

BASTELN 2000

oder

Die Freizeit schlägt zurück.

Geld muß fließen. Und zwar im Kreis

Während der letzten 100 Jahre oder so hatte eine solide gesellschaftliche und intrapersonale Arbeitsteilung Bestand: In der regulären Arbeitszeit erwirtschaften wir immer schneller Güter, die in die globale Ökonomie eingespeist werden und füllen unsere EC-Karte. Letztere nehmen wir nach Feierabend mit in die Freizeit und hauen das gesammelte Geld wieder raus: Hier wird gefressen, gehobbiet, und Lebenssinn gesucht, was das Zeug hält und dabei am laufenden Meter Güter verbraucht und dem globalen Markt Geld zugeschossen. Die allerorts fest etablierten Szeneökonomien sind kein Gegenbeispiel: Entweder man produziert hier Güter wie Kunst, Musik, Texte, Filme für eine schöne kleine Szene nach seinen eigenen Prämissen, was real cool ist, oder man arbeitet irgendwann nach der Logik der Sonys oder Bertelsmänner für den Weltmarkt, was unbestreitbar auch seinen ganz eigenen diskreten Charme hat.

Ausnahmen bestätigen die Regel.

Doch ein paar Szenen weiter ...

... bei unseren Freunden den Nerds finden wir auf einmal etwas ganz anderes. Das Phänomen ist bekannt, aber niemand weiß genau, warum es funktioniert: Netzwerke von teilweise mehreren zigtausenden Hackern produzieren in ihrer Freizeit freie Betriebssysteme wie Linux, Webserver wie Apache, Programmiersprachen wie Perl, Desktops wie KDE und verschenken die fertige Software. Anfangs nahm das keiner ernst, inzwischen äußern sich sogar Microsoft-Sprecher nervös. Apache läuft auf der Hälfte aller Webhosts, Linux hat gerade ca. 5% des sehr viel größeren Betriebssystemmarktes okkupiert, es sind hunderte Millionen bis Milliarden Dollar, die kommerzielle Hersteller von Konkurrenzprodukten nicht verdienen, weil es die freien Alternativen gibt. Und die Anteile der freien Produkte wachsen in vielen Bereichen stetig. Angeblich überlegt sich Microsoft gerade in diversen Geheimdokumenten Gegenstrategien (www.opensource.org/halloween.html).

Telearbeitende Freizeitparks

Die meisten Hacker haben ganz normale Du-und-Ich-Jobs wie Student, Programmierer, Sysadmin. Also nichts, was wirklich Sinn macht oder dem eigenen Leben distinguierende Wichtigkeit verleiht. Im Gegensatz zu anderen, deren Hobbies aus Pferde, Weggehen, Bodybuilding und Kuschnen bestehen, meißeln sie in ihrer Freizeit an einem Ganz Großen Projekt. In den Anfangszeiten des Hackens waren es meist Studenten in den computer science Forschungslaboratorien einiger amerikanischer Unis, die relativ abstrakte Unix-Tools zusammenklopften. Seit der Entstehung des ARPANET wurde es immer üblicher, Mitentwickler, Betatester und am Ende Nutzer für das Produkt über das Netz zu finden. Projekte werden announced, Mailinglisten und Newsgroups aufgesetzt, jeder, der sich berufen fühlt und programmieren kann, darf mitmachen. Viele tragen auch nur eine Codezeile zu dem Programm bei, einen Bugfix, eine winzig kleine Erweiterung oder Anpassung, bis vielleicht nach einigen Jahren ein sicheres, funktionales Programm entsteht, das teilweise die gesamte kommerzielle Konkurrenz in dem jeweiligen Bereich um Längen schlägt. Der telearbeitende Freizeitpark einmal anders als sich das unsere Zukunftsministerien vorstellen: Hunderttausende von bleichen Menschen mit geröteten Augen, die in den Nachtstunden zuhause vor Ihren Terminals gemeinsam das große Ding drehen und dabei nicht im Traum an die kommerzielle Verwertbarkeit denken.

Soziales Patchen

Wie und wieso funktioniert das? Die Community, die sich um ein Softwareprojekt bildet, verletzt wohl alle Lehrbuchsätze des Projektmanagements, der Organisationslehre und der Betriebswirtschaft: Keiner bekommt Geld, keiner ist wirklich weisungsbefugt, die Leute tun wann sie wollen was sie wollen, arbeiten über mehrere Kontinente verstreut und schaffen es trotzdem teilweise Killerapplikationen

zusammenzuschrauben. Die Produktionsgemeinschaften der Hacker stellen unbestreitbar einen neuen Typ von Organisation dar, der aus mindestens vier einander widersprechenden sozialen Samples zusammengepatcht wurde. Diese sind allerdings im Lauf der Jahre und des Rumprobierens eine ideale Symbiose eingegangen: Gurutum, Kommunismus, Marktwirtschaft und Anarchie. Eric Ramond, ein hackender Hacker-Soziologe, ist in einigen Papers der Sache etwas tiefer nachgestiegen (www.opensource.org/~esr/writings/).

Mad Professors

Wir finden bei fast jedem Softwareprojekt den großen Guru, der das ganze ins Leben gerufen hat und dessen Wort wie ein Sektengründer gilt. Abgestimmt wird praktisch nie. Zum Glück sind die Gurus mental nicht ganz so abgeraucht wie Bagwahn und Consorten und Gott sei Dank auch nicht so glatt und schleimoid wie die Zombies, die uns sonst immer als die Wichtigen des kommenden Jahrtausend präsentiert werden: Kein Jungunternehmergehebe, kein gutes Aussehen, keine Medienfertigkeiten: Es sind oft einfach charmante mad professors oder mad kids mit wirrem Blick, die ihr Ding fahren: Larry Wall, der Erfinder der Programmiersprache Perl, brachte es auf diesjährigen Perlkongress fertig, den Zuschauern schlicht zwei Stunden was vom Kreis zu erzählen. Wörtlich. Vom Kreis. Hatte eigentlich weder mit Perl, noch mit sonstwas nachvollziehbarem zu tun, war ihm aber offensichtlich ein Anliegen.

Egoless Egos

Ein Mad Professor macht noch keinen Freeware-Frühling, der Guru braucht natürlich Leute, die mit ihm zusammenarbeiten: Der Grund für die anderen, ihr Ego ein Stück zurückzustecken und an der Ganz Großen Gemeinsamen Sache mitzuarbeiten, liegt unter anderem darin, daß alles, was man macht, unwiderruflich allen gehört und nie von einem einzelnen privatisiert werden kann, auch nicht vom Guru. In den 80ern robbte die freie Software immer noch an der Marginalitätsgrenze herum. Ende der 80er wurde unter anderem von Richard Stallman (einem entlaufenen Chef-Programmierer des KI-Lab am MIT - ja, es sind immer die selben Läden) die GNU General Public Licence geschaffen (>www.gnu.ai.mit.edu/copyleft/gpl.html). Unter ihr (oder unter ähnlichen Bedingungen) wird fast die gesamte freie Software produziert: Die Bedingung lautet in etwa: Der Code ist frei, mach damit was du willst, aber alles was du machst, mußt du wieder frei geben. Dieser Ansatz nennt sich "Copyleft" und man kann ihn mit dem urkommunistischen "Jeder nach seinen Fähigkeiten, jedem nach seinen Bedürfnissen übersetzen". Wenn man will.

Das ganze führt dazu, daß jeder aus diesen Projekten lastwagenweise Ansehen, Credibility und Ruhm rausholen kann, weil der eigene Name draufsteht, aber keiner das Projekt durch Kommerzialisierung, oder durch Copyrightansprüche untergraben kann. Da nicht alle Hacker Mütter Therasas sind, ist es ganz vorteilhaft, daß sich die Credibility, die man sich bei einem freien Softwareprojekt holt, wiederum einfach versilbern läßt: Wer sich einen Ruf als Mitentwickler eines erfolgreichen Softwareprojektes gemacht hat, hat meistens keine großen Schwierigkeiten, gut dotierte Positionen oder Beraterverträge abzugreifen, bzw. Geld für eine Firma locker zu machen: Die Praxis zeigte, daß fast alle Heroen der freien Softwareproduktion bald gut versorgt waren. Womit sich der Kreis zur Geldwirtschaft wieder schließt.

Anarchistischer Basardarwinismus

Teilweise hat es eine richtig betriebsmäßig organisierte Produktion von freier Software gegeben (die freie Softwareschmiede GNU macht das). Letzlich hat sich aber fast überall der von Raymond als "Bazaar-Style" bezeichnete Arbeitsmodus durchgesetzt: Jeder macht im Prinzip was er will, patcht hier rum und da rum. Projekte wollen gute Leute finden. Gute Leute wollen vielversprechende Projekte finden, um ihren Ruhm zu mehreren und ihr Ego zu streicheln. Kreative Programmierer finden weiße Flecken auf der Softwarelandkarte, und stecken Claims ab, die sie beackern. Gute Projekte überleben, finden mehr Nutzer und Entwickler, schlechte werden ausgedarwinet. Ein Mischung aus Marktwirtschaft, Anarchie und Evolution. Jeder will eigentlich nur sein eigenes Bestes und dadurch wird alles besser: The survival of the fittest features. Und alles wird übers Netz kommuniziert von Leuten, die eigentlich ganz andere Sachen machen. Und am Ende funktioniert es so gut, daß jetzt auch kommerzielle Firmen wie Sun oder Netscape Code Ihrer Programme zum Patchen freigeben.

Und andere Software?

Wenn man sich die Entwicklung der freien Software ansieht, dann hört man vielleicht die Totenglocken für Teile der Software-Industrie leise bimmeln und möglicherweise auch für andere

Content-produzierende Branchen ein neues Stündchen einläuten: In postanalogen Zeiten ist die Produktion von Musik, Filmen oder Texten auch nur eine spezielle Form der Softwareproduktion, warum sollten sich hier nicht ähnliche Produktionsweisen etablieren? Man findet bereits einige Ansätze, von denen aber nur wenige, wie das Projekt Gutenberg (www.promo.net/pg) wirklich schon ernstzunehmen sind. Es gibt allerdings keinen Grund, warum nicht mittelfristig auch eine Encyclopedia Britannica oder ein netter Mainstream-Hit als Freeware produziert werden sollten.

Rewind to zero

Natürlich ist nichts neu: Die moderne Gesellschaft beruhte zwar darauf, daß die wichtigen, die eigentlichen Sachen in der Erwerbsarbeitssphäre stattfinden, aber das war nicht immer so: Unsere gesamten klassischen Staubfänger, wie die Werke Platos, Aristoteles oder Ciceros wurden nicht von professionellen Schriftstellern geschrieben. Die Jungs waren vielmehr Sklavenhalter, Großgrundbesitzer, Rechtsanwälte, Lehrer, Berater von Diktatoren und so weiter. Da einige dieser Menschen Ambitionen hatten, die über Mehren ihres Besitzes hinausgingen, haben sie in ihrer Freizeit ein wenig das Weltkulturerbe aufgefüllt. Sie hätten es wahrscheinlich relativ abstrakt gefunden, diese Art von Softwareproduktion für Geld zu machen.

Das Schöne heute ist, daß man keine Vermögen oder Latifundien braucht, um in seiner Freizeit das Ganz Große Projekt anzugehen. Wie die Hacker zeigen, tuts ein mittelguter Mac Job manchmal auch.