Anarchie!?

von Stephan Richter

Das politische System der Free-Software-Gemeinden

Free Software-Entwickler sind nicht nur eine Gruppe von Leuten, die Computer-Code schreiben. In den letzten fünf bis sieben Jahren wurden Gemeinden gegründet und ein politisches System, welches ich Sozial-Demokratische Anarchie nenne, entwickelte sich. Dieses System beinhaltet Elemente von Demokratie, Anarchie und Sozialismus – alles Konzepte, welche oft unabhängig voneinander benutzt werden.

Disclaimer

Obwohl einige von Ihnen wahrscheinlich die Idee von einem politischen System in Free Software-Gemeinden komplett ablehnen, habe ich jedoch die Erfahrung gemacht, dass es zu zumindest eine soziale Hierarchie gibt, die sich in den letzten Jahren zu einem proto-politischen System entwickelte. Daher glaube ich nicht, dass es Zufall ist, dass wir Guido van Rossum – Erfinder von Python – unseren „Benevolent Dictator For Life" (Gütiger Diktator auf Lebenszeit) oder Jim Fulton den „Pope of Zope" (Zopes Papst) nennen, da er, Jim Fulton, sozusagen das Gehirn und der Mann mit dem Vetorecht für Zope 2 war, bis der Entwicklungsprozess Mitte 2001 offen gelegt wurde. Denken Sie immer noch, diese Namen wurden zufällig gewählt? Ich glaube nicht; ich bin davon überzeugt, dass solche Namen Führern verschiedener Gemeinden gegeben werden, da eine politische Struktur der „reellen Welt" – bewusst oder unbewusst – von den Gemeindemitgliedern akzeptiert wird. Natürlich gibt es auch Gemeinden, in denen jedes Kernmitglied ein gleiches Stimmrecht hat.

Ich weiß, dass nicht alle Gemeinden gleich sind. Ich selbst bin sehr stark in der Zope (www.zope.org/)-Gemeinde seit dreieinhalb Jahren auf allen Ebenen eingebunden, angefangen vom Beginner, der viele zuvor beantwortete Fragen stellte, bis zum Kernentwickler des kommenden Zope 3-Release. Dieser Artikel basiert auf den Erfahrungen, die ich in der Zope-Gemeinde machte, und kann daher stark von Berichten anderer Gemeinden abweichen. Des Weiteren reflektiert dieser Artikel meine Voreingenommenheit für Free Software-Entwicklungsmethoden, welche sehr unterschiedlich von anderen sein können, selbst in der Zope-Gemeinde. Einige Leute behaupten, dass mein Ansatz nicht sehr demokratisch und oft autoritär wirkt, aber bitte bilden Sie sich Ihre eigene Meinung. Letztendlich sehe ich meinen Ansatz zu Free Software als sehr idealistisch und verteidige oft Richard Stallmans Meinungen; ich bin sehr stolz darauf, dass ich in den letzten drei Jahren nur an Free Software gearbeitet habe.

Gemeindebeteiligung

Die meisten Leute betreten eine Free Software-Gemeinde als Anfänger, oft durch eine oder mehrere Mailinglisten-Abonnements, auf denen sie Fragen zur betreffenden Software stellen. Jedoch gibt es auch Interessierte, die nur die Listen lesen, aber nicht aktiv an den Diskusionen teilnehmen und daher als passive Gemeindemitglieder gelten. Über die Jahre habe ich oft viele solche Leute auf Konferenzen und Messen kennen gelernt. Wenn Neulinge sich mit der Gemeinde und der Software angefreundet haben, dann befördern sich diese Mitglieder oft selbst – ja, in Free Software-Gemeinden wird die soziale Ebene selbst durch Taten definiert – und werden somit zu Mentoren, die Fragen von Einsteigern und anderen Gemeindemitgliedern beantworten.

Obwohl viele Mitglieder im Supportlevel bleiben, gibt es immer wieder auch solche, die an der Softwareentwicklung selbst interessiert sind. Da neue Entwickler normalerweise die inneren Funktionen der Software noch nicht gut genug kennen, beteiligen sie sich am Anfang meist an Diskussionen. Diese Entwicklerdiskussionen, oft auf Mailinglisten und in Newsgroups ausgeführt, debattieren neue Feature-Ideen, Design- und Ausführungsdetails oder administrative Entscheidungen. Aufgrund ihrer offenen Natur (die ganze

Gemeinde kann diese Diskussionen einsehen und verfolgen) sind sie oft der Mittelpunkt des demokratischen Prozesses und erinnern mich an die griechische Demokratie. Sobald eine Entscheidung oder Übereinstimmung erreicht ist, ernennen sich Mitglieder selbst, um die vorgeschlagenen Ideen umzusetzen. Darunter fallen die Softwareentwicklung, das Schreiben von Dokumentation, das Entwerfen von Grafiken oder das Ausführen jeglicher administrativer Aufgaben. Ich nenne diese Leute von jetzt an die „Handelnden". Jedoch ist es offensichtlich, dass dieser Prozess des Entscheidens und die Ausführung nicht immer glatt verlaufen können.

Motivation

Okay, Leute entwickeln Free Software, aber warum? Obwohl diese Frage auf den ersten Blick offensichtlich erscheint, denke ich, dass die Antwort ziemlich komplex ist. Als Free Software erstmals als ernsthafte Alternative auftrat, entwickelten die meisten Entwickler sie hauptsächlich als Hobby und um einen Kick daraus zu bekommen. Andere wollten einfach nicht für Software gutes Geld bezahlen. Punkt. Durch die Kommerzialisierung von Free Software in den letzten Jahren verschob sich die Motivation für viele Entwickler.

Viele Geschäftsleute haben mittlerweile das Potenzial von Free Software erkannt und versuchen nun, in solche Projekte via vieler verschiedener Businessmodelle einzusteigen, sodass sie einen Teil vom Kuchen abhaben können. Die akademischen Gemeinden, speziell Universitäten, sind höchstwahrscheinlich die größte Teilgruppe. Ihre Ideologien überschneiden sich sehr gut mit denen der Free Software-Bewegung. Daher finden wir oft Universitäten unter den großen Sponsoren und Studenten stellen wahrscheinlich den größten Teil der Entwickler.

Jedoch gibt es auch große Beiträge von Firmen. Aber was erhoffen sie sich davon? Erstens glauben viele Firmen, dass durch die Freistellung ihrer Software die Akzeptanz des Produktes vorangetrieben wird und dass sie einen Marktvorteil haben, da sie die originalen Entwickler des Systems sind. Außerdem haben diese Firmen auch oft das beste Know-how über das Produkt. Ein anderer Grund, die Sourcen freizulegen, ist die Hoffnung, dass sich eine Gemeinde bildet, welche das Produkt in Schuss hält (zum Beispiel Bugs korrigiert) und es vielleicht sogar durch neue Features erweitert. In diesem Falle wechseln Firmen oft von produktbasierenden zu serviceorientierten Businessmodellen, wie es zum Beispiel Zope Corporation (früher Digital Creations) getan hat. Diese Variante funktioniert meistens sehr gut, da die Profitmargin für Consulting höher ist als

Von Handelnden und Rednern

bei Produkten. Letztlich ist es oft auch viel billiger und sicherer, in ein Free Software-Projekt zu investieren, da die Produkte nicht vom Wohl und Wehe der Firma abhängig sind und allemal eine bessere Lösung als selbstgebastelte Software bieten.

Kommunikation

Da Free Software-Gemeinden nicht in ein räumliches Gebiet eingegrenzt werden können, sind traditionelle Kommunikationskanäle wie Gemeindeversammlungen im Rat- oder Clubhaus nicht möglich. Neue Methoden mussten entwickelt werden, um das Kommunizieren über den ganzen Globus und Zeitzonen möglich zu machen. Existierende Technologien wie eMail, Newsgroups und FTP boten einen guten Einstieg und wurden bald durch Chat-Services wie zum Beispiel IRC und Websites bereichert. Einige dieser Technologien wurden sogar für naturwissenschaftliche Gemeinden, die Vorfahren der Free Software-Gemeinden, erfunden. Jedoch ist nicht ein einziges Kommunikationsmittel für alle Kommunikationsebenen geeignet.

Vor ein paar Monaten schlug mir Lalo Martins (Brasilien), einer der Zope 3-Entwickler, einen Mechanismus vor, den ich von allen mir bisher bekannten Methoden am besten finde. Als Erstes sortierte Lalo alle Kommunikationsmittel nach der Persistenz (Lebensdauer) ihrer Informationen. Er identifizierte Chat-Services, speziell IRC, als schnelllebigste Quelle, da Diskussionen mit der Schließung des Clients verloren gehen (abgesehen von Logs). Daher sind Chat-Services ideal für Brainstorming, sodass „schlechte" Ideen nicht einmal als Möglichkeiten dokumentiert werden können. Diese Diskussionen werden oft auch nur von ein paar interessierten Entwicklern gehalten, die sich in dem zu diskutierenden Gebiet mehr oder weniger auskennen. Die Ergebnisse der Chat-Sessions werden dann in einer Mail kompiliert und zu einer Entwickler-Mailingliste zur weiteren Diskussion gesandt. Jetzt haben alle anderen Entwickler Zeit, sich die neuen Vorschläge anzuschauen und ihre Kommentare und Ideen beizusteuern. Wenn einmal diese Vordiskussionen abgeschlossen sind, wird ein formales Proposal geschrieben und auf eine Website für letzte Kommentare gestellt. Die Zope-Gemeinde nutzt eine Implementation von Wikis (c2.com/cgi/wiki?WikiWikiWeb) für diesen Zweck. An diesem Punkt ist das Proposal für die ganze Gemeinde zur Einsicht freigegeben und oft werden auch nur kleine Veränderungen vorgenommen.

Jedoch reichen manchmal die eben beschriebenen Kommunikationsmittel nicht aus. Persönlichere Kommunikation ist nötig. In diesen Fällen können Entwickler Telefongespräche und Konferenzschaltungen bevorzugen oder sogar ein Entwicklertreffen auf Konferenzen und anderen Events organisieren. Diese Treffen waren speziell für die großen Projekte wie KDE wichtig, welche jetzt regelmäßig Entwicklertreffen halten, um Entscheidungen über zukünftige Entwicklungen zu machen. Als weiteres Beispiel übernahm die Zope-Gemeinde in den letzten Monaten eine Methode von eXtreme Programming, die „Sprints". Während eines Sprints trifft sich eine Gruppe von Entwicklern für einige Tage, um in Paaren an Teilen der Software (in diesem Falle Zope 3) zu arbeiten. Die Sessions werden oft von einem erfahrenen Entwickler, dem Mentor, geleitet.

Natürlich ist der hier beschriebene Prozess nicht perfekt. Manchmal werden aus Diskussionen Flame Wars über oft nebensächliche Probleme. In diesen Situationen ist es dann oft wichtig, zwischen den „Handelnden" – Mitglieder, die regelmäßig zu der Entwicklung beisteuern –

und „Rednern" – diejenigen, die nur in (meist unkonstruktiven) Diskussionen teilnehmen – zu unterscheiden.

Handelnde versus Redner

Da keine Gemeinde perfekt ist, gibt es immer Leute, die viel zu sagen haben, aber nie etwas tun. Sie mögen nie ein Proposal und schicken Links zu riesigen Artikeln anderer Technologien, die man lesen sollte, selbst wenn diese völlig unzutreffend sind. Sie wissen immer noch nicht, welchen Typ ich meine? Tim Peters, einer der fünf Python Labs-Entwickler, schrieb mir einst: „Du musst einen kennen, um zu verstehen." Diese Gruppe nenne ich die „Redner", da diese nicht nur unproduktive Beiträge liefern, sondern auch meistens die Diskussionen unnötig verlängern und dadurch die Entwickler zurückhalten, was einem oft den letzten Rest an Motivation raubt.

Die „Handelnden" auf der anderen Seite versuchen, Aufgaben so schnell wie möglich zu erledigen. Sie geben konstruktive Kritik und helfen mit der Verwirklichung der Proposals durch gute Ratschläge. Gewöhnlich sind Macher auf Basis einer persönlichen Agenda tätig, die nicht unbedingt den Interessen der anderen Entwickler gleichen muss. Jedoch kann allgemeiner, gemeindebasierender, ökonomischer oder politischer Druck auch den Handelnden dazu bewegen, nicht direkt eigene Interessen durchzusetzen, da die allgemeine Akzeptanz der Gemeinde wichtiger ist.

Das Wichtigste zu wissen ist jedoch, dass *nur* Handelnde eigentliche Entwicklungsarbeit leisten, und sie treffen daher die ultimativen Entscheidungen, was sie in die Lage von Führern mit Einfluss versetzt. Allein aus diesem Grund musste sich eine Form von Politik in den Free Software-Gemeinden herauskristallisieren. Es können drei Aspekte der Politik von Free Software-Gemeinden identifiziert werden: Anarchie, Demokratie und Sozialismus. Aus diesem Grunde nenne ich dieses politische System eine Sozial-Demokratische Anarchie.

Anarchie-Aspekt

Die Handelnden regieren! Und da sich jeder qua eigener Autorität selbst auf diese Ebene begeben kann, ist es möglich, dass jeder auch regieren kann. Martijn Faassen, ein Python- und Zopeveteran aus den Niederlanden, sagte einst scherzhaft: „Mein Ziel ist Weltdominierung!" Dies war eine Referenz zu seinem Traum, dass jedes Gemeindemitglied sein Produkt akzeptieren und nutzen würde, ohne dass dies notwendig Core-Komponenten wären (sein Traum hat sich in Zope 3 erfüllt, da sein Produkt in den Kern aufgenommen wurde).

Anarchie wird durch das Nichtvorhandensein einer strukturierten regierenden Institution definiert, welche das egozentrische Regieren verschiedener Individuen zur Folge hätte. Da eine Free Software-Gemeinde hauptsächlich aus Freiwilligen besteht, gibt es keine formalen Kräfte, Mitglieder zu organisieren, und der einfache autoritäre Ansatz funktioniert aus offensichtlichen Gründen nicht.

Mein Ziel ist Weltdominierung!

Anarchie wird auch dadurch ausgedrückt, dass Individuen willkürlich mit der Software machen können, was sie wollen.

Ohne Entscheidungen geht es jedoch nicht. Daher ist es oft üblich, dass die Entwickler eine Art von Aristokratie gründen, einen „Vertrauensrat". Wenn ich zum Beispiel nicht an einem Aspekt der Software beteiligt bin, vertraue ich oft den anderen Entwicklern, dass sie eine gute Lösung finden, solange jemand anderes das Proposal überprüft.

Demokratische Aspekte

Jedes Gemeindemitglied hat das *Recht* zu wählen. Ob eine Person sich dafür entscheidet, von dem Recht Gebrauch zu machen, ist eine andere Frage. Obwohl jede Stimme in allgemeinen Entscheidungen gleichwertig ist, passiert es manchmal, dass Entwicklerstimmen mehr zählen, wenn es um Entwicklungsprobleme und Vorschläge geht, was meiner Meinung nach sehr wichtig ist, da Entwickler sich in diesen Fällen gewöhnlich besser auskennen.

Es ist jetzt offensichtlich, dass ich nicht von einer Demokratie in Aristoteles' Sinne spreche, sondern mehr vom Gemeinwesen (Polis), einer Regierungsform, die Aristoteles als das rechtmäßige Regieren der Vielen beschrieb. Ganz im Gegenteil, Aristoteles war sehr gegen das Regieren des Mobs (Pöbel) und er bevorzugte gebildete und informierte Entscheidungsträger, ganz wie wir es heute in Free Software-Gemeinden vorfinden.

Sozialistische Aspekte

Wie in sozialistischen Systemen wird auch hier angenommen, dass alle Beteiligten für das Wohlergehen der Gemeinde arbeiten, da das Gedeihen der Community langfristig jedem Teilnehmer zu Gute kommt. Dieses Verhalten nennt man „Ethischer Egoismus". Aber im Vergleich zu dem ultra-liberalen Standpunkt wird nicht die Annahme getroffen, dass alle Individuen von Natur aus „gut" (der Gemeinde wohlgesonnen) sind. Das System schützt sich selbst durch ständiges gegenseitiges Kontrollieren der Teilnehmer gegen böswillige Angriffe, was es unmöglich für einzelne Personen macht, ohne Kontrolle Entscheidungen zu treffen oder Artefakte zu bearbeiten.

Der zweite Teil des sozialistischen Aspektes ist die Frage der Vermögensverteilung. Dieses Problem ist meiner Meinung nach sehr gut gelöst, da Diskussionen und Entscheidungen für jeden in der Free Software-Gemeinde und darüber hinaus zur gleichen Zeit und am gleichen Ort zugänglich sind. Das bedeutet, dass niemand einen Wettbewerbsvorteil durch das Verschweigen oder Zurückhalten von Informationen erreichen kann. Der einzige Vorteil, den jemand haben kann, entsteht in Bezug auf Fähigkeiten und Know-how. Ich denke, dass Free Software-Gemeinden jeden bisher bekannten Versuch, sozialistische Systeme zu erstellen, an Erfolg übertroffen haben, da wir bisher noch nie solch eine Gleichheit in der Vermögensverteilung erlebt haben.

Defizite des Systems

Die EuroZope-Gemeinde gründete 2000 einen in Deutschland sitzenden gemeinnützigen Verein, sodass wir in der Lage waren, Spenden von Regierungsorganisa-

tionen zu empfangen und Firmen ihre Spenden von der Steuer absetzen können. Im Gründungsprozess war es notwendig, einen Vorstand zu wählen. Die Gemeinde erwartete nun automatisch Führung von dem Vorstand, was ihr Gleichgewicht zerstörte. Jetzt bin ich mir nicht mehr so sicher, ob das Formalisieren der EuroZope-Gemeinde eine so gute Idee war. Wir haben ein paar wichtige Lektionen gelernt: Nicht alle Vorstandsmitglieder sollten aus einem Land (sehr wichtig in unserem Fall) sein und jemand muss sicherstellen, dass alle Stimmen angehört werden. Also, wenn es keinen dringenden Grund für eine formale Institution gibt, dann sollte man keine gründen. Es wird sehr schmerzlich sein!

Wir haben nun gesehen, wie das politische System der Free Software-Gemeinden funktioniert. Was haben wir dabei gelernt?

Neue Gemeindemitglieder/-teilnehmer

„Neulinge" sollten so schnell wie möglich nicht nur durch Fragenstellen, sondern auch – wenn möglich – durch Beantworten der Fragen anderer in das Gemeindeleben einbezogen werden. Mitglieder sollten auch an Diskussionen teilnehmen, wobei die Kommentare konstruktiv sein sollten. Andere Formen konstruktiver Beiträge beinhalten wesentliche und konkrete Vorschläge sowie das Erstellen von wertvoller Dokumentation über eine Diskussion, die später genutzt werden kann.

Des Weiteren haben neue Leute einen weiteren Vorteil aufgrund des mangelnden Komforts der Software. Während Langzeitmitglieder sich an oft ernst zu nehmende Nutzbarkeitsfehler gewöhnen, merken neue Mitglieder diese Mängel sofort. In diesen Fällen sollten neue Mitglieder spezifische Vorschläge zur Verbesserung machen.

Firmen, die der Free Software-Bewegung beitreten

Wenn eine Firma ihre Software freigibt, dann sollte diese nicht den Fehler machen und Software einfach veröffentlichen und dann hoffen, dass einige Free Software-Entwickler vorbeikommen, um die Software weiterzuentwickeln. Es ist jedoch auch nicht genug, einfach eine Mailingliste oder einen IRC-Channel aufzusetzen. Die Firma sollte die interessierten Entwickler in die richtige Richtung führen; vielleicht sogar die ersten Beiträge überwachen und zensieren, bis genügend Gemeindemitglieder ausreichend Erfahrung haben.

Ein weiterer Aspekt der Fürsorge und Ernährung der Free Software-Gemeinde ist der Prozess der Artefaktenaktualisierung. Die Gemeinde wird nie alles tun, was man selbst will. Während Entwickler Beiträge liefern, um die Software zu erweitern oder Bugs zu beheben, interessieren sie sich oft nicht für die veraltete Dokumentation.

Das war auch ein Problem in der Jugendzeit von Zope. Wenn es besser gelöst worden wäre, würden wir sicherlich jetzt ungefähr dreimal so viele Nutzer haben. Aber seitdem hat Zope Corporation viel gelernt. Dokumentation wird nun auf allen Ebenen gefördert und die Kernentwickler des neuen Zope 3-Frameworks sind sogar dazu verpflichtet, ihren Code zu dokumentieren, bevor sie ihn einchecken.

Entwickler

Entwickler sollten niemals ihre Zeit mit anstrengenden, unmotivierenden Diskussionen verbringen! Wenn Sie als Entwickler merken, dass die Diskussion in eine Richtung führt, die Sie nicht interessiert, ignorieren Sie diese einfach. Sie sollten Kritik an Ihrer Arbeit ernst nehmen und von ihr lernen, aber nehmen Sie die Kritik nie so persönlich, dass Sie aufhören zu entwickeln. Entwickler gehören zu den wichtigsten Mitgliedern der Gemeinde, daher braucht die Gemeinde Sie; manchmal erhalten Sie viel Respekt (wenn ein großartiges Feature entwickelt oder ein Show Stopper-Bug aufgelöst wurde) für Ihre Arbeit und in anderen Zeiten werden Sie vergessen.

Als Entwickler ist man auch ein Symbol der Gemeinde. Sie halten – ob Sie es mögen oder nicht – ein öffentliches Amt. Das überträgt die Verantwortung auf Sie, der Gemeinde zu helfen, Fragen respektvoll zu beantworten und andere Gemeindemitglieder zu ermutigen, in der Gemeinde aktiv zu sein.

Schlussfolgerung

Nachdem ich nun seit über drei Jahren in einer Free Software-Gemeinde Mitglied bin und sie von einer unreifen Sammlung von Mailinglisten zu einem reifen Organismus heranwachsen gesehen habe, kann ich überzeugt sagen, dass das politische System der Free Software-Gemeinden sehr gut funktioniert. Es scheint auch sehr flexibel zu sein, sodass es sich an Größe, Typ und die Bedürfnisse der Gemeinde anpassen kann. Obwohl die inneren Abläufe von Free Software sich stark von üblichen Geschäftsmodellen unterscheiden, gibt es jedoch auch viele Business-Möglichkeiten in Free Software-Gemeinden. Wenn erst einmal eine Firma die Politik versteht, kann eine bestimmte Gemeinde ein Segen sein, da die Gemeinde nicht nur Bugs beseitigt und andere übliche Entwicklungsarbeit leistet, sondern auch die Software erweitert und für ungeahnte Aufgaben umwandelt. Zope war am Anfang als reines Web-basierendes Content Management-System (CMS) gedacht, um mit Produkten wie Vignette zu konkurrieren. Aber die Gemeinde sah ein größeres Potenzial und Zope wird heute für viele Zwecke eingesetzt. Es kreuzt sogar als Logistik-Server auf, der viele Hunderttausende von Transaktionen pro Tag verarbeiten kann, angefangen von Kreditkartenberechnungen bis hin zum Versenden der Bestellungen an das Warenhaus.

Wenn eine Free Software-Gemeinde genug Auftrieb (mit oder ohne Hilfe einer Firma) bekommt, dann ist es möglich, dass die Ideale sich auf andere Software-Produkte und Firmen ausbreiten. Ein gutes Beispiel dafür ist die Lizenzänderung der Grafikbibliothek Qt zu einer General Public License (GPL)-kompatiblen Form. Die KDE-Gemeinde wurde so groß, dass es fast unmöglich wurde, alle Nutzungen zu überprüfen und die Lizenz zu erzwingen. Heute profitiert Trolltech (Hersteller von Qt) von den vielen Nutzern und Entwicklern, da diese das Toolkit ständig testen, verbessern und bekannt machen.

Free Software-orientierte sozial-demokratische Anarchien im hier skizzierten Sinn sind demnach für mich das einzig wahre Modell, um Software-Projekte erfolgreich durchzuführen. ■