaties ook OSS aanschaffen.⁴⁴ Een voorbeeld van specifiek beleid is inkoop voor de ontwikkeling van een databasemanagementsysteem dat overheidsinstanties met elkaar verbindt. Specifiek beleid kan effectief zijn in het laten 'overhellen' van de markt in de richting van een levensvatbaar marktaandeel voor OSS. Het onderliggende mechanisme is dat de overheid hierdoor een significant aantal gebruikers realiseert voor een alternatief OSS-product, en het aantrekkelijker te maken voor andere consumenten om te volgen.

De volgende tabel somt op welke beleidsinstrumenten algemeen en welke specifiek zijn. De tabel gaat ervan uit dat interventie de welvaart bevordert en heeft daarom alleen betrekking op gevallen waarin de effecten voor de welvaart zodanig zijn dat beleidsmakers het algemeen de

⁴² Zie ook Lee (2006).

⁴³ Voor meer informatie hierover in het licht van de Europese Microsoft-rechtszaak, zie Larouche (2008).

⁴⁴ Zie Zwiebel (1995).

moeite waarde vinden in te grijpen. De tabel doet geen uitspraken over de relatieve effectiviteit van de verschillende instrumenten.

Tabel 4.2 Algemene versus specifieke beleidsinterventies

	Algemene	Specifieke
Ex ante interventie		
Aanjagende klant met betrekking tot OSS		✓
OSS bij openbare aanbestedingen verplicht stellen		✓
Compatibiliteit tussen concurrerende producten afdwingen	✓	
Verhogen van transparantie van softwareproducten	✓	
Ex post interventie		
Toepassen van concurrentiewetgeving	✓	

Ten slotte laat de onderstaande tabel de punten zien waar de verschillende interventies worden toegepast, namelijk het niveau van de structuur van de markt of het niveau van de houding van de marktdeelnemers.

Tabel 4.4 Punten waar overheidsinterventie wordt toegepast

Interventie beïnvloedt:	Marktstructuur	Markthouding
Vraag naar software	OSS bij overheidsaankopen verplicht stellen (NB: deze beleids optie beïnvloedt ook de houding, door het effect van de verplichting)	Optreden als aanjagende klant met betrekking tot OSS
Aanbod van software	Compatibiliteit tussen concurrerende producten afdwingen Toepassen van concurrentiebeleid	Verhogen van transparantie van softwareproducten Toepassen van concurrentiebeleid

Merk op dat deze lijst niet volledig is, maar eerder een overzicht van de soorten interventies die zo lichtgewicht mogelijk zijn. Een volledige lijst is waarschijnlijk niet mogelijk, omdat de mogelijkheden voor interventie zo goed als eindeloos zijn. Onze korte lijst bevat met name geen mogelijkheden voor het subsidiëren van de vraag (bijvoorbeeld door de overheid gesponsorde distributie van OSS onder de burgers, opleiden op scholen) of het aanbod (bijvoorbeeld belastingsubsidies voor softwareontwikkelaars die OSS ontwikkelen, jaarlijkse onderscheidingen voor het beste OSS-product). Op voorhand is het de moeite waard dergelijke interventies mee te nemen bij het overwegen van de mogelijke opties. Laten we echter niet vergeten dat behalve de risico's van overheidsfalen met betrekking tot de distributie van subsidies (zie hieronder voor meer informatie over overheidsfalen) subsidies ook voor extra verstoringen zorgen. De reden is dat om de subsidies te financieren, belastingen op andere plaatsen in de economie moeten worden verhoogd, terwijl belastingheffing over het algemeen verstorend van aard is. Deze observatie suggereert dat een kosten-batenanalyse wenselijk is om beleids opties, die in specifieke gevallen van toepassing zouden kunnen zijn, te kunnen

vergelijken.⁴⁵ Ten behoeve van de beknoptheid houden we ons in het onderstaande bij onze korte lijst van mogelijke interventies, maar we zijn ons ervan bewust dat in die gevallen waar een specifiek type interventie wenselijk is meer opties kunnen worden overwogen.

4.2 Overheidsfalen

Het is belangrijk om te beseffen dat ‘goede bedoelingen’ niet genoeg redenen vormen voor overheidsinterventie, omdat interventie zowel voordelen als nadelen kan hebben. Bijgevolg moet men, om vast te kunnen stellen dat beleid marktfalen daadwerkelijk corrigeert, zich afvragen of beleid werkelijk de inefficiëntie of het verlies aan welvaart vermindert die was veroorzaakt door het desbetreffende marktfalen.⁴⁶ Met andere woorden, men moet rekening houden met de mogelijkheid van overheidsfalen, hetgeen falen is dat ontstaat wanneer de overheid een nieuwe inefficiëntie introduceert “omdat het niet had moeten ingrijpen of wanneer het een bepaald probleem [...] effectiever had kunnen oplossen, dat wil zeggen door het genereren van hogere netto-opbrengsten” (Winston, 2006, p. 2-3).

Helaas bestaan er met betrekking tot mogelijk overheidsfalen geen ‘algemene inzichten’, in tegenstelling tot marktfalen, waarover de economen het eens zijn (marktmacht, natuurlijk monopolie, imperfecte informatie, netwerkexternaliteiten en publieke goederen). De effecten van interventie kunnen uiteindelijk alleen empirisch worden vastgesteld. Ten behoeve van dit rapport is het echter nuttig om een korte lijst met vormen van falen op te stellen waarmee ex ante rekening gehouden moet worden, dat wil zeggen voorafgaande aan het implementeren van bepaald beleid.

Overheidsinterventie die gericht is op het direct stimuleren van OSS kan de volgende welvaartsverstoringen veroorzaken:

- Wanneer de overheid optreedt als aanjagende klant of de markt op een andere manier aanstuurt, kan niet worden gegarandeerd dat zij een keuze maakt die beter is dan de resultaten van de marktprocessen of de keuzes die op gedecentraliseerd niveau door marktdeelnemers worden gemaakt. In de economieliteratuur is dit soort kritiek op innovatiebeleid standaard, waarbij economen over het algemeen tegen beleid adviseren dat gericht is op het kiezen of ondersteunen van ‘winnaars’ (zie bijvoorbeeld Boone en van Damme, 2004). Zoals Schmidt en Schnitzer (2003) stellen, kan een onjuiste productkeuze in de context van softwaremarkten, onafhankelijk van de aanwezigheid van marktfalen, de markt naar de verkeerde kant laten overhellen. Het middel is dan erger dan de kwaal. Ook kunnen hierdoor concurrentie- en innovatieprikkels afnemen.

⁴⁵ Het valt buiten de reikwijdte van dit rapport om de voor- en nadelen van de verschillende beschikbare opties met elkaar te vergelijken.

⁴⁶ Winston (2006).

- Beleidsmakers zijn misschien niet in staat het succes op de lange termijn van een publiekelijk ondersteund OSS-project te bepalen, bijvoorbeeld door onvolledige informatie. Ten gevolge van overheidsaanbestedingen met een voorkeur in de richting van OSS kan een op dat moment winstgevende leverancier van PS haar motivering om te investeren kwijtraken.
- Inkoopprocessen bij de overheid met een voorkeur in de richting van een bepaald type technologie, hetgeen ook het maken van keuzes tussen verschillende projecten of leveranciers inhoudt, nodigt uit tot lobbyen door potentiële leveranciers en resulteert normaal gesproken in inefficiënties als gevolg van winstnastrevend gedrag.

4.3 Samenvatting

In dit gedeelte hebben we de vraag aan de orde gesteld wat beleidsmakers kunnen doen om markten beter te laten werken en de welvaart in softwaremarkten te verhogen. In sectie 3.2 hebben we een centrale rol toegewezen aan de volgende vormen van marktfalen: *customer lock-in* als gevolg van netwerkexternaliteiten, schaalvoordelen veroorzaakt door de noodzaak specifieke, complementaire inbreng in te kopen tegen aanzienlijke kosten en kennispillowers met betrekking tot de mogelijkheden voor het contracteren van vervolginnovaties. We stellen voor om een hiërarchie te creëren in het vaststellen van deze vormen van marktfalen, beginnend met het meest problematische type dat kan voorkomen in softwaremarkten. Beleidsmakers moeten eerst in beschouwing nemen of er sprake is van ernstige *customer lock-in* in een specifieke softwaremarkt. Als dat het geval is, kan beleidsinterventie wenselijk zijn. Er moet echter worden bepaald welk type bedrijfsmodel eventueel het meest geschikt is in termen van dynamische efficiëntie. Als er voor softwareontwikkeling specifieke complementaire inbreng vereist is, terwijl er geen obstakels zijn met betrekking tot het contracteren van vervolginnovaties, dan is er geen belangrijke reden om aan te nemen dat een open source model beter presteert dan PS als bedrijfsmodel om ontwikkeling van de grond te krijgen. In dat geval kunnen algemene beleidsopties, gericht op het verlagen van de toetredingsbarrières, worden toegepast. Het verplichten van interoperabiliteit (compatibiliteit) en het verhogen van de transparantie van softwareproducten zijn als voorbeelden hiervan genoemd. Als geen van de hierboven genoemde voorwaarden van toepassing zijn, dan kunnen bedrijfsmodellen op basis van open source beter presteren dan PS in het stimuleren van interactieve en sequentiële innovatieprocessen. Naast de eerder genoemde algemene beleidsinstrumenten kan concurrentie in dat geval worden versterkt door specifiek op OSS gerichte beleidsinstrumenten, zoals het optreden als aanjagende klant of het verplicht stellen van OSS in openbare aanbestedingen. Overigens kan over het algemeen invoering van concurrentiewetgeving noodzakelijk zijn als aanvullende vorm van interventie of supervisie. Voorafgaande implementatie van deze beleidsmaatregelen is het echter van groot belang dat men zich realiseert dat interventie kan leiden tot overheidsfalen. Daarom moet geprobeerd worden het netto-effect van interventie op

de welvaart te schatten, rekening houdend met het feit dat de overheid normaal gesproken niet even goed is in het maken van technologische keuzes als de markt.

5 Conclusies en discussie

Deze studie analyseert de economische effecten van het publiekelijk stimuleren van open source software (OSS) op concurrentie en innovatie in softwaremarkten. In de inleiding hebben we de diverse vragen gesteld:

1. Welke prikkels zijn er om software te ontwikkelen onder de verschillende licentiemodellen?
2. Wat zijn de soorten marktfalen in softwaremarkten?
3. Hoe wordt welvaart beïnvloed door innovatieprikkels en marktfalen?
4. Wat kunnen beleidsmakers doen om softwaremarkten beter te laten werken en de welvaart te verhogen?

Aan de aanbodzijde kunnen softwaremarkten zowel financiële als niet-financiële motieven van softwareontwikkelaars vertonen, en marktmacht en kennisillovers als mogelijke vormen van marktfalen. Wanneer beleidsmakers ingrijpen in softwaremarkten moeten ze rekening houden met het feit dat er wisselwerkingen bestaan tussen het stimuleren van vervolginnovaties (ex post) en het voorkomen dat anderen gratis kunnen meeliften met bestaande innovaties (wat goed is voor ex ante prikkels). Het antwoord op de vraag of propriëtaire of open source-licenties optimaal zijn, ligt daarom niet voor de hand en is niet eenduidig. Op basis van economische argumenten concluderen we dat vanuit het gezichtspunt van de welvaart propriëtaire modellen over het algemeen beter presteren wanneer de ontwikkelingskosten hoog zijn als gevolg van de noodzaak voor specifieke inputs en optimale contractering voor een licentievergoeding voor afgeleide werken haalbaar is. In tegenstelling hiermee hebben open source-modellen over het algemeen de voorkeur ten opzicht van propriëtaire modellen, als er voor de ontwikkeling voornamelijk programmeervaardigheden nodig zijn (die wel zeer geavanceerd kunnen zijn) en creatieve spillovers sociale waarde hebben terwijl propriëtaire licenties te beperkend zijn om voldoende prikkels te bieden voor vervolginnovaties. Dit zijn natuurlijk gestileerde argumenten. Voor specifieke gevallen zijn afzonderlijke onderzoeken noodzakelijk die dieper op de details ingaan.

Voordat we de beleidsaanbevelingen bespreken, moeten we benadrukken dat er voor specifieke interventie sterke gronden moeten bestaan, te meer omdat er een risico van overheidsfalen bestaat. In de praktijk zijn concurrentiebeleid en vrije keuzemogelijkheden voor de beste middelen om tegenwicht te bieden aan marktmacht en om toetredingsbarrières te verlagen. De reden hiervoor is dat consumenten beter geïnformeerd zijn over hun eigen voorkeuren met betrekking tot nieuwe producten dan de overheid. Niettemin zijn er argumenten die spreken voor aanvullende ex ante interventie. Geverifieerd dient echter te worden dat dit soort interventie geen averechtse uitwerking heeft op de handhaving van concurrentiewetgeving. Een meer diepgaande analyse van de in dit document behandelde beleidsinterventies ter aanvulling op concurrentiebeleid valt buiten het bestek van deze studie.

De mate waarin ex ante interventie wenselijk is, hangt af van de aanwezigheid van marktfalen. Allereerst moet worden vastgesteld of er sprake is van ernstige *customer lock-in*-problemen in de markt. Als dat het geval is, kan beleidsinterventie wenselijk zijn, afhankelijk van de vraag welk bedrijfsmodel geschikter is om innovatieprikkelers te beschermen en te stimuleren. Als propriëtaire businessmodellen beter presteren, kunnen algemene beleidsopties, gericht op het verlagen van toetredingsbarrières, worden toegepast om *customer lock-in* te reduceren. Het opleggen van interoperabiliteit (compatibiliteit) en het verhogen van transparantie van softwareproducten zijn voorbeelden hiervan. Het stimuleren van OSS kan wenselijk zijn wanneer open source-licenties de voorkeur hebben boven propriëtaire licenties met betrekking tot dynamische efficiëntie. Beleidsopties die kunnen helpen *customer lock-in*-problemen te verminderen en de dynamische efficiëntie te verhogen, zijn het optreden als aanjagende klant of het verplicht stellen van OSS in openbare aanbestedingen. Handhaving van concurrentiewetgeving kan in het algemeen overigens noodzakelijk zijn als aanvullende vorm van interventie of toezicht.

We sluiten dit gedeelte af met een korte observatie. We hebben in dit document de effecten van beleidsinterventie op het niveau van concurrentie en innovatie in softwaremarkten besproken, zonder rekening te houden met landspecifieke kenmerken, met name omdat softwaremarkten kunnen worden beschouwd als mondiale markten. Wanneer we echter toch kort naar de Nederlandse softwaremarkten kijken, is het onwaarschijnlijk dat Nederland een grote invloed kan uitoefenen op mondiale softwaremarkten. Beleidsmaatregelen die gericht zijn op het stimuleren van OSS, die wel kunnen helpen een significant klantenbestand voor OSS-producten in Nederland op te bouwen, moeten in dit licht worden gezien.

Literatuur

Aghion, P. en R. Griffith, 2005, *Competition and Growth: Reconciling Theory and Evidence*, MIT Press.

Alexy, O. en M. Leitner, 2008, Rewards, en Their Effect on the Motivation of Open Source Software Developers, Mimeo, Technische Universität München.

Bennett, M., P. de Bijl en M. Canoy, 2001, Future Policy in Telecommunications, An Analytical Framework, CPB Document 5.

Bessen, J. en E. Maskin, 2004, Intellectual Property on the Internet: What's Wrong with Conventional Wisdom? Beschikbaar op: <http://www.researchoninnovation.org/iippap2.pdf>.

Bessen, J. en E. Maskin, 2006, Sequential, Innovation, Patents, and Imitation, NajEcon Working Paper Reviews.

Boone, J. en E.E.C. van Damme, 2004, Marktstructuur en innovatie, in: B. Jacobs en J.J.M. Theeuwes (eds.), *Innovatie in Nederland: De Markt Draalt en de Overheid Faalt*, KVS Amsterdam, pp. 71-92.

De Cock Buning, M., 1993, Auteursrecht en 'reverse engineering': techniek en theorie, *Intellectuele eigendom & reclamerecht*, 9(5), pp. 129-137.

Gallini, N. en S. Scotchmer, 2002, Intellectual Property: When Is It the Best Incentive System?, in: A. Jaffe, J. Lerner en S. Stern (eds.), *Innovation Policy and the Economy*, Vol. 2, MIT Press, pp. 51-78.

Geest, S.A. van der, R.C.G. Haffner, P.T. van der Schans en M. Varkevisser, 2000, *Marktwerking en nieuwe ICT-markten: de markt voor software*, Onderzoekcentrum Financieel Economisch Beleid (OCFEB) en Erasmus Universiteit Rotterdam.

Ghosh, R.A., R. Glott, B. Krieger en G. Robles, 2002, The Netherlands Free/Libre and Open Source Software: Survey and Study FLOSS, Mimeo, International Institute of Infonomics, University of Maastricht, available at <http://www.infonomics.nl/FLOSS/report/>.

Giuri, P., F. Rullani en S. Torrisi, 2008, Explaining leadership in virtual teams: The case of open source software, *Information Economics and Policy*, 20(4), pp. 305-315.

- Groenenboom, M.M., 2002, Software licencies: van closed source tot open source, *Computerrecht* 1, pp. 21-29.
- Guibault, L. en O. van Daalen, 2005, Unravelling the Myth around Open Source Licenses: An Analysis from a Dutch and European Law Perspective, Mimeo, Institute for Information Law, University of Amsterdam.
- Farrell, J. en P. Klemperer, 2007, Coordination and Lock-In: Competition with Switching Costs and Network Effects, In: Armstrong and Porter (Eds.), *Handbook of Industrial Organization*, Vol. 3, Elsevier.
- Hann, I-H., J. Roberts, S. Slaughter en R. Fielding, 2004, An empirical analysis of economic returns to open source participation, Mimeo, Carnegie Mellon University.
- Henkel, J., 2005, Selective revealing in open innovation processes: The case of embedded Linux, *Research Policy*, 35, pp. 953–969.
- Knubben, B.S.J., 2001, *Open Source Software, De Bron Geopend: Een economische analyse van Open Source Software*, Ph.D. thesis Faculteit der Economische Wetenschappen en Econometrie, University of Amsterdam.
- Lakhani, K.R. en R.G. Wolf, 2005, Why hackers do what they do: understanding motivation and effort in free/open source software projects, In: J. Feller, B. Fitzgerald, S. Hissam en K.R. Lakhani (eds.), *Perspectives on Free and Open Source Software*, MIT Press, Cambridge, MA.
- Larouche, P., 2008, The European Microsoft Case at the Crossroads of Competition Policy and Innovation, TILEC Discussion Paper 2008-021, Tilburg University, available at SSRN: <http://ssrn.com/abstract=1140165>.
- Lee, J.-A., 2006, New Perspectives on Public Goods Production: Policy Implications of Open Source Software, *Vanderbilt Journal of Entertainment and Technology Law*, 9(1), pp. 45-112.
- Lerner, J., P.A. Pathak en J. Tirole, 2006, The Dynamics of Open-Source Contributors, *American Economic Review*, 96(2), pp. 114-118.
- Lerner, J. en J. Tirole, 2002, Some Simple Economics of Open Source, *Journal of Industrial Economics*, 50(2), pp. 197-234.

Lerner, J. en J. Tirole, 2005a, The Economics of Technology Sharing: Open Source and Beyond, *Journal of Economic Perspectives*, 19(2), pp. 99-120.

Lerner, J. en J. Tirole, 2005b, The Scope of Open Source Licensing, *Journal of Law, Economics, and Organization*, 21(1), pp. 20-56.

Mateos-Garcia, J. en W.E. Steinmueller, 2008, The institutions of open source software: Examining the Debian community, *Information Economics and Policy*, 20(4), pp. 333-344.

Maurer, S.M. en S. Scotchmer, 2006, Open Source Software: The New Intellectual Property Paradigm, In: T. Hendershott (ed.), *Handbook of Economics and Information Systems*, Elsevier, pp. 285-319.

McGowan, D., 2005, Legal Aspects of Free and Open Source Software, In: J. Feller (ed.), *Perspectives on Free and Open Source Software*, MIT Press, pp. 439-440.

Mendys-Kamphorst, E., 2002, Open vs. closed: Some consequences of the open source movement for software markets, CPB Discussion Paper 13.

Ministerie van Economische Zaken, 2007, Nederland Open in Verbinding: Een actieplan voor het gebruik van Open Standaarden en Open Source Software bij de (semi-)publieke sector, Den Haag. Beschikbaar op <http://www.ez.nl/dsresource?objectid=153181&type=PDF>.

Mockus, A., R.T. Fielding en J.D. Herbsleb, 2002, Two Case Studies of Open Source Software Development: Apache en Mozilla, *ACM Transactions on Software Engineering and Methodology*, 11(3), pp. 309-346.

Pomp, M., V. Shestalova en L. Rangel, 2005, Switch on the competition. Causes, consequences and policy implications of consumer switching costs, CPB Document 97.

Raymond, E., 1999, *The Cathedral and the Bazaar*, O'Reilly Media.

Rochet, J.-C. en J. Tirole, 2006, Two-Sided Markets: A Progress Report, *RAND Journal of Economics*, 35(3), pp. 645-667.

Shy, O., 2001, *The Economics of Network Industries*, Cambridge University Press.

Schmidt, K. en M. Schnitzer, 2003, Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market, CEPR Discussion Paper 3793.

Stiglitz, J.E., te verschijnen, Economic Foundations of Intellectual Property Rights, *Duke Law Journal* (paper delivered at 2007 Annual Frey Lecture, Duke University).

Van Wendel de Joode, R., 2005, *Understanding open source communities: an organizational perspective*, Ph.D. thesis, University of Technology, Delft.

Von Hippel, E, 2002, Open source software projects as user innovation networks, Mimeo, MIT Sloan School of Management.

Von Hippel, E. en G. von Krogh, 2003, Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science, *Organization Science*, 14(2), pp. 209-223.

Winston, C., 2006, *Government Failure versus Market Failure. Microeconomics Policy Research and Government Performance*, AEI – Brookings Joint Center for Regulatory Studies, Washington, D.C.

Zwiebel, J., 1995, Corporate Conservatism and Relative Compensation, *The Journal of Political Economy*, 103(1), pp. 1-25.