

KASSA!

Zaken doen met vrije software

Alhoewel vrije software meestal gratis wordt verspreid, kun je er prima geld mee verdienen. En in tegenstelling tot bedrijven als Red Hat hoeft je ook geen miljoenenomzet te draaien om hier een levensvatbaar businessmodel van te maken.

Eerst wat geschiedenis. In 1969 koppelde IBM de verkochte hardware nadrukkelijk los van de software en zo gaf het bedrijf ruimte aan wat nu een miljardenindustrie is. Voor die tijd werd programmatuur bij de dure mainframes geleverd en pro-

grammeurs konden deze vrijelijk aanpassen en uitbreiden. Niet iedereen was in zijn nopjes met deze ontwikkeling. Richard Stallman startte begin jaren tachtig de vrije-softwarebeweging en zette zijn gedachten in 1985 uiteen in het GNU Manifesto. In dit manifest beschreef hij

niet alleen de ideologie achter softwarevrijheid, maar ook de mogelijkheden om hier geld mee te verdienen. Hij verwierp de gedachte dat dit programmeurs tot de bedelstaf zou veroordelen en nam nadrukkelijk de ruimte om aan te geven dat hij de free in free software bedoelde als in

free speech, niet als in free beer.

HONDERDVIJFTIG DOLLAR Stallman bracht meteen een businessmodel in de praktijk. Hij verkocht de broncode van de teksteditor GNU Emacs. Deze deed hij voor honderdvijftig dollar op de post, waarna de

ontvanger een tape thuiskreeg. Maar het duurde nog even voordat de commercialisatie van vrije software een hoge vlucht nam. Dit gebeurde in 1989, toen het Amerikaanse Cygnus Solutions werd opgericht. Michael Tiemann, John Gilmore en David Henkel-Wallace boden hiermee

commerciële ondersteuning voor vrije software en wisten een succesvol bedrijf op te zetten rond de support en ontwikkeling van de GNU-tools voor programmeurs. In de jaren negentig groeide het aantal bedrijven dat hun diensten aanbood rond open

source. Met name het succes van de Linux-kernel, gecombineerd met de webserver Apache en het GNU-besturingssysteem, speelde hier een grote rol in. Het stelde mensen in staat om veel goedkoper een webserver te kunnen neerzetten dan voorheen. In 1998 werd de term 'open source' gemunt om de aandacht wat af te leiden van de meer ideologische aspecten van vrije software. Men wilde zich meer richten op de praktische kant ervan. Dit zette de deuren open naar brede vermarkting van vrije software. Red Hat, een bedrijf met zijn eigen GNU/Linux-distributie, ging in 1999 uiterst succesvol naar de beurs, gevolgd door de recordbrekende beursgang van VA Linux Systems, dat voorgeïnstalleerde Linux-computers verkocht. Ondanks het barsten van de internetbel bleef het aandeel van open source- en vrije software in de jaren hierna stijgen. Met deze ontwikkeling is er ook een overvloed aan businessmodellen ontstaan.

VOORDELEN EN VALKUILEN

Voor freelancers of directeurs van softwarebedrijfjes die de open source-richting willen inslaan, zetten we de voordelen en valkuilen van de verschillende businessmodellen uiteen. Het belangrijkste om je te realiseren is dat je – in verhouding tot de traditionele software-verkopers – het accent van je arbeid verlegt. In plaats van het product komen de gebruiker, zijn beleving en zijn wensen centraal te staan.

De klassieke manier om geld te verdienen met open source is door support te verkopen. Deze support kun je bieden rond producten die je niet zelf hebt ontwikkeld. Dit is een uitstekende manier om je geld te verdienen. Je kunt eenvoudig een content-beheersysteem als WordPress, Drupal of Joomla! gebruiken. Dit vormt dan de basis voor het inrichten van de websites van je klanten. De support die je dan levert, is het tweakken naar de smaak van je opdrachtgever en

het passend maken in de huisstijl. Deze aanpak komt erg veel voor en is vaak zo vanzelfsprekend dat gebruikers ervan niet eens doorhebben dat ze feitelijk een honderd procent open source-bedrijfje runnen.

Het voelt als programmeur ook vrij logisch om uit te gaan van bestaande code. Dan hoeft je minder zelf te programmeren en onder de streep ook minder door te berekenen aan je klanten. Ook zit je klant niet aan jou vast als ontwikkelaar.

Je kunt support verkopen rond een bestaand product, maar je kunt er ook één zelf ontwikkelen. Red Hat doet dit bijvoorbeeld met zijn eigen Red Hat Enterprise Linux. De broncode van RHEL kun je vrijelijk downloaden en verspreiden. Deze programmatuur staat echter niet zozeer centraal. Red Hat verdient zijn geld met de handholding, support en updates. Deze kun je krijgen door een abonnement bij het bedrijf af te sluiten en een deel van de inkomsten hiervan wordt vervolgens weer gestoken in het doorontwikkelen van Red Hats portefeuille aan open source-projecten die zich voornamelijk richten op de enterprisemarkt.

MILJARD DOLLAR

Red Hats aanpak is enorm succesvol. Het haalde als eerste open source-bedrijf een jaaromzet van een miljard dollar. Dit neemt echter niet weg dat je dit businessmodel ook prima kunt gebruiken als klein softwarebedrijfje of freelancer. Het grote voordeel hiervan is dat je zelf de ontwikkeling van het product kunt sturen en gebruik kunt maken van bugfixes en code van mensen van buitenaf. Als je het goed doet, is het mogelijk om een gemeenschap rond jouw project te creëren. Deze enthousiaste gebruikers leveren jou dan weer klanten op die aankloppen voor support.

Wat je je wel moet realiseren, is dat het vrij veel tijd kan kosten om zo'n open source-project te ontwikkelen. Als je slim bent,

zoek je klanten die dit mee kunnen financieren. De core developers achter het Erlang webframework Zotonic doen dit bijvoorbeeld handig. Zij maken als verschillende kleine bedrijfjes gebruik van het pakket en breiden het uit aan de hand van de wensen van hun klanten. De code die dit oplevert, stroomt regelmatig weer terug in Zotonic. Je kunt er echter niet van uitgaan dat je meteen al klanten bij je project zult hebben en niet alle open source-projecten zijn levensvatbaar. Dit kan dus betekenen dat je weekenden en avonden in een project moet steken, zonder direct uitzicht op geld. Je kunt er dan maar het best een echte passie voor hebben, want anders kun je je beter richten op bestaande systemen.

Als je toch wel gehecht bent aan het traditionele beeld van de softwareverkoper, dan zijn er ook mogelijkheden om direct geld te verdienen aan je programmatuur. We komen regelmatig thema's en plug-ins voor WordPress tegen waarvoor moet worden betaald. Dit lijkt in eerste instantie wat merkwaardig, want de licentie van WordPress vereist dat de daaraan gekoppelde software ook open source is. Dit neemt echter niet weg dat je in navolging van Stallman met GNU Emacs geld kunt vragen. Je biedt je extensie dan aan via een webshop, waar de klant kan afrekenen met bijvoorbeeld PayPal en er zelf mee aan de slag kan. Dat laatste is natuurlijk een vrees van menig traditionele softwareverkoper. Ze zijn bang dat anderen met hun pakket aan de haal gaan en er over hun rug geld mee verdienen. In de praktijk blijkt dit niet zo vaak te gebeuren, want het is vrij veel werk om zo'n pakket te voorzien van het nodige onderhoud. De extensies en thema's zijn vaak ook dusdanig klein dat het voor andere programmeurs al gauw makkelijker is om hun eigen systeem te programmeren. Deze methode van programmatuur verkopen kom je veel tegen bij onder andere WordPress,

Magento, OpenCart en Joomla!

DUAL LICENSING

Je kunt je software ook onder twee verschillende licenties aanbieden. Deze techniek wordt dual licensing genoemd en houdt in dat je enerzijds je pakket beschikbaar maakt onder een vrije licentie, maar anderzijds ook een commerciële, gesloten licentie hanteert. Je zorgt er dan voor dat de open source-zijde gebruikmaakt van copyleft. Dit betekent dat als mensen wijzigingen aanbrengen in de broncode en het pakket verder distribueren, deze wijzigingen ook weer vrijelijk beschikbaar moeten komen onder dezelfde voorwaarden. Bedrijven die hier dan niet veel voor voelen, kunnen ervoor kiezen de gesloten licentie aan te schaffen en zodoende hun wijzigingen voor zichzelf te houden. Voor MySQL en Qt wordt zo'n soort systeem gehanteerd en financieel gezien is dit interessant.

Een nadeel van dual licensing is wel dat je er waarschijnlijk de wrok van menig vrijesoftware-zeloot mee op de hals haalt. Zij zien het verkopen van vrije software in een gesloten pakket als een misdaad en zodra je deze route inslaat, is het handig om te zijn voorbereid op een mailbox vol kritische berichten. Als je daartegen kunt en niet zo'n ideologische aanhanger bent van softwarevrijheid, maar het voornamelijk bekijkt vanuit het praktische nut, dan is dit wel een levensvatbaar model om je bedrijf op te baseren.

Er kleeft echter ook voor de pragmatischer medemens een belangrijk nadeel aan dual licensing en dat is het feit dat je alle auteursrechten of toestemming van de programmeurs moet hebben om de software te verkopen onder een andere licentie. Dit zorgt ervoor dat je niet kunt profiteren van bugfixes en code uit de gemeenschap zonder haar ook te vragen om jou het auteursrecht toe te kennen. Ook minder diehard open source-liefhebbers beschouwen dit als

onkies. Dus moet je er rekening mee houden dat je voor de ontwikkeling van het project op jezelf bent aangewezen. Dit model is dan ook niet voor een freelancer weggelegd.

OPEN CORE

Een bepaald smaakje van dual licensing dat de afgelopen jaren sterk is gegroeid, wordt ook wel 'open core' genoemd. Dit betekent dat je niet alleen diverse licenties hanteert, maar ook nog bepaalde features ontwikkelt die specifiek zijn bedoeld voor de betaalvariant van je pakket. Onder andere Zafra, SugarCRM en Puppet Labs gebruiken dit. Net als bij het verkopen van gesloten software kan dit financieel gunstig uitpakken, maar naast de algemene kritiek op dual licensing worden deze pakketten vaak als crippleware bestempeld. Deze term stamt nog uit de DOS-tijd, toen je goedkoop floppy's kon krijgen met sharewareprogramma's of spelletjes die maar een klein deel van de functionaliteit hadden. Net genoeg om je naar de gehele software te doen verlangen. Het is de vraag of zo'n stempel altijd terecht is, maar wie voor deze route kiest moet daar wel mee kunnen leven.

Er zijn nog meer – creatievere – manieren om geld te verdienen met vrije software. Je kunt donaties vragen voor je project, maar je kunt ook geld proberen te verdienen aan de wens tot gemak van je gebruikers. Het downloaden van een pakketje met broncode, dit compileren en vervolgens aan de gang krijgen kost tijd en met name als het een groot pakket is dat weer afhankelijk is van veel andere pakketten, dan kan dit snel een dagdeel in beslag nemen. Daarom kun je net als Paul Davis van het digitale audiomontage-systeem Ardour de voorgedownloadde pakketjes gaan verkopen. Hier vraag je dan een vriendelijk bedrag voor. Davis vraagt kopers zelf een bedrag in te vullen vanaf één dollar. Ook is het mogelijk een financieel evenzeer

vriendelijk abonnement af te sluiten. In het geval van Ardour werkt dit vrij aardig. Het project zit in een kleine niche, maar Davis kan er toch van leven. En in tegenstelling tot wat je zou verwachten, kiest niet iedereen ervoor om het minimumbedrag te betalen. Deze aanpak leverde Ardour veel sympathie op bij de gebruikers en het heeft nu 276 abonneementhouders, die goed zijn voor een bedrag van 1777 dollar per maand.

CROWDFUNDING

Je kunt ook geld ophalen via een crowdfunding-actie. Dit is erg handig als je een startkapitaal nodig hebt en niet naar de bank wilt. MediaGoblin haalde hiermee 42 duizend dollar op. De hoofdontwikkelaar van dit mediapubliceerplatform kan er zo zonder financiële zorgen een jaar aan werken. Dit geeft hem de tijd om een businessmodel voor de tijd daarna uit te vogelen. Hij zou hierna de route van WordPress kunnen kiezen en net als Wordpress.com een gehoste variant van het pakket kunnen aanbieden.

Zoals je ziet, zijn er veel verschillende manieren om zaken te doen met vrije software. Of je nu als freelancer support wilt leveren voor externe systemen, met een gewaagd project geld van de eindgebruiker vraagt of je eigen bedrijf wilt oprichten met een eigen product, de gebruiker staat binnen vrije software altijd centraal en zal uiteindelijk jouw succes bepalen. De meest geslaagde open source-projecten vormen een eenheid met die gebruikers. Zij leveren bugfixes, vertalingen, suggesties voor verbeteringen en wellicht zelfs hele stukken code. Misschien willen ze het ook zakelijk gebruiken en kun je zo je krachten bundelen. In plaats van een eenzijdig verkoopkanaal ontstaat er een levendige gemeenschap van mensen die een project niet alleen downloaden en installeren, maar het ook verbeteren en er zelf deel van uitmaken. [↗](#)