

Berlin Open 2009: Wissen, Vernetzung, Innovation.

Offen gesprochen - Konzepte der Offenheit in Praxis und Wissenschaft, Unternehmen und Verwaltungen

Open Source im public sector: open innovations am Beispiel der GPLI-Suite

Jürgen Renfer, München

1. Einleitung

Lassen sich offene Konzepte mit ökonomischen Theorien vereinbaren? Diese zentrale Frage stellt sich in einem Zeitalter revolutionärer Technologiefortschritte, fortschreitender Globalisierung und fundamentaler Weltfinanzkrise in elementarer Weise. Oder sind umgekehrt vielleicht sogar gerade offene Konzepte einer der Schlüssel zu einer Wirtschafts- und Finanzordnung neuer Qualität?

Offene Konzepte finden sich seit jeher in vielen Lebensbereichen. Man denke beispielsweise an Musik: „Spielanleitungen“, bspw. in Form von Partituren, sind offen gelegt und können für vergleichsweise kleines Geld erworben werden. Dennoch bleibt das Werk geschützt, und auch monetäre Aspekte werden berücksichtigt, so z.B. durch Abgaben für öffentliche Aufführungen. Der Künstler hat an der Verbreitung seines Werkes Interesse. Nur so kommt er zu Bekanntheit, Ruhm und vielleicht auch Reichtum. In der Wissenschaft ist ähnliches zu beobachten: nur durch Veröffentlichung kann wissenschaftlicher Diskurs gelingen, wenn gleich damit im Gegensatz zum Beispiel der Musik noch kein unmittelbar handelbares Produkt oder gar eine Ware entsteht. Der Wissenschaftler hat also ebenfalls Interesse an der Verbreitung seines Werks.

Die Motivation zur Veröffentlichung liegt bei den Beispielen in ihrer Verbreitung. Dem gegenüber entstand mit wachsender Industrialisierung ein stetiger Trend zu geschlossenen Systemen. Dieser beruhte zunächst auf dem Schutzbedürfnis, industrielle Fertigungsverfahren insbesondere dann absichern, wenn zur Entwicklung dieser Verfahren erhebliche Ressourcen eingesetzt werden müssen. Im Ergebnis ist festzuhalten, dass offene Konzepte dort ihre Grenzen fanden und geschlossen wurden, wo hohe Anteile des Produktwerts im Produktionsverfahren liegen. Besonders ausgeprägt ist dieser Effekt bei Software zu beobachten, deren Wert nahezu ausschließlich im Produktionsverfahren – also der Entwicklung – liegt, während die Distributionskosten gegen Null tendieren.

2. Die PC-Historie als Geschichte offener Systeme?

Umso überraschender muss es auf den ersten Blick erscheinen, dass neben diesem sog. „closed source“-Ansatz als Gegenpol der sog. „open source“-Ansatz zu beobachten ist. Historisch betrachtet war Software zunächst immer „offen“: geliefert wurden zum Beginn der Datenverarbeitungsära programmierbare Rechenmaschinen bzw. Computer, deren Herstellungsaufwand sehr groß und Preis entsprechend hoch war. Diese wurden entweder im Paket mit Programmen geliefert oder die Software wurde eigens für den vorgesehenen Einsatzzweck erstellt. Der Wert lag damals in der Hardware, Software wurde eher als Betriebsmittel wie Kraftstoff für ein Fahrzeug gesehen. Damit bestand auch kein Grund, Software besonders zu schützen.

Mit weiterer Digitalisierung des kommerziellen und später auch des privaten Lebens änderte sich diese Sichtweise bald: der Computer wurde zum preiswerten Massenprodukt und damit stieg der Absatzmarkt für Software ebenso drastisch an. Gleichzeitig entstand die Möglichkeit, statt individuelle Programme der Anfangszeit immer mehr Standardsoftware zu erstellen: Tabellenkalkulationen, Textverarbeitung, Grafik usw. Deren Wert lag überwiegend in der Entwicklung, denn nun mussten die Anforderungen

einer Vielzahl – potentieller – Kunden erhoben und in der Entwicklung berücksichtigt werden. Dem gegenüber standen vergleichsweise sehr geringe Distributionskosten. Gleichzeitig eröffneten sinkende Hardwarekosten, das dadurch entstehende Delta zumindest teilweise abzuschöpfen: Software wuchs in der Wertschätzung und damit auch an Wert. Also galt es diesen Wert zu schützen: offene Ansätze wurden „geschlossen“.

Mangels Kenntnis der „Spielanleitungen“ waren jetzt auch Kenntnisse über Schnittstellen zum Austausch der Daten zwischen Software-Systemen verborgen. Die Zeit der proprietären Systeme begann. Daraus resultierte ein wirksames Mittel zur Kundenbindung, was die Entwicklung geschlossener Systeme weiter forcierte. Die Entscheidung für ein Softwaresystem konnte einer Entscheidung „fürs Leben“ gleichkommen.

Keine Entwicklung ohne Gegenentwicklung: von starken Abhängigkeiten gezeichnet, wurde der Ruf zur Interoperabilität zusehends lauter. Zunächst auf Schnittstellen fokussiert entstanden Bestrebungen, die stark geschlossenen, proprietären Systeme wieder zu öffnen. Besonders die von mehreren konkurrierenden Herstellern beherrschte Unix-Landschaft erkannte die Gefahren proprietärer Systeme und gab einschlägigen Initiativen sprechende Namen wie z.B. X/Open. Dieses Hersteller- und Anwenderkonsortium verfolgte seit Mitte der 1980er-Jahre das Ziel, gemeinsame Schnittstellen zu schaffen und Anwendungssoftware zu vereinheitlichen, somit Unix zu einem offenen System zu entwickeln – welches es in frühen Jahren bereits war. Ähnliche Ziele verfolgte das Ende der 1980er-Jahre gegründete Herstellerkonsortium Open Software Foundation (OSF), so dass es nicht verwundert, dass sich beide Gruppierungen Mitte der 1990er-Jahre zur Open Group zusammenschlossen.

Zwischenzeitlich wanderte der Fokus des Marktes von Unix-basierten Systemen zu Windows-basierten Systemen der Fa. Microsoft, der eine umfassende Marktbeherrschung erst im Bereich der Endgeräte und dann im Serverbereich gelang. Damit wanderte der Fokus gleichzeitig aus einem oligopolen zu einem monopolen Markt, der wiederum von geschlossenen Systemen gekennzeichnet war.

Interessanterweise basierte diese Entwicklung aber auf der Tatsache, dass erst durch ein offenes Konzept die nötige Installationsbasis für die geschilderte Marktbeherrschung weltweit geschaffen wurde. Das wachsende Feld der damals so bezeichneten Mikrocomputer – die zunächst überwiegend für private Anwendungen gedacht waren – wollte auch die Fa. IBM nicht tatenlos der Konkurrenz wie z.B. Apple überlassen. Daher kam neben den kommerziellen IBM-Linien wie z.B. Systeme /360 (Großrechner) und /23 sowie später /36 (mittlere Datentechnik) Anfang der 1980er-Jahre der IBM PC – korrekt als IBM 5150 bezeichnet – für den Endanwender auf den Markt. Das System wurde „mit heißer Nadel“ in sehr kurzer Entwicklungszeit und unter Einsatz einfachster Komponenten entwickelt. Ein folgerichtiger Schritt, denn der erste Apple stand im Ruf, in einer Garage entwickelt worden zu sein. Er war entsprechend preisgünstig in der Fertigung wie im Verkauf, also musste IBM vom Entwicklungsaufwand bisheriger, kommerzieller Systeme grundlegend Abstand nehmen, um das Feld der Microcomputer konkurrenzfähig bedienen zu können. Der IBM PC war von drei entscheidenden, offenen Konzepten geprägt:

- A. Sein einfaches Konzept unter Rückgriff auf marktgängige Komponenten wie Computerchips (z.B. iAPX86-Prozessoren der Fa. Intel) und Peripherie konnte problemlos nachgebaut werden, was bald zu Nachbauten aus Fernost führte, sog. PC-Klone. Hierzu trug auch der Umstand bei, dass das Systemkonzept im Grunde nicht rechtlich geschützt war, also offen war. Die Ursache dafür lag in der Einschätzung des zuständigen IBM-Managements, dass der Microcomputermarkt und damit auch der IBM 5150 keine nennenswerte Bedeutung am Markt erlangen würde, weshalb er lizenzfrei nachgebaut werden konnte.
- B. Ähnlich wie Apple war – und ist – sein Konzept einfach erweiterbar, weil Steckkarten jederzeit nachträglich eingebaut werden können. Die Spezifikation dieser Schnittstelle,

dem sog. ISA-Bus, lag offen, da die mitgelieferten Systemhandbücher ausreichende Dokumentationen enthielt.

C. Die notwendige Betriebssoftware – damals DOS 1.0 – war zwar nicht selbst offen, aber immerhin frei am Markt verfügbar – von einer gewissen Firma Microsoft...

Aus dieser Kombination offener Konzepte entwickelte sich mit der Verbreitung des IBM PC, dessen Nachfolgesysteme wie bspw. IBM AT sowie der vielfältigen Klone ein faktischer Industriestandard, der bis heute dominiert.

IBM erkannte diese Entwicklung und versuchte, sich mit PC-Weiterentwicklungen dadurch vom Markt abzuschotten, dass geschlossene Systeme entstanden, so bspw. der sog. Microchannel als Ersatz für das bis dahin offene Steckkarten-Konzept. Diese technologisch ohne Zweifel weit fortgeschrittenen Ansätze konnten sich am Markt nicht – mehr – durchsetzen und scheiterten, so dass IBM die PC-Produktion schließlich aufgab.

Dagegen profitierte die Fa. Microsoft von der Verbreitung der PCs und Kompatibler enorm. Neben anderen Plattformen wurde überwiegend Software diesen IBM-Industriestandard entwickelt, so auch das Windows-System. Dabei konnte die schlichte Masse weltweit zwischenzeitlich im Einsatz befindlicher PCs verbergen, dass es niemals verlässliche und verbindliche Testverfahren für die Kompatibilität von preiswerten Klonen gab. Der Markt regelte die Standardisierung: inkompatible Systeme waren schlicht nicht mehr verkäuflich.

Nach einigen Fehlstarts gelang spätestens mit Windows 3.0 der kometenhafte Aufstieg der Firma aus Redmond, die später ähnliche Erfolge im Serverbereich mit Windows NT feiern sollte. Auch dort hatte sich der IBM-Industriestandard zwischenzeitlich durchgesetzt. Damit konnten die Systeme vom Einzelplatzcomputer bis zum leistungsfähigen Serversystem skalieren, was Microsoft in die Lage versetzte, die Grundlagen seines Windows-System ebenfalls zu vereinheitlichen.

Grundlage der skizzierten Entwicklungen sind die offenen Konzepte des IBM-PC-Industriestandards aus den frühen 1980er-Jahren. Die Offenheit führte zur weltweit beherrschenden Verbreitung möglich, womit die eingangs bemühten Zusammenhänge zwischen Offenheit und Verbreitung auch an der Historie der Computer belegbar erscheinen.

Was bedeutet dies für die Zukunft geschlossener Systeme wie der Windows-Software, die offenkundig ebenfalls von ausserordentlichem Erfolg gekennzeichnet sind? Windows wurde im Serverbereich ursprünglich für mehrere Systemplattformen entwickelt. Es ging aus dem gemeinsamen Entwicklungsprojekt OS/2 der Firmen IBM und Microsoft hervor. Nach einer ersten gemeinsamen Version trennten sich die Entwicklungslinien beider Firmen, wobei IBM die OS/2-Linie und Microsoft die bisher auf Einzelplatzsysteme ausgerichtete Windows-Linie weiterverfolgten. Der erste Windows-Server bot ebenso wie die OS/2-Linie noch Unterstützung für verschiedene Systemplattformen. Neben dem IBM-Industriestandard mit Intel-Prozessoren wurden auch MIPS-, PowerPC- und DEC Alpha-Systeme unterstützt. Die drei letztgenannten konnten gegenüber der beherrschenden Intel-Plattform keine bedeutende Marktposition in der Welt IBM-kompatibler PCs aufbauen. Daran änderte auch die Verfügbarkeit mindestens zweier Serversysteme bedeutender Hersteller, nämlich OS/2 von IBM und Windows NT von Microsoft, nichts. Letztlich konnte es sich Microsoft erlauben, gleichzeitig mit der Zusammenführung dem Windows-System für PCs und dem Windows-System für Server die Unterstützung sämtlicher anderer Plattformen aufzugeben. Hier hat das geschlossene Windows-System ebenfalls von dem offenen System „Windows“ und dessen enormer Verbreitung profitiert.

3. Das Internet – die digitale Revolution auf Basis offener Systeme

Die Erfolgsgeschichte des Internets ist zweifellos mit dem Siegeszug des PCs untrennbar

verbunden. Die Wurzeln des Internets reichen in Form des sog. ArpaNet in die frühen 1960er-Jahre zurück. Ursprünglich als militärische, ausfallsichere Infrastruktur geplant, wurden zunächst Computer einiger US-Universitäten dergestalt miteinander verbunden, dass Datenaustausch und Kommunikation möglich war. Damit bildete das Internet eine erste Infrastruktur-Grundlage für ein umfassendes Computernetz. Für die bequeme Kommunikation fehlten damals noch geeignete Protokolle und Benutzeroberflächen, die langsam in Form des heute fast schon wieder vergessenen Gopher oder UseNet entstanden.

Entscheidenden Einfluss auf den offenkundigen Siegeszug des Internet nahm in den späten 1980er-Jahren die Entwicklung des sog. Hypertext-Systems von Tim Berners-Lee an der europäischen Kernforschungsorganisation Conseil Européen pour la Recherche Nucléaire (CERN) in Genf/Schweiz. Berners-Lee suchte nach einem offenen System, das es erlauben sollte, Forschungsergebnisse effizient unter Fachkollegen auszutauschen. Diese Forschungsergebnisse lagen damals bereits überwiegend in digitaler Form auf Internet-vernetzten Computern vor. Was fehlte war die Verweismöglichkeit aus einer Textstelle auf eine andere, auf einem anderen System liegende Textstelle. Hierzu ersann Berners-Lee Verweise, die als Markierung im Quelltext hinterlegt werden und auf eine Passage des Zieltextes verweisen. Da der Verweis auf ferne Computer erfolgen kann, wurde er als Hyperlink und die entsprechende Auszeichnungssyntax als Hypertext Transfer Language (HTTP) bezeichnet. Das entstehende Geflecht nannte Berners-Lee World Wide Web (WWW).

Sowohl das Protokoll HTTP als auch geeignete Programme dafür (Web-Server sowie Browser) waren frei verfügbar und damit offen. Wenngleich der erste Web-Server von Berners-Lee nicht auf einem IBM-kompatiblen Standardsystem lief, sondern auf einer ebenso technologisch wegweisenden wie kommerziell erfolglosen NeXT-Maschine, zeigt eben dieser Umstand die Stärke offener Systeme: nachdem Berner-Lee's WWW überzeugte, war es ohne lizenzbedingte Innovationshürden frei verfügbar und daher schnell möglich, das offene Protokoll auf Grundlage der vorhandenen Referenzimplementierungen auf andere Systeme zu portieren. Mit der Portierung auf den zwischenzeitlich schon 10 Jahre am Markt etablierten, sehr stark verbreiteten, offenen IBM-kompatiblen Standard-PC, traten zwei offene Systeme in Wechselwirkung: je weiter sich das WWW verbreitete, desto mehr Interesse bestand an dessen Einsatz mit der Folge, dass wiederum mehr Endgeräte und Server benötigt wurden. Interessenten wählten dafür wiederum gerne den verbreiteten Standard, also Standard-PCs.

Nebenbei erwächst aus dem ersten WWW-Server vom Typ NeXT eine andere Verbindung zum IBM PC. Der NeXT wurde vom damaligen Firmeneigner Steve Jobs vorgestellt, der ursprünglich an eben jener Apple-Entwicklung beteiligt war, die IBM zum Markteintritt in die Welt der damals so bezeichneten Microcomputer mit dem IBM PC veranlasste. Zwischenzeitlich ist Jobs zu Apple zurückgekehrt und hat dabei auch seine Firma NeXT eingebracht.

4. Linux – die Verbreitung des offenen Betriebssystems

Die Idee eines offenen Betriebssystems reicht bis zur entsprechenden GNU-Initiative (Akronym „GNU ist Not Unix“) von Richard Stallman in die frühen 1980er-Jahre zurück. Dabei folgte Stallman der eingangs skizzierten Praxis aus den Pionierzeiten der Computerentwicklung, als ein lebhafter Austausch zwischen Entwicklern sowie Anwendern einschließlich der Offenlegung aller Quellcodes üblich war, um diese bei Bedarf einfach anzupassen. Während die Kommerzialisierung der Software-Industrie zunehmend zu geschlossenen, proprietären Systemen führte, entschied sich Stallman für den Erhalt offener Systeme und rief GNU ins Leben.

Es dauerte bis zum Anfang der 1990er-Jahre, als von Linus Torvalds ein erstes offenes, Unix-ähnliches System unter dem namens Linux entstand, welches bald mit den bis dahin ebenfalls offen verfügbaren GNU-Elementen zu dem bekannten Linux und daraus

vielfältig abgeleiteter Distributionen verschmolz.

Damit lässt sich auch an diesem Beispiel der Effekt gegenseitig verstärkender Wechselwirkungen offener Systeme beobachten: vereinfacht dargestellt fehlte es GNU lange Jahre an einem Systemkern, der GNU zur Verbreitung hätte bringen können. Andererseits fehlte es Torvalds schlicht an Ressourcen, seinen Systemkern mit dem nötigen Umfeld zu versehen, den ein komplettes System benötigt. Erst beide Systeme zusammen führten zur Verbreitung, die Linux zwischenzeitlich weltweit erfuhr.

5. Offene Software-Systeme: open source

Bisherige Ausführungen haben an exemplarisch ausgewählten Beispielen gezeigt:

- Die Offenheit von Systemen trägt wesentlich zur deren eigener Verbreitung bei.
- Offene Systeme können alleine aufgrund ihres Verbreitungsgrades im Konkurrenzkampf gegenüber technologisch weiterentwickelten, geschlossenen Systemen bestehen.
- Offene Systeme können in Wechselwirkungen treten, die den Verbreitungsgrad zuerst ermöglichen und/oder potenzieren.
- Verbreitete, etablierte offene Systeme können als „Wirt“ für geschlossene Systeme dienen, was zu ähnlich potenzierenden Wechselwirkungen führen kann.

Lassen sich diese Thesen in zukünftiger Praxis an weiteren Beispielen beweisen, so steht zu erwarten, dass open source ein Erfolgsmodell werden kann. Allerdings stellen sich dazu weitere Fragen:

- Wie können Distributionsmodelle und Geschäftskonzepte gestaltet werden, damit Urheber offener Systeme wirtschaftlichen Nutzen ziehen können um Entwicklungen motiviert weiter voranzutreiben?
- Welche Parameter müssen beachtet werden, damit „open source“-basierte Software-Systeme fachlich überzeugen können?

5.1 Distributions- und Geschäftsmodelle

Bisherige Systementwicklungen auf offener Grundlage zeigen, dass entsprechende Konzepte in solche ohne und solche mit kommerziellem Hintergrund grob gegliedert werden können.

Konzepte ohne kommerziellen Hintergrund (non-profit) finden sich insbesondere im öffentlichen bzw. gemeinnützigen Sektor, somit bei Behörden und Bildungsträgern. Beide finanzieren sich im Regelfall über Steuermittel, so dass Entwicklungen wiederum der Allgemeinheit bereitgestellt werden (Distributionspolitik). Ein typisches Beispiel hierfür sind Entwicklungen des Bundesamts für Sicherheit in der Informationstechnik (BSI), welches Bürgern Sicherheitstools auf Grundlage offener Software zur Verfügung stellt. Als weitere Beispiele können vielfältige Hochschulentwicklungen wie bspw. OTRS (ein Helpdesk-System) dienen, die als „open source“ bereitstehen.

Darüber hinaus bestehen in non-profit Bereichen häufig starke Anreize, gegebene Ziele mit minimalem Mitteleinsatz herbeizuführen (sog. Minimalprinzip). Somit ist es wünschenswert, durch den Einsatz offener Systeme Anschaffungskosten zu sparen. Dieser Ansatz rechnet sich allerdings nur soweit, als Implementierungs- und Betriebskosten den ersparten Investitionsbetrag nicht wieder aufsaugen (Geschäftspolitik). Vielfältige erfolgreiche open source-Strategien haben die Sinnhaftigkeit dieses Ansatzes bereits belegt, so bspw. Einführung von open source-

Systemen in der Bundestagsverwaltung oder im kommunalen Sektor wie bspw. Schwäbisch Hall.

Damit lässt sich subsumieren, dass gerade in non-profit-Bereich fruchtbarer Boden für offene Systeme zu finden ist.

Konzepte mit kommerziellem Hintergrund können in drei Gruppen gegliedert werden: sie können auf Entwicklungsmotivatoren (bspw. positive Einflussnahme auf die Etablierung von Industriestandards durch Verbreitung entsprechender Software) oder auf Unterstützungsleistungen zielen und schließlich ähnlich wie im non-profit Sektor zur Kostensenkung durch Investitionersparnis dienen. Näheres haben entsprechende Branchenverbände erarbeitet, wovon exemplarisch die Open Source Business Foundation (OSBF) als hochspezialisierte Einrichtung zur Abbildung von Geschäftsmodellen mit offenen Konzepten genannt sei.

5.2 Erfolgsparameter „open source“ am Praxisbeispiel

Am Einsatz einer offenen Software-Lösung im öffentlichen Sektor soll abschliessend exemplarisch dargelegt werden, welche fachlichen Parameter die dargelegte, latente Eignung offener Konzepte zur Verbreitung besonders unterstützen.

Dafür findet ein Produkt Verwendung, welches der umfassenden Verwaltung von Routineaufgaben einer IT-Abteilung dient. Infolge stetig steigender, legalmotivierter Anforderungen an einen kompetenten IT-Betrieb besteht in vielen IT-Bereichen Handlungsdruck zur Einführung einer integrierten Software-Lösung. Allerdings kann die IT-Abteilung durch den Einsatz einer solchen Lösung keinen unmittelbaren Mehrwert für die Anwenderseite schaffen, da letztlich lediglich die interne IT-Organisation betroffen ist und daher keine sichtbaren Ergebnisse „nach aussen“ entstehen.

Damit besteht hohe Motivation, ein geeignetes Werkzeug möglichst kostengünstig zu beschaffen und zu betreiben, weshalb offene Konzepte besonders geeignet sein können. Daneben lassen offene Konzepte Anpassungen zu, die gerade IT-Bereiche eigenständig vornehmen können.

Schließlich ist es wünschenswert, dass solch ein Werkzeug die wesentlichen Anforderungen aus einem Guss anbietet, also ganzheitlich arbeitet. Damit liegen hohe quantitative Anforderungen vor, die qualitativ dadurch verstärkt werden, dass möglichst die wesentlichen Standards zum IT-Betrieb abgebildet sein sollen.

An diesen Beispielen wird deutlich, dass hohe Anforderungen an das Produkt ebenso hohe Anforderungen an das Distributions- und Geschäftsmodell zur Folge haben werden. Damit schließt sich der Kreis, weshalb Software in der Historie ab einem gewissen Punkt vorwiegend „geschlossen“ angeboten wurde: kann für das anspruchsvolle Distributions- und Geschäftsmodell keine tragfähige Lösung in offenen Bereich gefunden werden, muss sie im geschlossenen Modell gesucht werden – was sich wiederum über allgemeine, betriebswirtschaftliche Mechanismen erschließt.

Die individuelle Erweiterungsfähigkeit offener Systeme wurde bei GLPI ähnlich dem Vorgehen bei vielen anderen erfolgreichen open source Produkten dadurch geordnet, dass eine sog. Plugin-Schnittstelle bereitsteht. Somit können individuelle Erweiterungen problemlos der Allgemeinheit zur Verfügung gestellt werden. Derzeit sind etwa 60 Plugins verfügbar, was den geschilderten, quantitativ und qualitativ hohen fachlichen Anforderungen entgegenkommt. Durch Arbeitsteilung entstand im Laufe der Jahre um den Produktkern eine Applikationssuite, wofür die Ressourcen der drei federführenden Entwickler alleine vermutlich nicht ausreichend gewesen wären.

Andererseits hält das Plugin-Konzept das Produkt schlank: installiert wird nur, was im konkreten Anwendungsfall benötigt wird. Die wirkt dem Pareto-Effekt effizient entgegen.

Neben dem Plugin-Konzept pflegt GLPI Partnerschaften zu Entwicklergruppen, die an ähnlichen Produkten arbeiten. Abstimmungen dienen dazu, dass Entwicklungsschwerpunkte ergänzend statt konkurrierend fokussiert werden. Eingesetzte Ressourcen ergänzen sich daher, was insbesondere bei dem eng verzahnten Partnerprodukt OCS_NG zur Netzwerkverwaltung beispielhaft umgesetzt ist. Konsequenterweise enthält GLPI keine Plugins zur Netzwerkverwaltung, sondern eine Schnittstelle zu OCS_NG.

Das Produkt Preise bspw. im Rahmen der "Trophées du Libre 2006" gewonnen, wozu sicher auch nützlichen Funktionen wie bspw. die automatische Prüfung auf Versionsupdates einschließlich der installierten Plugins sowie niedriger Einstiegshürden durch Live-Demonstrationen im Internet sowie vergleichsweise einfacher Produktinstallation beitragen konnten.

Der entscheidende Wert des Produktes liegt letztlich wohl in der lebendigen Entwickler- und Anwendergemeinschaft („community“), die Anfragen oftmals unter einem Tag und somit sehr schnell beantwortet. Damit findet die Vision des oben referenzierten Richard Stallman ihre reale und aktuelle Entsprechung.

Bei all den positiven Aspekten stellt sich die Frage, warum das Produkt zum wirtschaftlichen Nutzen der Entwickler nicht auch kommerziell vertrieben wird. Selbst die fehlende lokalisierte Dokumentation sollte dieser Frage nicht entgegenstehen, da das Produkt gemäß Referenzliste absolut überwiegend im französischsprachigen Raum Beachtung findet. Auch einige unerhebliche Programmfehler könnten durch geeignete Preispolitik kompensiert werden.

Dass selbst die Original-Dokumentation noch lückenhaft, nicht immer sehr aktuell und teilweise ebenso redundant wie widersprüchlich vorliegt, könnte bei kommerzieller Nutzung schon eher schädlich wirken. Hier würde auch die schnelle und kompetente „community“ nicht kompensieren, weil schlicht nicht vorhanden: welcher Anwender eines kommerziellen Produktes sähe eine Motivation, Zeit für Fragen anderer Anwender aufzuwenden.

6. Fazit: open innovations bedürfen der Kontrolle

Das Beispiel zeigt, dass offene Systeme nicht nur geeignet sein können, zur Verbreitung beizutragen. Aus der Verbreitung ergeben sich weitere wesentliche Effekte. In soziologischer Sicht bilden sich Gemeinschaften, sog. communities, die im Verständnis eines ausgewogenen „Nehmens und Gebens“ agieren. Dadurch entstehen arbeitsteilige Effekte und die Bereitschaft des Einbringens, die es selbst der kleinen Gruppe von drei Entwicklern erlaubt, mit einer offenen Basis beherrschbaren Umfangs die Grundlage für ein umfassendes offenes System zu legen.

Das Beispiel zeigt weiter, dass entscheidender Impuls für den Erfolg offener Systeme der oben ausgeführte Verteilungsgrad ist. GLPI wäre für englische oder deutsche Anwender

ebenso nützlich wie im weit verbreiteten französischen Sprachraum. Mangels Übersetzung konnte dieser Impuls jedoch hier – noch – nicht wirken. Dem gegenüber funktionieren die beschriebenen Wechselwirkungen zwischen offenen Systemen im sprachlich begrenzten Distributionsraum sehr gut, was die Vielzahl der Plugins sowie der Partnerschaften zeigt.

Diese Effekte hat Henry Chesbrough von der Berkeley University of California als Open Innovation beschrieben: Nutzung der Außenwelt zur Steigerung des Innovationspotentials (vgl. Chesbrough, H. W. (2003): Open Innovation. The New Imperative for Creating and Profiting from Technology, Boston: Harvard Business School Press).

Den Ansatz haben Grassmann und Enkel weiter differenziert und kommen zum Ergebnis, dass steigender Wettbewerbsdruck infolge Globalisierung die Optimierung der Innovationsprozesse erfordert, was insbesondere durch deren Öffnung erfolgen kann. (vgl. Gassmann, O./Enkel, E. (2006): Open Innovation. Die Öffnung des Innovationsprozesses erhöht das Innovationspotential, in: zfo, 3/2006 (75. Jg.), S. 132-138). Dies erfolgt durch Integration externen Wissens (sog. outside-in-Process), Externalisierung internen Wissens (sog. inside-out-Process) oder einer Mischform aus beiden (sog. coupled Process).

Was bei offenen Systemen zumindest an den vorgestellten Beispielen offensichtlich gut funktioniert und durchaus sogar mit closed innovations (vgl. Schumpeter, J.A. (1950): Kapitalismus, Sozialismus und Demokratie, zweite, erweiterte Auflage, Bern: A. Francke AG Verlag) in Symbiose vereint werden kann, birgt andererseits Gefahren. Sobald der Innovationsprozess nicht mehr hinreichend gesteuert wird, entstehen komplexe Systeme, die ausser Kontrolle geraten können. Dieses Risiko wohnt open innovations systemimmanent inne und gilt heute als einer der Mitauslöser der Finanzkrise (vgl. D. Fastnacht (2009), Offene Geschäftsmodelle als Mitauslöser der Finanzkrise, Chancen und Risiken von Open Innovation in der Finanzwirtschaft, in Neue Züricher Zeitung, Ausgabe vom 09.01.2009, online: http://www.nzz.ch/nachrichten/wirtschaft/aktuell/offene_geschaeftsmodelle_als_mitausloeser_der_globalen_finanzkrise_1.1668074.html).

Im überschaubaren Umfang der exemplarisch dargestellten GLPI-Suite als offenes System erfolgt die Kontrolle des offenen Systems durch die Entwickler, die bspw. die Vielzahl der Plugins verwalten. Im grossen Maßstab der Weltwirtschaft liegen Indizien vor, dass die Staatengemeinschaft zur maßvollen Kontrolle eines offenen Finanzsystems gefordert sein könnte. Unter dieser Prämisse könnte erwartet werden, dass sich offene Systeme zum Nutzen Aller ebenso positiv auf die Innovationsfähigkeit auswirken wie die dargestellten Beispiele technologischer Revolution.