

# Ondersteuning op open source software





oktober 2007

OSOSS

Tekst en opmaak: Daniël Vijge

Stichting ICTU

OSOSS (Open Source als Onderdeel van uw Software Strategie)

Wilhelmina van Pruisenweg 104

Postbus 84011

2508 AA Den Haag

T +31 (0)70 888 78 25

E [info@ososs.nl](mailto:info@ososs.nl)

I [www.ososs.nl](http://www.ososs.nl)

Op deze uitgave is de Creative Commons licentie van toepassing. Het is toegestaan informatie uit deze publicatie te kopiëren, te verspreiden en te bewerken mits deze uitgave als bron wordt vermeld en de informatie niet voor commerciële doeleinden wordt gebruikt. Indien de gebruiker het werk bewerkt, kan het daaruit ontstane werk uitsluitend krachtens dezelfde licentie worden verspreid. Meer informatie hierover vindt u op <http://creativecommons.org/licenses/by-nc-sa/3.0/nl/>

## Inhoudsopgave

<b>Inleiding</b>	<b>4</b>
<i>Leeswijzer</i>	4
<i>Het onderzoek</i>	5
<b>Ondersteuning theoretisch bekeken</b>	<b>6</b>
<i>Theorieën over ondersteuning</i>	6
<b>Verschillende soorten software</b>	<b>8</b>
<i>Off-the-shelf- en maatwerksoftware</i>	8
<i>Open source en diversie variaties van gesloten source software</i>	8
<b>Voordelen van open source software</b>	<b>9</b>
<b>Ondersteuning op software</b>	<b>11</b>
<i>Ondersteuning op gesloten software</i>	11
<i>Communities en open source software</i>	11
<i>Ondersteuning op open source software</i>	11
<i>Open source software en het ASL model</i>	12
<b>Drie voorbeelden van ondersteuning</b>	<b>14</b>
<i>eFormulieren</i>	14
<i>GemGids</i>	14
<i>Flamingo</i>	15
<b>Resultaten van het onderzoek</b>	<b>16</b>
<i>Wat gaat er goed?</i>	16
<i>Wat kan er beter?</i>	16
<i>Communities binnen overheidsorganisaties</i>	17
<b>Ondersteuning via communities</b>	<b>19</b>
<i>Wat te verwachten van een community?</i>	19
<i>Gebruikersondersteuning door andere gebruikers</i>	20
<i>Meedoen in een bestaande community</i>	20
<i>Het opzetten van een community</i>	21
<b>Conclusies</b>	<b>23</b>

## Inleiding

In overheidsorganisaties neemt software een heel belangrijke plaats in. Ambtenaren verwerken gegevens met software, data wordt opgeslagen in databases, of software zorgt voor een goed werkende website. De software die wordt gebruikt, wordt natuurlijk beoordeeld op prijs en kwaliteit; de software moet de taak vervullen waarvoor deze is bedoeld tegen redelijke kosten. Maar behalve dat, moet software ook goed te beheren zijn. Updates moeten bijvoorbeeld makkelijk verwerkt kunnen worden. Ook moet de dienstverlener problemen met de software snel kunnen oplossen. Kortom, de software moet goed ondersteund kunnen worden door de eigen organisatie. Zonder goede ondersteuning zullen veel organisaties niet eens overwegen de software te gebruiken.

Steeds meer overheidsorganisaties krijgen interesse in open source software. Dat is ook niet zo gek, want er zitten veel voordelen aan open source software. Er hoeft geen licentiegeld betaald te worden, er is minder leveranciersafhankelijkheid, het kan gemakkelijk worden aangepast en het kan gedeeld worden met andere overheden. Maar behalve deze voordelen ervaren veel overheidsorganisaties ook nadelen. Zo is er vooral op het gebied van ondersteuning veel twijfel en onzekerheid. Bij veel open source software is er niet een enkele dienstverlener aan te wijzen, zoals bij closed source software het geval is. En

het is juist de dienstverlener die een belangrijk deel van de ondersteuning op de software levert. Is het ontbreken van een dienstverlener dan een reden om open source software niet te gebruiken? Of zijn er andere manieren waarop goede ondersteuning op de software gewaarborgd kan worden? Over dit soort vragen gaat dit rapport.

## Leeswijzer

De belangrijkste vraag die in dit rapport wordt beantwoord is hoe ondersteuning op open source software voor overheidsorganisaties op een goede manier geregeld kan worden. Het zal blijken dat communities hierin een belangrijke rol spelen. Daarom zal ook dieper worden ingegaan op de rol van een community en de verschillende vormen van communities.

Op pagina 6 wordt uitgelegd wat ondersteuning op software nou precies is. Hierbij wordt theorie behandeld over ondersteuning, om zo een beter inzicht te krijgen in alle aspecten die bij ondersteuning van belang zijn.

Daarna wordt op pagina 8 de diversiteit van software, zoals maatwerk, off-the-shelf, gesloten en open software en waarom dit onderscheid van belang is bij ondersteuning.

Vanaf pagina 11 zal worden ingegaan hoe ondersteuning op verschillende soorten software traditioneel geregeld worden en wat waarde is van de theorieën over ondersteuning hierbij.

Om dit alles wat praktischer te maken worden op pagina 14 drie voorbeeld-

projecten beschreven. Deze projecten laten zien op welke verschillende manier ondersteuning geregeld kan worden.

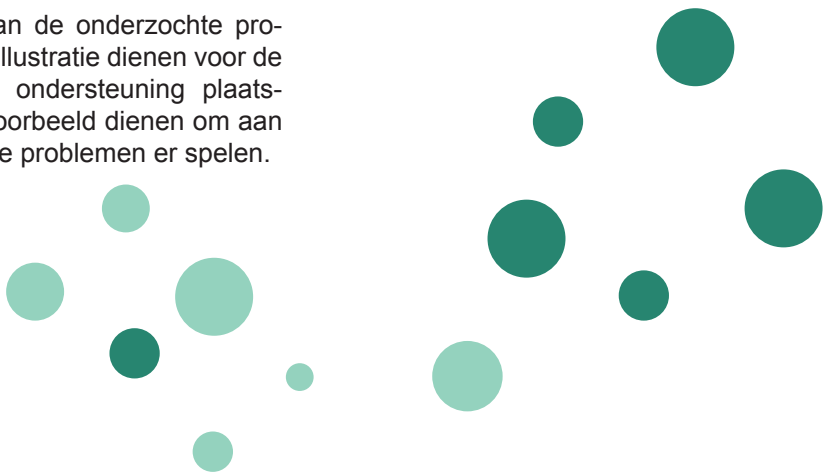
Vanaf pagina 16 zullen de feitelijke resultaten van het onderzoek worden behandeld. Deze resultaten leiden direct tot lessen die hieruit getrokken kunnen worden, met aanbevelingen voor overheidsorganisaties. Dit is te lezen op pagina 16.

Als laatste worden op pagina 22 de conclusies beschreven.

### *Het onderzoek*

Om de huidige stand van zaken met betrekking tot ondersteuning van open source software bij de overheid in kaart te brengen is een onderzoek uitgevoerd. Hiervoor zijn tien projecten binnen verschillende overheidsorganisaties bestudeerd. Hierbij is gekeken hoe deze organisaties omgaan met het feit dat de software open source is, hoe ze samenwerken met andere organisaties die de software ook gebruiken en hoe de ondersteuning is geregeld.

Een aantal van de onderzochte projecten zal als illustratie dienen voor de wijze waarop ondersteuning plaatsvindt en als voorbeeld dienen om aan te geven welke problemen er spelen.



## Ondersteuning theoretisch bekeken

Software moet niet alleen werken, het moet ook blijven werken. Hiervoor is ondersteuning (ook wel beheer genoemd) op de software essentieel. Alhoewel iedereen die met software te maken heeft wel een redelijk idee heeft wat ondersteuning precies inhoudt, is het toch nuttig dit duidelijk vast te stellen. Zo zijn er verschillende vormen van ondersteuning. Alle hebben betrekking op een net weer iets ander deel van de software of de bedrijfsvoering. Globaal gezien zijn er drie soorten van ondersteuning te onderscheiden: functioneel beheer, technisch beheer en applicatiebeheer. Alhoewel alle drie de vormen van beheer belangrijk zijn, gaat het bij de ondersteuning van individuele (open source) applicaties om applicatiebeheer. Dit wordt daarom verder uitgediept.

Applicatiebeheer heeft, zoals de naam al aangeeft, te maken met het beheer van individuele applicaties. Om een applicatie draaiende te houden is beheer nodig. Software moet onderhouden worden, zodat het wordt aangepast aan nieuwe situaties, bijvoorbeeld extra informatie die moet worden opgeslagen, of aanpassingen die nodig zijn om de software te laten werken met een nieuw besturingssysteem. Bijna alle software bevat wel fouten, die er natuurlijk zoveel mogelijk uit gehaald moeten worden. Daarnaast moet software ook geïnstalleerd en gecon-

figureerd worden. Tot slot moeten de eindgebruikers geholpen worden met vragen of problemen waarmee ze te maken hebben. Al deze zaken vallen onder ondersteuning. Bij ondersteuning op commerciële software worden deze zaken niet allemaal gedaan door de dienstverlener, meestal de maker van de software. Via service level agreements wordt vaak bepaald wat de commerciële partij levert aan ondersteuning en wat de organisatie zelf moet regelen.

### *Theorieën over ondersteuning*

Over softwareondersteuning zijn verschillende modellen ontwikkeld om ondersteuning beter te kunnen begrijpen.

Voor applicatiebeheer is er het Application Service Library, of ASL, model. Dit bestaat uit verschillende lagen. Acties in elke laag moeten goed geregeld worden om applicatiebeheer te laten slagen.

De onderste laag is de operationele laag. Het gaat er in deze laag om hoe de serviceorganisatie is geregeld en hoe ontwikkeling en onderhoud van de applicatie zijn geregeld.

De middelste laag is de tactische laag. In deze laag wordt beschreven aan welke tactische zaken aandacht moet worden besteed, zoals tijd, geld, arbeidsverdeling, kwaliteit, of zaken die met een serviceleverancier zijn geregeld. De hoogste laag is de strategische laag. Deze heeft betrekking op lange termijn beleid en op acties om

## Het ASL model



Het ASL model op globaal niveau. Op elk proces van nog verder worden ingezoomd. Op internet is veel meer te vinden over het totale ASL model.

ASL: <http://www.aslfoundation.org/>

(Bron: Van der Pols, 2001)

de organisatie als geheel flexibel te houden, bijvoorbeeld door niet te veel afhankelijk te worden van één enkele dienstverlener.

## Verschillende soorten software

### *Off-the-shelf- en maatwerksoftware*

Software is er in vele soorten en maten. Niet alleen de functies van software verschilt, maar ook de manier waarop het gemaakt en aangeboden wordt kan enorm verschillen. Een eerste vorm van software is algemene software die voor iedereen beschikbaar is. Dit is ook wel bekend als “off-the-shelf” software. Dit is software voor algemene taken en is voor veel gebruikers nuttig. Voorbeelden van dit soort software zijn het besturingssysteem, tekstverwerkers, spreadsheets en webbrowsers. Naast deze software die voor iedereen geschikt is, is er ook “maatwerksoftware”. Dit is software dat speciaal wordt ontwikkeld voor een bedrijf of organisatie. Het kan zijn dat een geheel nieuwe applicatie wordt ontwikkeld, of dat een bestaande applicatie wordt aangepast zodat het ingepast kan worden binnen een organisatie. Maatwerk-software zelf worden ontwikkeld door de organisatie die het gaat gebruiken, of wordt uitbesteed aan een gespecialiseerde softwareontwikkelaar.

### *Open source en diversie variaties van gesloten source software*

Behalve het verschil tussen maatwerk en off-the-shelf software kan software ook verdeeld naar soort licentie. Op software berust altijd auteursrecht. De schrijver is eigenaar en heeft het recht om de software te verkopen. De

eigenaar besluit wie de software mag gebruiken en onder welke voorwaarden. Voor gesloten software zal dit vaak betekenen dat de gebruiker zelf geen wijzigingen mag maken in de software. Om wijzigingen te kunnen maken is toegang en toestemming tot de broncode (“source code”) nodig. Als de broncode niet beschikbaar is, zijn wijzigingen zonder tussenkomst van de ontwikkelaar niet mogelijk.

Naast deze “gesloten source software” is er ook “open source software”. Hierbij is de broncode juist wel open en voor iedereen toegankelijk. Via een speciale open source licentie wordt geregeld wat wel en niet mag met de software. In de praktijk zorgt dit voor grote vrijheid met betrekking tot de software: deze mag geïnstalleerd, aangepast en verspreid worden. Dit brengt een aantal voordelen met zich mee, maar zorgt ook voor nieuwe uitdagingen. De voordelen en nadelen van open source software worden in de volgende paragraaf uitgebreid behandeld.

Geheel gesloten source en open source software lijken twee uitersten. Maar de voorwaarden bij gesloten source software kunnen heel divers zijn. Bijvoorbeeld als de broncode van de software wel toegankelijk en aanpasbaar is, maar alleen voor bepaalde organisaties. Of als de code is wel te bekijken is maar niet aangepast mag worden. Hoewel het in beide gevallen geen open source software is, maar duidelijk gesloten software, blijven wel



een aantal voordelen van open source software behouden.

### **Voordelen van open source software**

Er is al besproken dat open source software voordelen kan hebben. Het minder afhankelijk zijn van een dienstverlener is al genoemd als voordeel. In veel teksten worden kosten als belangrijk argument aangevoerd. Omdat open source software gratis is zouden de totale kosten lager zijn dan bij software die gekocht moet worden. Toch kunnen er een aantal vraagtekens gezet worden bij deze argumentatie. Vaak zitten de grootste kosten van software niet in de aanschaf, maar in de kosten voor ondersteuning, of kosten die gemaakt moeten worden om de software succesvol te kunnen invoeren, zoals aanpassingen, training, migratiekosten, etc. Software kan vaak niet direct geïnstalleerd worden, omdat er aanpassingen aangebracht moeten worden (in de software, of in de systemen). Hoewel kostenvoordeel vaak dus geen doorslaggevend argument is, is het wel een belangrijke aanzet om naar open source software te gaan kijken. Daarnaast zijn er nog andere grote voordelen.

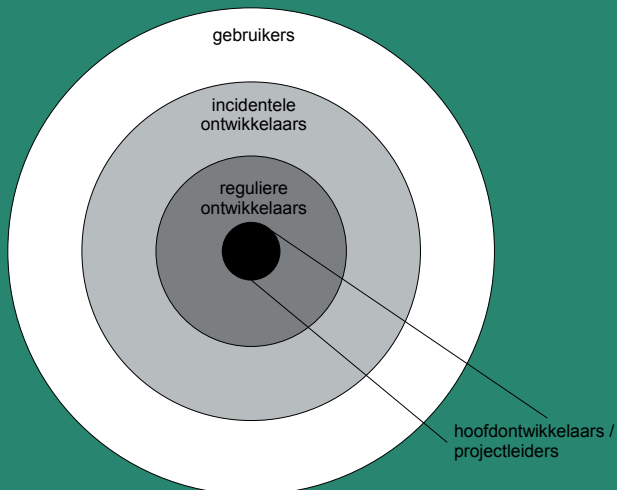
Software met een open source licentie kan heel gemakkelijk gedeeld worden met anderen, zonder dat er ingewikkelde contracten nodig zijn. Met open source software is kennisdeling tussen overheidsorganisaties makkelijker. Omdat open source software zelf

aangepast mag worden, is open source software beter aan een bedrijfsproces aan te passen. Met open source software kan de software aangepast worden aan de rest van het systeem, in plaats van het systeem aan de software.

Daarnaast is er bij open source software minder sprake van leveranciersafhankelijkheid. Juist de licentie maakt het mogelijk om over te stappen naar een andere leverancier, maar toch dezelfde software te blijven gebruiken.

Voor overheidsorganisaties zijn er nog een aantal andere redenen om gebruik te maken van open source software. Door eigen software uit te geven onder een open source licentie wordt code teruggegeven aan de burgers en bedrijven; het is immers betaald met publiek geld. Ook kan het gebruik van open source software een stimulans zijn voor de lokale economie. Het is niet meer nodig om software en ondersteuning te kopen van buitenlandse bedrijven. Het geld kan nu worden geïnvesteerd in de lokale economie door extra ontwikkeling uit te laten voeren door lokale bedrijven.

## Het ui model



Binnen een community vervullen verschillende gebruikers een verschillende rol. In het algemeen bestaat een community uit een of meerder projectleiders en/of hoofdontwikkelaars. Deze personen zorgen voor een belangrijk deel voor nieuwe functionaliteit, de roadmap, en het uitbrengen van nieuwe versies. Een iets grotere groep bestaat uit reguliere ontwikkelaars. Zij maken op regelmatige basis nieuwe functionaliteit, die vervolgens door de leiders kan worden toegevoegd aan het programma. Een nog wat grotere groep zijn actieve gebruikers. Zij gebruiken niet alleen de software, maar testen ook nieuwe versies en rapporteren fouten. De grootste groep betreft echter de reguliere gebruikers. Ze hebben weinig van doen met de ontwikkeling of het testen, maar zullen wel vragen stellen over het gebruik van de software (en waar mogelijk vragen van andere gebruikers beantwoorden).

Grafisch kan dit worden weergegeven als een ui met verschillende ringen. Naarmate men van de kern af beweegt neemt het aantal personen toe, maar neemt de betrokkenheid van personen bij de community af.

(Bron: Van Wendel de Joode, 2005)

## Ondersteuning op software

### *Ondersteuning op gesloten software*

Voor gesloten software is er vaak maar één partij die de ondersteuning kan verzorgen: de fabrikant van de software. Of het nou maatwerk betreft of off-the-shelf software, alleen de fabrikant heeft beschikking over de broncode en kan veranderingen aanbrengen. Wil een organisatie ondersteuning hebben dan moet er zaken worden gedaan met die fabrikant.

### *Communities en open source software*

Het feit dat open source software voor iedereen beschikbaar is creëert een aantal bijzondere mogelijkheden. Omdat de broncode voor iedereen beschikbaar is kan iedereen verbeteringen aanbrengen. Bijvoorbeeld in de vorm van nieuwe mogelijkheden of bij het oplossen van fouten. Deze verbeteringen kunnen dan worden gedeeld met alle andere personen of organisaties die de software gebruiken. Op deze manier kan iedereen ervan meeprofiteren.

Personen en organisaties die open source software gebruiken kunnen samen een community vormen. Dit is een samenwerkingsverband tussen alle gebruikers. Het begrip gebruiker moet vrij breed worden opgevat: zowel eindgebruikers als (commerciële) ontwikkelaars zijn onderdeel van de community. Of een community een vast,

of juist los verband heeft, verschilt per softwareproject. Binnen een community kunnen gebruikers verschillende rollen vervullen. Sommige gebruikers zijn actief bij de ontwikkeling van de software, andere houden zich bezig met de website. Weer andere gebruikers testen de software en komen met ideeën voor verbeteringen, en er zijn gebruikers die vragen van andere gebruikers beantwoorden en natuurlijk gebruikers die vragen stellen. Gebruikers van open source software kunnen zich dus zelf organiseren en zelf zorgen dat de ontwikkeling van de software wordt doorgezet.

### *Ondersteuning op open source software*

Er is geen relatie tussen de functionaliteit van software, wie het maakt en de licentie. Ook bij veel open source software zijn commerciële softwarebedrijven betrokken. De software zelf is wel altijd gratis beschikbaar. Bedrijven kunnen geld verdienen door ondersteuning, of door andere diensten rondom de software aan te bieden.

In eerste instantie is er dus weinig verschil tussen ondersteuning op gesloten en open source software. In beide gevallen is er een maker die ondersteuning kan verzorgen. Toch is er een heel belangrijk verschil: de aanbieder van open source software heeft geen alleenrecht. Hoewel de ontwikkelaar waarschijnlijk het meeste van de software afweet staat het elk bedrijf vrij om aanpassingen op de software te maken en ondersteuning commer-

cieel aan te bieden aan klanten. Zo'n nieuw bedrijf kan ook ondersteuning leveren op de software. In theorie zit een overheidsorganisatie dus niet vast aan een enkele dienstverlener. Dit hoeft echter niet altijd te betekenen dat er voor elk open source product automatisch meerdere partijen zijn die ondersteuning kunnen leveren. Dit zal ook in sterke mate afhangen van de vraag vanuit de markt.

Maar bij open source software kan een deel van de ondersteuning ook nog uit een andere hoek komen: de community. Het is in het belang van de community dat zoveel mogelijk mensen de software gebruiken. Hoe meer mensen het gebruiken, hoe groter de kans dat de community en dus de software blijft voortbestaan. Bij vragen of problemen is het dus in het belang van de community als geheel dat het probleem voor die gebruiker wordt opgelost. Iemand uit de community zal waarschijnlijk tijd besteden aan het beantwoorden van de vraag, of tijd besteden aan het verbeteren van de software. Maar het beantwoorden van vragen is voor alle deelnemers vrijwillig, dus echte zekerheden zijn er niet. Dat neemt niet weg dat voor echt grote communities de ondersteuning die zo'n community kan leveren tamelijk zeker is.

### *Open source software en het ASL model*

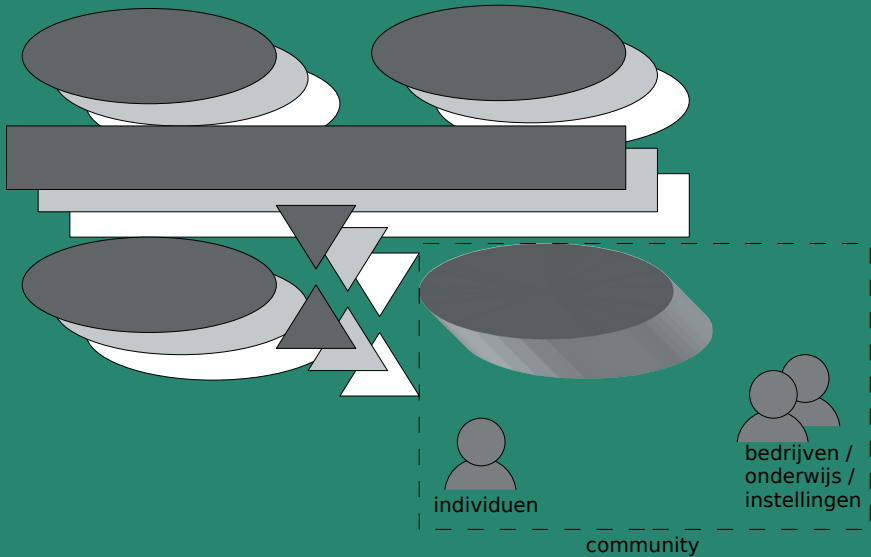
Eerder is het Application Service Library model gepresenteerd als een model

voor ondersteuning op software. Dit model is niet alleen geschikt voor gesloten source software, maar kan ook voor open source software gebruikt worden. Als het model op open source software wordt toegepast valt op dat de verschillen in ondersteuning tussen open source en gesloten source software helemaal niet zo groot zijn.

Veel van de acties die het ASL model voorschrijft zijn acties die een overheidsorganisatie zelf moet nemen en waar geen hulp van een community bij kan komen. Over bijvoorbeeld lange termijn strategie zal door de organisatie zelf nagedacht moeten worden.

Hiernaast is het ASL model nog een keer weergegeven, maar nu voor drie organisaties die hetzelfde softwareproduct gebruiken. Alle drie de organisaties hebben op een eigen manier ondersteuning geregeld via de procedures van het model. Maar juist omdat het open source software betreft kan er samenwerking plaatsen vinden en kunnen bepaalde acties gedeeld worden. Dit zal voornamelijk zitten in de ontwikkeling en onderhoud van de applicatie. Hoewel het drie verschillende organisaties zijn, zijn ze toch met elkaar verbonden. Hierin is ook direct zichtbaar welke plaats die community inneemt binnen organisaties. De community als geheel kan zorgen voor de ontwikkeling en onderhoud van de applicatie. Belangrijk is wel dat voor elke organisatie geldt: "de community ben je zelf", dus elke organisatie moet zich voor ondersteuning inzetten. Niet

## De community



De totale community bestaat uit overheidsorganisaties die de software gebruiken, bedrijven die aan de software werken, en eventueel personen die vrijwillig aan de software meewerken.

In dit plaatje zijn er drie overheidsorganisaties (elke organisatie heeft een andere kleur) die dezelfde software gebruiken. Taken uit het ASL model moeten door de organisatie zelf worden uitgevoerd. Zo moet bijvoorbeeld elke organisatie zelf nadenken over wat er moet software gedaan moet worden en wat de toekomstplannen zijn. Maar er kan wel samenwerking plaatsvinden tussen organisaties. Dit vindt plaats voor de taak 'onderhoud & ontwikkeling' (rechts onder). Organisaties kunnen op deze taak goed samenwerken en ontwikkeling- en onderhoudstaken delen. Ook bedrijven die meewerken aan de software, of zelfs individuen kunnen meewerken aan de ontwikkeling en onderhoud van de software. Op dit punt wordt dus een *community* gevormd, zodat een deel van de ondersteuning tussen meerdere partijen gedeeld kan worden.

alleen de eigen ondersteuning, maar ook die van anderen. Hiermee wordt de eigen ondersteuning alleen maar beter.

uitmaken van de community: commerciële bedrijven en zelfs individuen kunnen zinnige bijdragen leveren aan ontwikkeling en onderhoud van een open source applicatie.

Behalve andere overheidsorganisaties kunnen ook andere partijen deel

## Drie voorbeelden van ondersteuning

Bij open source software is er niet één manier waarop ondersteuning altijd goed geregeld kan worden. Veel hangt af van het soort software, waar de software vandaan komt, welke ontwikkelaars en gebruikers er zijn, en hoe de community rondom het project functioneert.

Om dit duidelijk te maken zullen drie van de onderzochte projecten kort worden toegelicht, en wordt duidelijk dat ondersteuning op verschillende manieren geregeld kan worden. Een van de projecten heeft geen open source licentie, maar is een zogenaamde gated community. Toch is juist dit project ook onderzocht omdat veel voordelen van open source software hierin terugkomen en het belangrijke inzichten geeft.

### *eFormulieren*

(leverancierondersteuning)

eFormulieren is een programma van ICTU. Het doel van eFormulieren is een platform te creëren waarmee formulieren kunnen worden gemaakt die worden getoond in een webbrowser. Via deze formulieren kunnen burgers makkelijk communiceren met gemeentes en centrale overheidsinstanties. Dit zijn onder andere formulieren voor het doorgeven van een verhuizing, het indienen van een bezwaarschrift, of het doorgeven van schade. Via DigiD kan er worden ingelogd, zodat een aantal gegevens (naam, adres) al standaard

zijn ingevuld.

eFormulieren is gebaseerd op ePlatform, een open source ontwikkelplatform van LogicaCMG. eFormulieren wordt dan ook in nauwe samenwerking met LogicaCMG ontwikkeld. Omdat LogicaCMG veel expertise heeft op het gebied van ePlatform – en dus ook van eFormulieren, dat daar sterk op lijkt – is het logisch LogicaCMG in te schakelen wanneer er wijzigingen of onderhoud gedaan moet worden aan de applicatie. Hoewel de applicatie een open source licentie heeft en iedereen er wat mee kan doen, is er op dit moment in feite maar één partij die ondersteuning biedt.

### *GemGids*

(gated community)

GemGids is een bibliotheek voor applicaties waarmee via kaarten (o.a. Google Maps) informatie over een gemeente kan worden opgevraagd. Via een luchtfoto zijn bijvoorbeeld alle bouwvergunning op te vragen in een bepaald gebied, of kan de rioolafsluiting van een pand worden bekeken. GemGids maakt gebruik van het concept gated community. De applicatie is dus niet open source, maar is alleen toegankelijk voor instanties die zich bij de community hebben aangesloten. Hiervoor moet een bepaald bedrag betaald worden. Voor dit bedrag krijgt de instantie het recht om GemGids te gebruiken en krijgen ze via de bedrijven die in GemGids participeren ondersteuning. Wil een instantie iets extra's met GemGids dan kan ze

een van de deelnemende bedrijven hiervoor de opdracht geven. Voor het uitvoeren van de opdracht wordt gewoon aan het bedrijf betaald. Het werk is dan niet alleen beschikbaar voor de instantie die de opdracht heeft uitgeschreven, maar voor alle deelnemers van GemGids. Voor een relatief laag jaarlijks bedrag krijgen de deelnemers dus steeds de nieuwste mogelijkheden.

### *Flamingo*

(community ondersteuning)

Flamingo is een eenvoudige kaartviewer. Oorspronkelijk opgezet als een applicatie om risicokaarten te tonen op de websites van provincies, kan Flamingo nu veel meer. Flamingo is aan de sluiten op diverse geo-databases. De verschillende provincies vormen nog steeds het hart van de Flamingo groep. Het grootste deel van de ontwikkeling wordt gedaan door de provincie Friesland. De overige provincies zijn wel betrokken bij de ontwikkeling, maar voornamelijk door het geven van ideeën. Bij Flamingo is verder geen commerciële partij betrokken. De deelnemers zorgen zelf voor ondersteuning en doorontwikkeling. Hiervoor is een speciale website opgezet. Op deze website is de software te downloaden, is er een forum voor vragen en kunnen extra uitbreidingen op de software worden geplaatst. Organisaties die de software gebruiken geven dus zo ondersteuning aan de andere gebruikers. Alle gebruikers vormen samen een community.



## Resultaten van het onderzoek

Binnen de Nederlandse overheid zijn tien projecten bestudeerd die zelf een open source licentie hebben, veel gebruik maken van open source, of (als ze geen open source licentie hebben) veel van de voordelen van open source software vertonen. Bij alle projecten is onderzocht hoe de ondersteuning is geregeld, en hoe deze wordt ervaren. Ook is er gekeken of er een community is en hoe er dan met deze community wordt omgegaan. Het doel van het onderzoek is om huidige ondersteuning van open source software binnen de overheid in kaart te brengen en mogelijke verbeterpunten aan te dragen.

### *Wat gaat er goed?*

Uit het onderzoek is naar voren gekomen dat alle organisaties die werken met open source software hun ondersteuning op een goede manier hebben geregeld. Voor deze organisaties is het geen probleem om ondersteuning op de gewenste manier te regelen. Dat kan zijn via een commerciële partner, of via communities.

De organisaties gaven allemaal aan dat er zeker wel een rol voor communities is. Communities zouden vooral een rol kunnen spelen bij kennisuitwisseling en de doorontwikkeling van de software. Maar juist over communities bestaan grote onzekerheden. Het is alleen niet precies bekend hoe een community tussen verschillende over-

heidsinstanties kan worden gecreëerd, of wat er precies te verwachten valt als er eenmaal een community is opgestart.

### *Wat kan er beter?*

Binnen de projecten die zijn onderzocht, zijn een aantal opmerkelijke zaken geconstateerd die het gebruik van open source software, en het werken met communities, enigszins negatief kunnen beïnvloeden.

Binnen veel van de organisaties die onderzocht zijn heerst nog een redelijke onzekerheid over open source software. Vooral over wat open source software kan en wat er te verwachten valt van open source software en de community rondom een open source product. Via deze brochure en andere informatie van OSOSS moet in ieder geval een deel van die onzekerheid worden weggenomen.

Daarmee samenhangend is er binnen veel overheidsorganisaties nauwelijks spraken van een 'open source cultuur'. Het is zeker niet nodig dat alle overheidsorganisaties volledig omschakelen naar open source software, maar bij sommige organisaties heerst zelfs een bijna vijandige sfeer ten aanzien van open source software. Reden hiervoor kan zijn dat de precieze werking van open source software en communities nog onbekend is.

Ook zijn veel medewerkers niet op de hoogte van het feit dat hun organisatie open source software gebruikt, of



dat ze zelf met open source software werken. Zonder deze kennis, en het besef waarom, zullen ze nooit echt tot de community behoren, iets wat wel gewenst is.

Een derde punt is dat veel producten niet geheel open source zijn. Hiervoor kunnen goede redenen worden aangedragen – bijvoorbeeld in het geval van een gated community – maar vaak zijn er nauwelijks goede redenen. Sommige projecten bijvoorbeeld willen alleen code delen met andere overheidsorganisaties. Individuen en commerciële organisaties worden bij voorbaat uitgesloten. Hierdoor wordt de kans dat de community zich kan ontwikkelen tot een succesvolle community kleiner.

Als laatste worden veel open source projecten die door een externe partij zijn uitgevoerd alleen ondersteund door die ene partij. Er wordt geen tijd in gestoken om ook andere ontwikkelaars bij het product te betrekken. Door dit na te laten, blijft de originele ontwikkelaar grote invloed houden en worden een aantal van de voordelen van open source software minder benut.

### *Communities binnen overheidsorganisaties*

Binnen de organisaties die onderzocht zijn is nog niet duidelijk sprake van grote, volwassen communities. Wel zijn er een aantal kleinere communities en veel initiatieven voor het opzetten van communities. Binnen de

overheid zijn er andere vormen van communities dan de communities die bekend zijn van grote open source projecten.

Globaal gezien zijn er drie verschillende organisatievormen: de formele community, de informele community en de gated community. De formele community heeft een duidelijke leider (dit kan ook een organisatie zijn) en deze leider probeert actief nieuwe leden bij de community te betrekken. Dit zal vaak het geval zijn als er op centraal niveau software wordt gemaakt dat bedoeld is voor decentrale overheden. De centrale organisatie zal dan actief op zoek gaan naar decentrale overheden om te participeren in de community.

Binnen informele communities is er ook wel een leider, maar hier wordt minder actief gezocht naar nieuwe leden. Nieuwe leden sluiten zichzelf aan omdat het een duidelijk voordeel is om lid te zijn van de community. Zelforganisatie is hier erg belangrijk.

Als laatste zijn er gated communities. Deze communities zijn gesloten voor buitenstaanders, maar binnen de community wordt wel alles gedeeld. Er kunnen dan ook eisen worden gesteld voordat een organisatie wordt toegelaten tot de community, bijvoorbeeld het betalen van een bepaald bedrag. Omdat niet iedereen toegang heeft is de software geen open source software. Maar omdat het tussen de leden wel alle eigenschappen heeft die ook bij

## Voordelen en nadelen van community-vormen

### Gated community

#### Voordelen:

- Omdat alleen leden toegang hebben tot de code, is het duidelijk wie de software gebruikt
- Door een vergoeding te vragen kan van deze vergoeding ondersteuning voor alle leden betaald worden

#### Nadelen:

- De ontwikkeling is gesloten voor buitenstaanders, dus er zullen minder snel nieuwe leden bijkomen. Hierdoor zijn er minder potentiële ontwikkelaars

### Formele community

#### Voordelen:

- Er is een duidelijke leider binnen de community
- De leider heeft er een belang bij dat de community blijft voortbestaan, wat veel zekerheid geeft voor alle leden

#### Nadelen:

- Leden zullen heel snel naar de leider kijken en niet snel zelf taken op zich nemen

### Informele community

#### Voordelen:

- Leden organiseren zelf een community en bepalen dus zelf wat het beste is voor hun eigen community

#### Nadelen:

- Inzet hangt af van de motivatie van leden. Als te weinig leden gemotiveerd zijn om de software te ontwikkelen vormt dit een gevaar voor de hele community

open source software horen, blijven veel voordelen behouden.

Elke vorm heeft zijn eigen voor- en nadelen. Die voor- en nadelen staan hierboven in de tabel uitgelegd.

## Ondersteuning via communities

*Wat te verwachten van een community?*

Communities kunnen van groot nut zijn binnen de ondersteuning van de software. Een community kan zorg dragen voor de verdere doorontwikkeling van een product. Dit kan al bij zeer kleine communities. In feite vormt een organisatie die een applicatie heeft laten ontwikkelen al een community met de ontwikkelaar als er na oplevering contact blijft. Als er daarna meerdere organisaties bij komen groeit de community. Elke organisatie die de software gebruikt, ideeën levert of meedoet aan de ontwikkeling maakt deel uit van de community. Daarnaast kan een community nuttig zijn om ervaringen te delen. Bijvoorbeeld over het gebruik van de software, of andere software die gebruikt wordt.

Wat een community voor een organisatie kan doen hangt af van de plaats van de eigen organisatie in die community. Om dit duidelijk te maken moet het ui model weer in herinnering worden genomen. Organisaties kunnen zich in het midden van de community bevinden, of meer aan de rand. In beide gevallen is het zowel geven als nemen. Doordat iedereen zowel geeft als neemt ontstaat er een eerlijke community.

Als de eigen organisatie zich in het midden van de community bevindt, zal er veel aan ontwikkeling worden ge-

daan. Aan organisaties aan de buitenkant wordt dan de software geleverd. Andere organisaties zullen zich verder van het midden bevinden. Deze organisaties zullen vooral ideeën aandragen, vragen stellen en misschien af en toe iets aan ontwikkeling doen. Voor de organisatie in het midden zijn deze ideeën en vragen belangrijk, omdat het aangeeft wat de eventuele problemen met de software zijn. Dat helpt om de software te verbeteren.

Als de eigen organisatie zich meer aan de buitenkant van de community bevindt, zal het vooral vragen om ondersteuning. In eerste instantie zullen vooral organisaties die zich meer in het midden bevinden aan deze ondersteuning beantwoorden. Maar naarmate er meer kennis komt binnen de eigen organisatie, kunnen ook andere organisaties aan de rand van de community worden geholpen. Nieuwe organisaties kunnen zo door de andere organisaties aan de rand van de community worden geholpen.

Een community ontstaat niet vanzelf. Het is een illusie om te verwachten dat elke open source applicatie meteen andere overheidsorganisaties, vrijwilligers en bedrijven zal aantrekken die zorg gaan dragen voor de ontwikkeling. Een community zal opgezet moeten worden en er moet aandacht worden besteed aan de groei en ontwikkeling van de community.

### *Gebruikersondersteuning door andere gebruikers*

Communities bij grote open source projecten zorgen niet alleen voor de ontwikkeling en onderhoud van de software, maar ze zorgen ook voor het beantwoorden van vragen van andere gebruikers. Op forums kunnen gebruikers vragen stellen, waarop andere gebruikers antwoorden kunnen geven. Het is in het belang van de gehele community dat dit gebeurt. Een project is gebaat bij veel gebruikers, maar daarvoor moeten gebruikers in het begin soms wel een beetje geholpen worden. Anders bestaat er de kans dat potentiële gebruikers er niet zelf uitkomen en de software verder links laten liggen. Aan het aantal vragen kunnen de ontwikkelaars ook zien waar problemen met de software liggen en ideeën opdoen hoe dit verbeterd kan worden. Bovendien is elke nieuwe gebruiker een potentiële ontwikkelaar: een gebruiker kan langzaam doorgroeien van gebruiker, naar actieve gebruiker, naar incidentele ontwikkelaar, en misschien zelfs naar reguliere ontwikkelaar.

Niet elk open source project zal op een even goede manier ondersteuning kunnen geven aan alle gebruikers. Hiervoor is het nodig dat er genoeg gebruikers, met voldoende kennis zijn die tijd willen steken in het beantwoorden van vragen van andere gebruikers. Dit kan alleen bij een project van redelijke omvang. Voor grote, bekende open source projecten is

die omvang zeker al aanwezig. Overheidsorganisaties kunnen hier gebruik van maken. Commerciële ondersteuning kan waarschijnlijk nooit helemaal vervangen worden, maar voor simpele vragen is het heel goed mogelijk om van de diensten gebruik te maken van de community.

### *Meedoen in een bestaande community*

Als er bestaande open source software wordt gebruikt, is het verstandig mee te doen aan de community. Op deze manier krijgt de organisatie meer kennis van de software die gebruikt wordt. Deze kennis kan later goed van pas komen.

Wanneer er vragen zijn over de software, kunnen op bijvoorbeeld het forum van het project vragen worden gesteld. Als er vragen door andere personen worden gesteld waarop het antwoord bekend is, kan men deze vragen beantwoorden. Het voordeel hiervan is dat men als persoon een bepaalde status verwerft binnen de community. Veel van wat binnen een community gebeurt, gebeurt op basis van wederkerigheid: personen die veel voor de community doen, kunnen ook veel van de community terug verwachten.

Behalve met het beantwoorden van vragen is het ook mogelijk om op een hoger niveau mee te doen met een community, bijvoorbeeld bij de ontwikkeling van de software. Doordat iemand van de eigen organisatie

een belangrijk lid is van de community, krijgt de organisatie ook invloed op de richting van de ontwikkeling. Daarnaast krijgt deze persoon heel veel kennis, die ook goed gebruikt kan worden bij de interne ondersteuning. Zo'n oplossing hoeft voor de organisatie niet eens duur te zijn. Iemand een halve dag per week laten meedraaien kan al snel goedkoper zijn dan licentie- en ondersteuningskosten van gesloten software.

### *Het opzetten van een community*

Als een open source applicatie door meerdere partijen wordt gebruikt is het creëren van een community een goede stap. Er zijn zeer veel factoren die het succes van een community kunnen beïnvloeden. Hoe een community georganiseerd is zal per community verschillen. Grote communities van bekende open source producten zullen anders in elkaar zitten dan communities van software die bij enkele organisaties wordt gebruikt. Het direct kopiëren van een community van een succesvol open source product zal dan ook niet goed werken. Het is belangrijk dat een community past bij het product. Terwijl de software wordt doorontwikkeld moet ook de community zich ontwikkelen. Een community kan zich zo vormen dat zij het beste past bij het product dat ondersteund wordt. Maar hoe moet dan worden begonnen met het opzetten van een community?

Allereerst moet er bepaald worden hoe de community georganiseerd moet worden. Plannen voor het opzetten van een community hoeven niet gedetailleerd van te voren gemaakt te worden. Een van de belangrijkste aspecten van communities is dat ze in hoge mate zelforganiserend zijn. Naarmate een community groeit, groeit een community toe naar een vorm die het best bij een community past. Daarvoor is het wel nodig dat een nieuwe community de gelegenheid krijgt zichzelf te vormen. Van te voren te veel vastzetten, zal leiden tot een te rigide structuur waarin nauwelijks sprake kan zijn van het zelf zoeken naar de juiste vorm.

Het is wel goed om op voorhand na te denken over de vorm van de community, dus een gated community, een formele of informele community. Deze keuze hoeft niet blijvend te zijn. Een gated community kan besluiten om iedereen toe te laten en zo een open community te worden. Leaders in een formele community kunnen meer taken delegeren en zo meer gaan lijken op een informele community.

In een community kunnen zowel overheidsorganisaties als bedrijven een plaats hebben. Een bedrijf dat zorgt voor ondersteuning voor een van de overheidsorganisaties zal dus ook deel uitmaken van de community. In een community, bedoeld voor overheidsorganisaties, zullen dus niet alleen andere overheidsorganisaties zitten.

• Het opzetten van een community kan

zowel door de overheidsorganisatie als door bedrijven worden gedaan. Beide zullen voordeel hebben van een community, dus het is van belang dat beide partijen vanaf het begin samenwerken en samen zorgen dat er een succesvolle community ontstaat. Mocht één van die partijen niet meewerken, dan doet de andere partij er goed aan om zelfstandig te zorgen voor nieuwe organisaties die de software willen gebruiken en onderdeel willen worden van de community.

## Conclusies

Open source is eigenlijk niet zo anders als andere soorten software. Het feit dat de broncode van de software open is heeft niet zo heel veel implicaties voor ondersteuning. De eisen die gesteld worden aan ondersteuning veranderen niet bij het gebruik van open source software.

Voor bekende open source producten zijn er op de markt ook voldoende bedrijven die aanvullende ondersteuning op de software kunnen leveren. Hiermee kan open source software op exact dezelfde wijze binnen een organisatie ondersteund worden als closed source software.

Uit het onderzoek is gebleken dat de voordelen van open source software niet altijd optimaal worden benut. Veel organisaties maken zich afhankelijke van één commerciële ondersteuner voor beheer of doorontwikkeling. Hoewel het risico om geheel afhankelijk te worden van die ene dienstverlener kleiner is dan bij closed source software, bestaat dat risico wel degelijk. Het is daarom aan te bevelen om al in een vroeg stadium meerdere dienstverleners te betrekken bij het project. Hierdoor krijgen ook andere bedrijven ervaring met het product en kan er bij het uitvoeren van nieuwe opdrachten gekozen worden uit meerdere partijen. Daardoor wordt de afhankelijkheid van één enkele dienstverlener voorkomen. Door concurrentie tussen de dienstverleners kan onderhandeld worden over een opdracht, wat de

prijs waarschijnlijk ten goede komt.

Rondom bijna alle open source software projecten bestaan er communities. Deze communities kunnen een zeer nuttige bijdrage leveren aan de ondersteuning. Dit heeft met name betrekking op de ontwikkeling en het onderhoud van de software. Een organisatie kan meedraaien in een bestaande community, of kan een nieuwe community opzetten met organisaties die eenzelfde product gebruiken. Het is het in het belang van elke (overheids)organisaties om deel uit te maken van de community van de open source software die gebruikt wordt. Door (actief) deel uit te maken van de community, kan de organisatie invloed uitoefenen op de richting van het project.

Bij veel open source projecten binnen de overheid is ondersteuning redelijk goed geregeld. Projecten zijn goed in staat om ondersteuning op de gewenste manier te regelen. Nog niet alle projecten bezitten een goed functionerende community rondom het project. Voor elk project kan het veel voordelen opleveren om een community op te zetten met de organisaties die ook de software gebruiken. Hoe een community opgezet en ingericht moet worden verschilt per project. Er is niet slechts één enkele succesformule die het project kan veranderen in een succesvol open source project met een actieve community. Er zijn verschillende soorten communities mogelijk. Drie brede categorieën zijn

genoemd als de informele, de formele, en de gated community. Maar ook hierin bestaan weer vele variaties. Dit grote aantal variaties hoeft niet te leiden tot nog meer onzekerheid over open source software en communities. Communities zijn in hoge mate zelf-organiserend. Wanneer er niet een te rigide structuur is zal een community doorgroeien naar een vorm die het beste bij het project past. Omdat elk project anders is zal ook elke community er anders uitzien.

Van groot belang is dat de 'cultuur' binnen de deelnemende organisaties goed moet zijn, zodat een project zich optimaal kan ontwikkelen. Hiervoor is nodig dat mensen weten wat open source software is, dat medewerkers weten dat ze open source software gebruiken – en vooral waarom hun organisatie gebruik maakt van open source software. Ook moeten mensen enige vrijheid hebben in het omgaan met de software. Iedereen, van hoog tot laag moet de kans (en ook de tijd) krijgen om zich desgewenst met de community in te laten. Door medewerkers te laten participeren bij de ontwikkeling en het onderhoud van de software kan uiteindelijk ondersteuning op een betere manier geregeld worden en zal er betere software ontstaan.

## Dankwoord

De volgende mensen en projecten hebben meegewerkt aan dit onderzoek. Niet alle projecten zijn beschreven, maar zonder de interviews met deze personen was dit onderzoek niet mogelijk geweest.

Bart Kerver, A-Select  
 Carien van Leeuwen, GovUnited  
 Herman de Haan, Gemeente Delft  
 Holger Peters, GemGids  
 Michiel Bouten, Modernisering GBA  
 Piet Feddema, InterWad  
 Pim Schouten, eFormulieren  
 Ron Wardenier, Flamingo  
 Ronald Houtsma, DigiD  
 Ted van Geest, MMPProject

En daarnaast voor de samenwerking bij dit onderzoek:

Maarten Wijnen-Meijer, OSOSS

## Gebruikte bronnen

Fogel, K. (2006). Producing open source software. <http://producingoss.com/>

Van der Pols, R. (2002). ASL. Een framework voor applicatiebeheer. Den Haag: ten Hagen Stam.

Van Wendel de Joode, R. (2005). Understanding open source communities. An organizational perspective. Delft: Technische Universiteit Delft.

Vijge, D.S. (2007). Organising support on open source software in government organisations. Eindhoven: Technische Universiteit Eindhoven.