



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

De Open Generatiekloof

Auteurs

Emiel Brok
LPI Nederland en AT Computing
Email: emiel.brok@lpi-nederland.nl

Maurice Verheesen
AT Computing
Email: maurice@atcomputing.nl

Samenvatting

De Open Generatiekloof is het verschil dat bestaat tussen de hoeveelheid open source software professionals die gevraagd worden op de arbeidsmarkt ten opzichte van de hoeveelheid open source software professionals die van de school afkomen. Dit gat blijft bestaan, omdat scholen geen onderwijs geven in open source software, want zij zien geen vraag. En op hun beurt vragen bedrijven niet naar open source software professionals omdat, scholen er geen aanbieden. In dit artikel wordt het probleem geanalyseerd met behulp van het adoptie-framework van Everett Rogers (Rogers, 1984). Deze methode leidt tot een aantal aanbevelingen die men kan uitproberen om de uit de vicieuze cirkel te komen.

Trefwoorden

Open Generatiekloof, open source, open standaarden, Linux, arbeidsmarkt, curriculum, vicieuze cirkel, certificering, kwaliteitsborging, Blauwdruk, Edsger Dijkstra, Lady Ada, besturingssysteem, adoptie framework van Rogers, relative advantage, compatibiliteit, enabling factor, trialability, communicatie, awareness.

De Open Generatiekloof

1 Inleiding

In dit artikel wordt het concept “open generatiekloof” geïntroduceerd. Uit het onderwijsveld komen praktische aanwijzingen die het bestaan van deze kloof onderbouwen. Het verschijnsel lijkt hoofdzakelijk te maken te hebben met de adoptie van open source software in het onderwijs. Daarom wordt in dit artikel via het adoptie framework van Rogers gekeken naar een betere verklaring voor het verschijnsel en mogelijke handvaten voor een oplossing.

In volgende paragraaf wordt de geschiedenis van het IT-onderwijs gerecapituleerd. Daarna volgt een sectie waarin het concept van de “open generatiekloof” nader wordt toegelicht. Vervolgens wordt ‘het adoptie framework van Rogers’ gebruikt om het verschijnsel beter te begrijpen. Hieruit volgen dan mogelijke oplossingsstrategieën. Deze worden samengevat in de sectie aanbevelingen. Daarna volgt nog een sectie over de limitaties waar we mee te maken hebben.

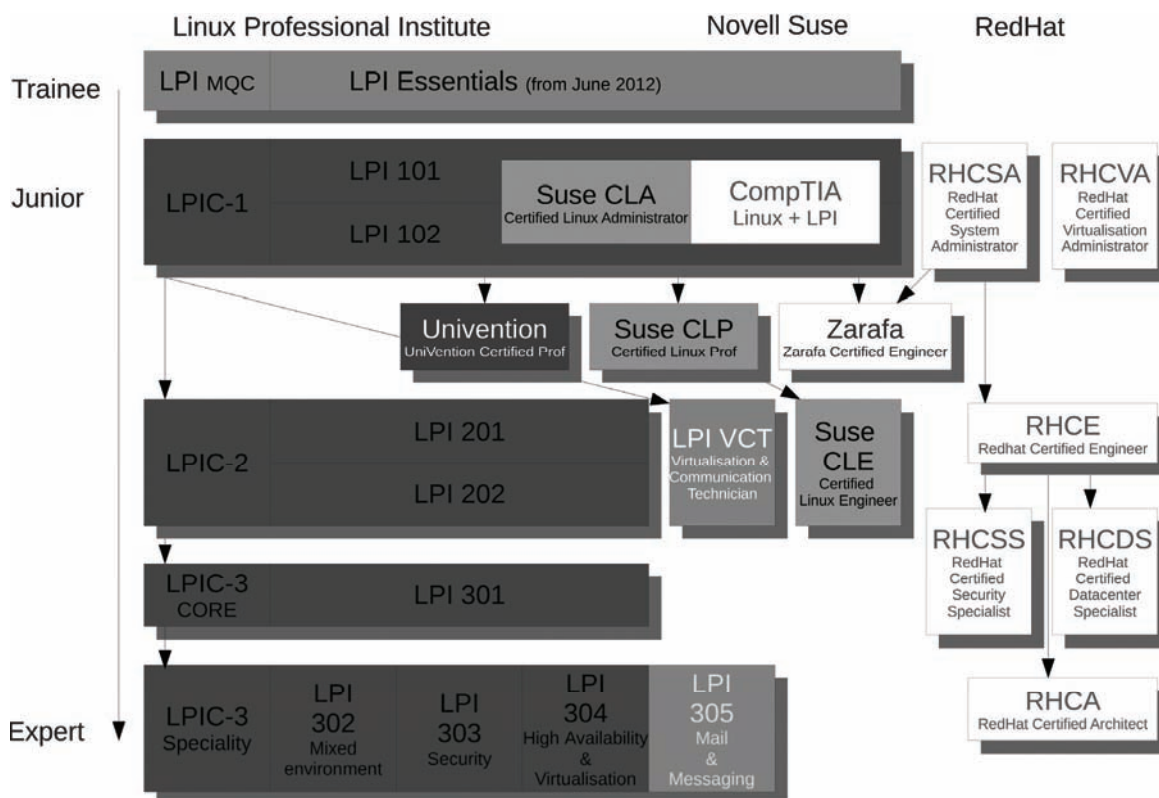
2 Geschiedenis

Een bekende quote van Edsger Dijkstra “een computer heeft evenveel met computerwetenschap te maken als een telescoop met astronomie” (wikipedia 1). Het eerste “computerprogramma” stamt uit 1852 en was geschreven door Lady Ada (wikipedia 2). Het laat zien dat computerwetenschap veel meer te maken heeft met wiskunde dan met iets anders.

Met de komst van de microprocessor, werd het mogelijk om de computation theorie ook in de praktijk te toetsen. In de jaren ‘60 en ‘70 werden daar veel experimenten mee gedaan (denk o.a. aan Xerox Parc en Doug Engelbart). Het doel van de computer was om te helpen bij het rekenwerk en om experimenten te doen met de principes op computerwetenschappelijk gebied. De programma’s werden net als wetenschappelijke artikelen verdeelt onder de wetenschappers. De broncode was dus vrij en het delen van die kennis was vanzelfsprekend. Het delen van de source code werd namelijk als hetzelfde gezien als het delen van mathematische formules. In wiskunde lag immers de basis van de computerwetenschap. De “IT-lessen” waren dus vooral gericht op het gebied van de wiskunde en niet specifiek op computergebied.

Toen in de jaren ‘80 er verschillende besturingssystemen opkwamen en de computer ook binnen het bereik van consumenten kwam, ontstond het idee om via property rights software te veranderen in een product (Wikipedia 3) door artificiële schaarste te creëren. De source code werd niet meer gedeeld en werd beschouwd als eigendom die een bedrijf “in bezit had”, de werking werd beschouwd als bedrijfsgeheim. Dit leidde ertoe dat er allerlei eilandjes ontstonden, combinaties van besturingssysteem en hardware samen met andere programma’s. In feite deden al die eilandjes hetzelfde.

Rond die tijd begonnen scholen het nut in te zien van de computer. Mede omdat het bedrijfsleven de computer op grote schaal ging toepassen. De scholen wilden de computer niet alleen zelf gaan gebruiken, maar ook kinderen opleiden in het gebruik ervan. De scholen maakten hierbij een argeloze keuze in welk besturingssysteem en software geleerd werd. Men realiseerde zich de consequenties niet van deze keuze. Leidend voor hun keuze was dat de hard- en software vooral goedkoop was en ook thuis in gebruik was. Die thuis computer werd dus het ‘de facto’ platvorm welke in de lessen gebruikt werd.



Het conceptuele idee van computers en programmeren verdween hierdoor gaande weg naar de achtergrond. Daarvoor kwam in de plaats, het kunnen omgaan met specifieke applicaties. Menigeen kan zich de Wordperfect lessen nog herinneren.

In de jaren '90 ging dit verder en werd er steeds minder gekeken naar algemene vaardigheden (zoals het "omgaan met een tekstverwerker"), maar naar specifieke ervaring in het werken met Word en/of Wordperfect. Deze vorm van reclame kwam de bedrijven achter deze technologie natuurlijk goed uit.

In de jaren na 2000 begon een kentering. Men begon meer en meer open source software te gebruiken. Linux distributies werden "volwassen" en ook steeds meer toegepast in het bedrijfsleven. Hierdoor realiseerde men zich dat er meer platformen bestaan. Vele bedrijven gebruiken Linux distributies dezer dagen op hun servers. En start-ups gebruiken open source software om innovatieve producten en diensten te realiseren. Het is bijvoorbeeld ondenkbaar dat Google, Facebook en Twitter hadden kunnen bestaan zonder open source software. Door dit toenemende gebruik stijgt ook de vraag naar professionals. Echter er zijn sterke aanwijzingen dat onderwijs in open source software achterblijft, dus ook de aanwas van professionals. Zo is open source software vaak niet opgenomen in het onderwijscurriculum. Indien de school "gratis" licenties kan krijgen worden de leerlingen vaak alleen opgeleid in deze specifieke applicaties.

3 Probleemstelling

Er is een verschil tussen wat de markt vraagt en wat de onderwijsinstellingen te bieden hebben. Scholen nemen open source software niet op in hun curriculum, omdat ze nog te weinig perspectief zien in het vinden van goede stageplaatsen en werkplekken voor junior open source software-professionals.

Veel werkgevers geven aan de overstap naar open source software niet te willen maken omdat zij een tekort constateren aan open source software-professionals (vooral junior professionals).

Hierdoor ontstaat er dus een patstelling. Geen van beide partijen profiteert op dit moment van wat open source software te bieden heeft. Deze vicieuze cirkel noemen wij de open generatiekloof. Een van de gevolgen van de kloof is, dat er een verschil ontstaat tussen de sectoren en regio's waar open source software wel geadopteerd wordt en sectoren en regio's waar dit door de patstelling niet gebeurt.

Recent onderzoek van The Linux Foundation laat zien dat circa 80 procent van de ondervraagde ICT-ers de komende vijf jaar (meer) gebruik wil gaan maken van het GNU/Linux besturingssysteem (Linux Foundation, 2010). Een onderzoek van Accenture (Accenture, 2010) naar de marktontwikkeling rondom open source software laat deze stijging in de vraag ook zien. Daarnaast beginnen bedrijven als IBM, Logica, Ordina en CapGemini zwaar in te zetten open source software. Als de vraag toeneemt, zullen er wél ICT-ers moeten zijn die om kunnen gaan met open source software. Op dit moment is er maar een klein deel dat voldoende kennis heeft. Vaak is deze kennis zelf geleerd in de vrije tijd. Het lijkt erop dat een hele generatie IT-professionals de boot aan het missen is. Het breder opleiden van IT-professionals zal er dus voor zorgen dat ze breder inzetbaar zijn. Hierdoor krijgen ze een betere kans op de arbeidsmarkt van de (nabije) toekomst. Dit artikel gaat in op de vraag hoe deze vicieuze cirkel doorbroken kan worden, zodat men IT-professionals breder kan opleiden.

4 Analyse

Men kan van twee kanten naar de "open generatiekloof" kijken. Waarom er geen adoptie plaatsvindt door de bedrijven en waarom er geen adoptie plaatsvindt in het onderwijs. In dit artikel wordt gekeken naar wat de onderwijssector kan doen om de situatie te verbeteren. Omdat het probleem vooral te maken heeft met adoptie van open source software-opleidingen (zoals Linux-trainingen) wordt er in dit artikel gebruik gemaakt van het adoptie-framework van Rogers (Rogers, 1984). Dit framework is speciaal bedacht voor de adoptie van baanbrekende technologie. Een voorbeeld van zo'n baanbrekende technologie is open source software. Met het adoptie-framework is onderzocht welke factoren het meest belangrijk zijn in de adoptie van een technologie. Er blijken zes factoren van belang te zijn. In onderstaande tabel worden deze factoren beschreven.

Relatief voordeel	Hoe waardevol is de nieuwe technologie? Vaak is bij complexe technologie het nut niet onmiddellijk duidelijk.
Compatibiliteit	Past de nieuwe technologie in de reeds bestaande manieren van werken en cultuur?
Complexiteit	Als het nieuwe systeem moeilijk te gebruiken is, dan heeft het ook minder waarde.
Mogelijkheid om te testen	In hoeverre kan men het nieuwe systeem uitproberen voordat men het aanschaft?
Observatiemogelijkheid	Zijn de voordelen van de nieuwe technologie duidelijk? En zijn deze voordelen ook duidelijk bij andere gebruikers die de technologie reeds toepassen?
Communicatie	Hoe gemakkelijk zijn de voordelen van de techniek over te brengen en te marketen?

In de volgende paragrafen worden de zes adoptiefactoren toegepast op het concept van de "open generatiekloof". Hierbij bestaan de paragrafen telkens uit twee delen. Ten eerste een analyse en ten tweede een aanbeveling voor een oplossing.

4.1 RELATIEF VOORDEEL (RELATIVE ADVANTAGE)

Een van de voordelen van open source software is dat er geen licentiekosten betaald hoeven te worden. Afhankelijk van het (gewenste) interne kennisniveau van een organisatie worden er wel diensten afgenomen rondom open source software. Daarnaast hebben verbeteringen aan open source software een wereldwijd effect. Dit helpt dus direct ook ontwikkelingslanden.

Het is lastig duidelijk te maken dat verandering plaatsvindt op de arbeidsmarkt. Dat maakt het dus ook lastig aan te tonen dat er open source software onderwijs zou moeten plaatsvinden. Feit is echter dat open source software in rap tempo terrein wint. Het is dus een groot voordeel voor studenten om hier de nodige kennis van op te doen. Een IT'er die naast kennis van legacy systemen ook verstand heeft van open source software is breder inzetbaar en dus ook waardevoller voor het bedrijfsleven. Daarnaast is te zien dat veel bekende succesvolle start-ups uitvoerig gebruik maken van open source software. Het opleiden van studenten met open source software kennis geeft hen dus ook de kans om zelf de volgende succesvolle start-up te worden.

4.2 COMPATIBILITEIT (COMPATIBILITY)

Hier zijn o.a. twee situaties te noemen die een probleem vormen. Ten eerste dient het niveau van de open source software lessen te passen bij het denkniveau van de doelgroep. Ten tweede vormen de bestaande legacy IT-systemen (en manier van werken) een belemmering voor de introductie van nieuwe systemen. Deze twee situaties worden hier behandeld.

Op dit moment sluit het niveau van de open source software lessen nog niet aan bij de doelgroep. Zo heeft bijvoorbeeld een deel van het LPI-lesmateriaal een hoog theoretisch gehalte en is er weinig tot geen aandacht voor praktijk of competentie gericht onderwijs. De leerniveaus van LPI sluiten echter wel aan op de Europese normen. Ook wordt er gekeken of, en hoe, LPI past in de MBO-leerdoelen. Daarnaast is LPI volop bezig met de ontwikkeling van een nieuwe certificering die beter aansluit op het Nederlandse MBO3- en MBO4-niveau. Deze certificering zal er speciaal op gericht zijn om aan te sluiten op het kennis en kunde-niveau van deze doelgroep. Daarnaast ligt de focus niet op specifieke Linux distributie kennis, maar juist ook op algemene computer-kennis. Ten tweede is er een trend in onderwijsland van grootschalig uitbesteden van ICT. Hierdoor verliest men de controle over de IT-infrastructuur. De keuze van toegepaste programma's en producten wordt gemaakt door de partij waaraan de infrastructuur is uitbesteed. Studenten in aanraking laten komen met alternatieven is er dan niet meer bij. Deze manier van denken is niet van deze tijd. Kijk maar naar de top 5 internationals die state-of-the-art technologie gebruiken: Google, Facebook, Twitter, RedHat en Apple. Zij voeren precies het tegenovergestelde beleid. Zij zitten boven op de technologie, terwijl ze omzet realiseren met ongerelateerde zaken. Technologie vormt bij hen de enabling factor waardoor men geld kan verdienen. Wil men goed ICT-onderwijs geven zal men dus af moeten stappen van oude denkpatronen, zoals het uitbesteden van IT en het vast blijven houden aan het concept software als product. Keuzevrijheid en dus flexibiliteit is veel belangrijker en daarbij is er behoefte aan dienstverlening die waarde toevoegt rond/aan de gebruikte software.

4.3 COMPLEXITEIT (COMPLEXITY)

Er zijn drie elementen die open source software complex doen lijken in het veld. Ten eerste de leercurve die als steil wordt ervaren. Ten tweede de "absorptive capacity" van MBO-docenten en als laatste de angst voor het verlies van relevantie van bestaande kennis.

Linux heeft een steile leercurve. Dat heeft er enerzijds mee te maken dat men Linux pas op latere leeftijd gebruikt omdat men is “opgegroeid” met andere systemen. Hierdoor heeft open source software dus een soort “achterstand” in de algemene computerkennis die mensen de facto beheersen. Anderzijds heeft het ermee te maken dat de kracht van bijvoorbeeld Linux vooral ligt in de command-line. En die gebruikers-interface kent nu eenmaal een steilere leercurve dan grafische interfaces.

Doordat studenten bij Linux veel meer de concepten achter de techniek te zien en te leren krijgen, is men beter in staat problemen creatief op te lossen. Het voordeel van meer kennis op conceptueel niveau is dat complexe problemen op andere platformen efficiënter op te lossen zijn. De techniek achter de schermen is namelijk hetzelfde, alleen de presentatie is anders. Door het conceptueel te snappen, is er makkelijker van de ene interface en platform over te stappen naar een ander. Hierdoor heeft men meer oplossingen ter beschikking ten opzichte van de vaak beperkte set mogelijkheden die geboden worden door de grafische interfaces.

Naast de steile leercurve, is er ook een probleem met de zogenaamde absorptie capaciteit (Cohen and Levinthal, 1990). Dit is de mate waarin men in staat is (nieuwe) kennis op te doen. Het kennisniveau van de MBO-docenten zal omhoog moeten om lessen in open source software mogelijk te maken. Dit heeft vooral te maken met onbekendheid en niet zozeer met mogelijkheid om het te kunnen begrijpen. Wat hierin meespeelt is de taalbarrière. Veel termen en software zijn alleen beschikbaar in het Engels. Dat maakt het complex om te leren voor een gemiddelde MBO'er.

Oplossing is investeren van de private sector en de overheid in Nederlandstalig lesmateriaal en examens. Daarnaast stimuleert LPI haar partners om systeembeheerders opgeleid te krijgen in Linux, met als kroon op het werk de LPI-certificering. Hierdoor stijgt het kennisniveau in het algemeen en verandert de opstelling van “onbekend is onbemind” naar gezonde nieuwsgierigheid.

Een derde barrière vormt de angst voor het verlies aan waarde. Sommige docenten die specialist zijn in andere gebieden dan open source software voelen zich bedreigd door de kennis die studenten nu al hebben van open source software. De kennis van de studenten is vaak omvangrijker dan die van de docenten zelf. Hierdoor schieten deze docenten in de verdediging en blijven vooral krampachtig vasthouden aan hun huidige kennis. Bang dat de huidige kennis haar waarde verliest. Dat is soms terecht soms onterecht. Hier ligt een taak om de angsten weg te halen door te laten zien dat juist een opleiding in open source software en Linux distributies meerwaarde biedt op de arbeidsmarkt, juist ook voor docenten zelf.

4.4 MOGELIJKHEID OM TE TESTEN (TRIALABILITY)

Open source software is bijzonder simpel uit te proberen. Men kan het immers gewoon downloaden en gebruiken, zonder dat men zich druk hoeft te maken over andere zaken. Er zijn echter drie redenen waarom het in de onderwijspraktijk toch moeilijk blijkt te zijn om open source software uit te proberen. Ten eerste het heersende IT-beleid in de organisatie, ten tweede de bestaande IT-architectuur en ten derde het kostenplaatje. Op sommige scholen is een strikt uitbestedingsbeleid actief. Hierbij is de complete ICT-voorziening uitbesteed aan een derde partij en is de school de controle volledig kwijt. Dat laatste is vaak een bewuste keuze, met als argumentatie dat een school vooral geen IT-bedrijf moet willen zijn. Dit maakt het invoeren van alternatieven uitermate lastig, aangezien er dan met een externe partij onderhandelt moet over de mogelijkheid deze alternatieven te installeren en te beheren.

Ten tweede kan de bestaande legacy IT-infrastructuur een rem op adoptie vormen. Er is bijvoorbeeld “gestandaardiseerd” op platformen die niet onder controle zijn van de school. Daarnaast zijn er systeembeheerders die geen alternatieven willen uitproberen. Dit is te begrijpen, aangezien zij vooral betaald worden om de bestaande IT werkend te houden. Veranderingen zijn dan dus per definitie ongewenst.

Als laatste spelen de kosten mee. Vaak is er geen of weinig verschil in kosten tussen proprietary software en open source software. Het komt omdat scholen de proprietary software gratis of ver onder de kostprijs van de leverancier krijgen. De gangbare licentiekosten gebaseerd op marktwaarde komen daarmee te vervallen. Door deze marktinterventie wordt er vaak niet verder gekeken of deze keuze verstandig is en of er wellicht alternatieven zijn.

Deze situaties belemmeren dus het opleiden van breed geschoolde IT-ers want het maakt het uitproberen van alternatieven iets dat “not done” is in plaats van juist een mooie ervaring. Een oplossing voor het verhogen van trialability kan door het gebruik van een USB-stick, CD-rom of (open source) cloud oplossing. Daarmee kan men om de hele “gestandaardiseerde bedrijfsvoering” heen.

4.5 ZICHTBAARHEID (OBSERVABILITY)

Hierbij kunnen er twee groepen onderscheiden worden. De zichtbaarheid van open source software in het algemeen en de zichtbaarheid van lesmateriaal.

Het bedrijfsmodel van open source software is anders. Bedrijven kunnen geen opbrengsten halen uit licenties, dus komen de opbrengsten vaak uit diensten rondom de software, zoals opleiding en advies. De proprietary partijen daarentegen kunnen lesmateriaal gratis of onder de kostprijs leveren vanwege hun inkomsten uit de licentie verkoop. Dat kunnen open source software dienstverleners helaas niet, omdat de verkoop van goed lesmateriaal juist voor de opbrengsten moet zorgen. Dat betekent dat in de praktijk het lesmateriaal van de proprietary leverancier meer zichtbaarheid heeft omdat het vaak gratis verspreid wordt.

Daarnaast is veel open source software niet zichtbaar. Dat komt vooral omdat de Linux desktop heel weinig gebruikt wordt. Het aantal installaties van Linux op de server is vele mate groter. Voorbeeld is dat 80% van de web servers op Linux draait denk aan Google, Facebook en twitter. Dus voor studenten is het belangrijk dat ze hier wat van leren. Ook het succes van Apache, OTRS en Nagios is onderbelicht. Dit zijn allemaal belangrijke populaire open source projecten. Ook zijn er maar een paar scholen die daadwerkelijk iets doen met open source software in hun curriculum. Een belangrijk deel van de zichtbaarheid van software en platformen komt voort uit de ervaring die men ermee opdoet tijdens de schoolcarrière.

Het ontbreekt dus aan duidelijke business-cases wat de adoptie moeilijk maakt. Een oplossing zou kunnen liggen in concurrentie. Sommige scholen concurreren met elkaar om studenten. Het laten zien dat men voorop loopt door Linux onderwijs te geven zal differentiërend werken op het vlak om zich als innovatieve school te presenteren.

Daarnaast helpt LPI Nederland de zichtbaarheid te verbeteren. Dit door het showcasen van goede voorbeelden. LPI Nederland vertelt vaak over de succesverhalen. Hierbij worden ook het kleine aantal scholen waar er al les gegeven wordt met Linux of waar Linux gebruikt wordt als voorbeeld genoemd (o.a. Hogeschool Zuyd, ROC Nijmegen en ROCvA).

4.6 COMMUNICATIE (COMMUNICATION)

Het blijkt erg lastig om helder de voordelen van open source software uit te leggen. Vaak is er sprake van onbekend is onbemind. Maar open source software is nogal paradigma-doorbrekend. Zo'n 20 jaar bestaat er het concept dat software een product is. Maar het open source software paradigma gaat weer terug naar de basis en beschouwt een computerprogramma niet als een product, maar als kennis. Dit verandert nogal wat aspecten ten aanzien van het omgaan met software. Vaak escaleert een gesprek over het uitleggen van de principes achter open source software in een soort ideologische discussie. Dat laat zien hoe lastig het is om de voordelen van open source software helder te maken. Oplossing kan zijn het beter communiceren van de voordelen; “what’s in it for me”.

Ook het feit dat scholen op dit moment denken aan te sluiten op de huidige markt, maakt het moeilijk om hen te overtuigen van de toekomstige situatie waarin het open source software gebruik dominant is. Voorbeeld van hoe snel de markt situatie kan veranderen is de bijzonder snelle adoptie van Android, wat gewoon Linux gebruikt. Voorbeelden van organisaties die de markt voorzien van objectieve informatie zijn LPI Nederland, ECABO en Stichting Praktijkleren. Dit doen ze onder andere door het opleiden van docenten, ontwikkelen en leveren van lesmateriaal en andere informatie en het gezamenlijk optrekken in een lobby.

Aanbevelingen

Zoals bij iedere patstelling, zal één van de twee posities (het bedrijfsleven of de onderwijsinstellingen) een andere stelling moeten innemen. In dit artikel hebben we de arbitaire keuze gemaakt om naar onderwijsinstellingen te kijken en wat zij zouden kunnen doen. Om op de genoemde adoptiefactoren in te spelen volgen hieronder de aanbevelingen op een rijtje:

- Stap af van focus de korte termijn strategie. Op korte termijn kan het aantrekkelijker lijken om alleen te onderwijzen in een beperkte set aan legacy producten. Op de lange termijn is dat zeer zeker niet het geval. De toekomst is de capaciteit te hebben om IT naar je hand zetten en daar zijn brede IT-vaardigheden voor nodig. Het doel moet dus zijn IT-kennis en geen product-kennis.
- Stap af van focus op je eigen microcultuur. De wereld is een dorp. Investeren in lokale open source software ontwikkeling helpt mee aan de innovatie en globalisering van de wereldwijde economie.
- Stap af van oude denkpatronen, zoals het uitbesteden van IT en het vast blijven houden aan het concept “software als product”. Leveranciers die waarde toevoegen aan de software of toegang geven tot IT diensten hebben de toekomst.
- Gebruik USB-stick, CD-rom of cloudoplossingen om Linux distributies te onderwijzen indien er geen ander besturingssysteem op de IT-omgeving van de school mag worden geïnstalleerd. Daarmee ga je om de gehele “gestandaardiseerde bedrijfsvoering” heen.
- Laat zien dat men voorop loopt door open source software onderwijs te geven en zich zodoende als innovatieve school te presenteren.
- Vermijd een ideologische discussie door het beter communiceren van de voordelen, “what’s in it for me”. Iedereen wil immer vrij zijn, dus software vrijheid is logisch.

5 **Limitations**

Het zou de schrijvers erg geholpen hebben als er statistieken werden bijgehouden over de omvang van het open source software onderwijs in Nederland. Verder leunt dit onderzoek erg op verhalen uit het onderwijsveld. Er is nog bijzonder weinig wetenschappelijk onderzoek gedaan naar de individuele en maatschappelijke effecten van commercieel gesponsord onderwijs op IT-gebied. Helaas is dit artikel daarom gebaseerd op een beperkte set bronnen. Hierbij hebben we stellingen niet kunnen controleren via meerdere bronnen. Dat maakt dat er onder andere geen causale verbanden te trekken zijn. Vandaar dat dit artikel geen conclusie trekt, maar eindigt met aanbevelingen. Het doel is de lezer aan het denken te zetten over het nut en noodzaak van breder opgeleide IT-professionals.

6 **Wie is wie**

LPI Nederland richt zich op het promoten van leveranciersafhankelijke certificering van Linux-kennis en -vaardigheden. Zij stimuleert het gebruik van LPI-certificaten en informeert bedrijven, organisaties en HRM-medewerkers over de meerwaarde van ICT-ers die LPI gecertificeerd zijn.

ECABO is het kenniscentrum beroepsonderwijs bedrijfsleven voor de economisch/administratieve, ICT- en veiligheidsberoepen. Ongeveer twee miljoen Nederlanders werken in deze beroepen. Tien procent van de jongeren is er voor in opleiding. ECABO is een van de zeventien kenniscentra die ons land telt. Samen vormen zij de schakel tussen beroepsonderwijs en bedrijfsleven. Vanuit een gezamenlijk doel: Een arbeidsmarkt in balans voor iedereen.

Stichting Praktijkleren (SPL) ondersteunt ROC's bij binnen- en buitenschools praktijkleren. SPL ontwikkelt in samenwerking met de ROC's examenproducten, praktijkwijzers voor de beroepspraktijkvorming (bpv), bedrijfssimulaties en projecten in de vorm van beroepsprojecten voor de opleidingsgebieden sector Economie. Daarnaast heeft Stichting Praktijkleren een adviserende functie; zo adviseert SPL scholen over leerplannen of bij het opzetten van een onderwijsleerbedrijf. Verder verzorgt SPL trainingen voor docenten en assessoren.

7 Over de auteurs

Emiel Brok is mede-oprichter en bestuurslid van LPI Nederland, de zelfstandige Nederlandse afdeling van het wereldwijde LPI, het Linux Professional Institute. Hij zet zich in voor professionalisering van de open source software markt en dan vooral voor open source software onderwijs en certificering binnen de publieke onderwijssector. Emiel Brok werkt bij AT Computing, een opleider, consultancy en remote beheer organisatie gespecialiseerd in Linux en UNIX.

Maurice Verheesen is collega van Emiel bij AT Computing. Maurice is BSc. in Electrical engineering and MSc. in Innovation Management. Zijn master thesis gaat over het onderwerp hoe organisaties om gaan met open source software bij hun inkoop proces. Voor deze proceeding heeft hij Emiel geholpen het concept van de open generatiekloof te verklaren aan de hand van het adoptie framework.

Literatuur

- Accenture (2010) "Open Source Research Industry Findings"
<http://newsroom.accenture.com/images/20020/IndustryFindings.pdf>
- Cohen and Levinthal (1990), "Absorptive capacity: A new perspective on learning and innovation", *Administrative Science Quarterly*, Volume 35, Issue 1 pg. 128-152.
- Dedrick, J., West, J. (2004). An exploratory study into open source platform adoption. *IEEE Transactions on Software Engineering*, 8, 1530-1605. (Los Alamitos, CA, USA)
- Ecabo (2011) "Rapportage enquête Medewerker beheer ICT"
- Gartner (2010) "Open Source Predictions For 2010"
http://blogs.gartner.com/mark_driver/2009/12/08/open-source-predictions-for-2010/
- Glynn, E., Fitzgerald, B., Exton, C. (2005). Commercial adoption of open source software: an empirical study. *IEEE transactions on software engineering*, 1-10.
- Linux Foundation (2010) "Linux Adoption Trends: A Survey of Enterprise End Users"
<http://www.linuxfoundation.org/publications/linux-adoption-trends-end-user-survey>
- Nagy, D., Yassin, A., Bhattacharjee, A. (2010). Organizational adoption of open source software: barriers and remedies – *Communications of the ACM*, 53 (3), 148–151.
- Rogers, E.M. (1984) "Diffusion of innovations"
- Verheesen, M.P.M.W. (2010) How to buy something that is free? Master thesis Technische Universiteit Eindhoven alexandria.tue.nl/extra1/afstversl/tm/Verheesen%202010.pdf
- Wikipedia: http://en.wikipedia.org/wiki/Edsger_W._Dijkstra
- Wikipedia: http://en.wikipedia.org/wiki/Ada_Lovelace
- Wikipedia: http://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists