

Dies ist eine deutsche Übersetzung von [Felix Schwarz](http://www.felix-schwarz.name) (Homepage: [www.felix-schwarz.name](http://www.felix-schwarz.name)) unter tatkräftiger Mithilfe einiger Leser (s.u.). Ich habe mich bemüht, Formulierungen möglichst weitgehend zu übernehmen und die Tonalität zu Originalartikels zu treffen. Wo mir das nicht gelungen ist oder bessere Übersetzungen existieren, bitte ich um eine Mail mit Verbesserungsvorschlägen. Der eigentliche Artikel wurde inhaltlich nicht verändert, nur an einigen Stellen, die besonders häufig Missverständnisse hervorriefen, habe ich besonders gekennzeichnete Kommentare angebracht.

Der Originalartikel ist leider nur noch bei [archive.org](http://archive.org) einsehbar. Es gibt allerdings unter <http://linux.oneandoneis2.org/LNW.htm> einen vom Originalautor abgeänderten Artikel. Im Folgenden ist mit "ich" der ursprüngliche Autor Dominic Humphries gemeint.

*Im folgenden Artikel verwende ich nur den Oberbegriff "Linux", obwohl ich mich auf das GNU/Linux-Betriebssystem und verschiedene FOSS Projekte beziehe. Es liest sich einfach besser...*



## (Linux ist nicht Windows)

(Anmerkung:  bezeichnet einen Link zu einem relevanten Internetartikel.)

Wenn Sie (wie ich) einige Zeit in einem Linux-Forum verbringen, werden Sie letztendlich (wie ich) zunehmend verärgert über die Anzahl an Nachrichten, die etwa so lauten:

*"Hi! Ich benutze Linux seit ein paar Tagen und es ist eigentlich großartig. Aber es ist eine Schande, dass [dies oder das] nicht so funktioniert wie unter Windows. Warum schreiben nicht alle Programmierer die ganze Software neu, so dass sie mehr wie Windows funktioniert? Ich bin mir sicher, dass Linux viel mehr Benutzer hätte, wenn sie es täten!"*

Sie haben bei Ihrer Antwort vielleicht sogar jemanden angepflaumt, nur um anschließend von einem Linux-Neuling niedergemacht zu werden, der Ihre Antwort (basierend auf jahrelanger Erfahrung mit einem anderen Betriebssystem und ein paar Stunden Linux) so interpretiert, dass seine Idee revolutionär und brillant ist und nur Sie sie nicht mögen, weil Sie ein "altmodischer Linux-Nutzer" sind, der GUIs für eine Ausgeburt des Teufels hält und meint, jeder sollte gezwungen werden, das CLI ["Command Line Interface": Die Kommandozeile oder [Shell](#), Anm. von Felix Schwarz] zu benutzen.

Dieser Artikel zielt darauf ab, Neulingen zu erklären, warum Ihre Ideen eher abgelehnt als freudig begrüßt werden.

Zuallererst das beliebteste Argument: *"Wenn Linux das täte, würden viel mehr Leute von Windows zu Linux konvertieren!"*

Erlauben Sie mir etwas zu erklären, das für das Verständnis von Linux fundamental ist: Die [Linux-Gemeinschaft](#)  versucht **nicht**, dem durchschnittlichen Windowsbenutzer ein Ersatz-

Betriebssystem anzubieten. Das Ziel von Linux ist **nicht** "Linux auf jedem Schreibtisch".

Wirklich. Es ist wirklich nicht das [Ziel](#) . Ja, beides sind Betriebssysteme. Ja, beide **können** für den gleichen Zweck eingesetzt werden. Aber das macht Linux zu einer **Alternative**, nicht zu einem **Ersatz**. Dies scheint vielleicht nur eine unbedeutende Unterscheidung zu sein, aber sie ist tatsächlich extrem wichtig.

**Linux <=> Windows** ist wie **Motorräder <=> Autos**: Beides sind Fahrzeuge, die Sie über Straßen von A nach B bringen. Aber sie haben unterschiedliche Formen, unterschiedliche Größen, besitzen andere Steuerelemente und sie funktionieren auf völlig verschiedene Weisen. Man kann sie nicht beliebig austauschen. Beide sind für gewisse Dinge besser geeignet und für andere weniger gut. Sie sollten das Zweckmäßigere wählen und nicht einfach irgendeines und von diesem erwarten, dass es alles machen kann, was das andere könnte.

Jemand, der ein Auto fährt, könnte eines Tages in einem langen Stau stehen und ein Motorrad schnell neben sich vorbeifahren sehen. Er könnte die Fähigkeit des Motorradfahrers beneiden, etwas zu ignorieren, das ein lähmendes Problem für ein Auto darstellt. Wenn der Autofahrer dann sagen würde *"Ich weiß alles über Autos, also muss ich auch alles über Motorräder wissen!"*, läge er falsch.

- Wenn dieser Fahrer ein Motorrad kaufen würde und dann feststellt, dass er verwirrt wäre, dass das Gaspedal ein drehbarer Handgriff ist statt eines mit dem Fuß zu bedienenden Gaspedals, könnte er sich beschweren, dass Motorräder mit einem Gaspedal ausgestattet werden sollten.
- Wenn dieser Fahrer eine Frau und zwei Kinder hätte, könnte er es als Mangel betrachten, dass auf dem Motorrad nur eine Person Platz findet. Er könnte vorschlagen, dass das Motorrad neu konstruiert werden könnte, so dass vier Personen Platz fänden, mindestens aber zwei nebeneinander.
- Wenn dieser Fahrer beim Versuch zu fahren feststellen würde, dass er umkippt, weil er nicht daran gewöhnt ist, die Balance halten zu müssen, könnte er vorschlagen, dass Motorräder mit vier Rädern konstruiert werden sollten.
- Wenn der Fahrer feststellen würde, dass er sich in Kurven in der Schräglage wiederfindet, würde er vielleicht vorschlagen, dass Motorräder mit Stabilisatoren ausgestattet werden sollten, um sie in den Kurven aufrecht zu halten.
- Wenn der Fahrer sein Motorrad vor Diebstahl schützen wollte, könnte er sich beschweren, dass es keine Türen gibt, um potenzielle Diebe draußen zu halten, so dass sein Motorrad viel leichter gestohlen werden könnte als ein Auto.
- Wenn der Fahrer ein Motorradhelm als Last empfindet, könnte er vorschlagen, dass ein Airbag in den Lenker als Alternative zu dem lästigen Helm eingebaut werden könnte.

Und in jedem dieser Fälle würde er **falsch** liegen. Weil er denkt, dass ein Motorrad ein Auto **ersetzt**, meint er, es könne und solle alles tun, was ein Auto auch kann. Er denkt, dass es genauso funktionieren könnte wie ein Auto, dass "fehlende" Funktionen eines Autos einfach aufgepfropft werden könnten.

In der gleichen Weise machen wohlmeinende Linux-Neulinge Vorschläge, wie Linux ähnlicher dem gemacht werden könnte, an das sie gewöhnt sind. Und sie kommen aus den gleichen Gründen nicht voran. Linux und Windows können beide für die gleichen Zwecke eingesetzt werden, aber das gleiche gilt für Autos & Motorräder. Das heißt aber nicht, dass direkt eines gegen das andere tauschen kann und es heißt auch nicht, dass Eigenschaften unmittelbar austauschbar sind oder sein sollten.

Zu viele Leute denken, dass eine Migration von Windows zu Linux ähnlich sei wie ein Wechsel von BMW zu Mercedes. Sie denken, dass die Steuerungsmöglichkeiten die gleichen sein sollten, ihr

Wissen direkt übertragbar sein sollte und sämtliche Unterschiede sollten rein kosmetisch sein. Sie denken *"Ich brauche ein Auto, um eine Straße zu benutzen. Ich brauche ein Betriebssystem, um einen Computer zu benutzen. Autos funktionieren alle auf die gleiche Weise, daher sollten Betriebssysteme alle auf die gleiche Weise funktionieren."* Aber das stimmt nicht. *"Ich brauche ein Fahrzeug, um eine Straße zu benutzen. Ich brauche ein Betriebssystem, um einen Computer zu benutzen. Ich weiß, wie man Auto fährt, aber ich habe keine Ahnung von Motorrädern. Ich weiß, wie man Windows benutzt, aber ich habe keine Ahnung von Linux. - das wäre korrekt.*

Ein Windowsnutzer muss erkennen, dass er nur ein erfahrener **Windows**-Nutzer ist, nicht ein erfahrener **Computer**-Benutzer; genauso wie ein Autofahrer **nur** ein Autofahrer ist, nicht ein Alle-Straßenfahrzeuge-Fahrer. Ein Windows-Nutzer mit Linux muss erkennen, dass er gerade wieder ein Anfänger geworden ist, genauso wie ein Autofahrer auf einem Motorrad. Ein Windows-Nutzer muss gewillt sein zu lernen, dass es auch andere Wege gibt, eine Aufgabe zu erledigen, so wie ein Autofahrer sich daran gewöhnen muss, dass ein Lenker das Lenkrad ersetzt und ein Helm eine Notwendigkeit darstellt, obwohl er vorher nie einen brauchte. Und sie müssen darauf vorbereitet sein zu akzeptieren, dass "unterschiedlich" nicht "schlechter" heißt.

Diese simplen Fakten verursachen große Schwierigkeiten für die eingeweihten Windows-Benutzer. Sie kommen zu Linux mit vielen tief verwurzelten Windowsgewohnheiten und der Einstellung *"Ich weiß genau, wie man einen Computer zu benutzen hat, vielen Dank."* Das Problem ist nur, dass sie genau das nicht wissen. Sie wissen nur, wie man Windows benutzt. Wenn sie mit einem anderen Betriebssystem arbeiten, können diese "power user" genau die sein, die die größten Probleme haben: Sie müssen viel mehr umlernen.

Wo ein Neuling einfach sagen würde *"Ich weiß es nicht"* und anfangen würde, es auszuprobieren oder in Foren zu fragen, wird der Windows-Power-User sagen *"Ich weiß wie man das macht, ich mache nur das, das, das und dann ... Es funktioniert nicht! Blödes Betriebssystem!"* Und dann werden sie sagen *"Wenn erfahrene Nutzer wie ich es nicht zum Laufen bekommen, wird ein Anfänger keine Chance haben! Linux ist nicht einmal annähernd für die Desktop-Benutzung geeignet!"*. Sie erkennen nicht, dass all ihr Wissen gegen sie arbeitet und ihnen **mehr** Probleme bereitet als den **weniger** erfahrenen Nutzern. Sie haben den Fehler gemacht zu denken, dass Linux eine andere Software ist, die das gleiche macht wie Windows, wobei es in Wirklichkeit eine andere Software ist, die **andere** Dinge macht. Linux macht keinen schlechten Job für die gleichen Aufgaben, es macht einen **guten** Job für **alternative** Aufgaben.

Linux ist eine Alternative zu Windows, aber kein Ersatz. Es wird niemals ein Ersatz sein, weil beide unterschiedliche Ziele haben. Microsofts Ziel ist es, ihre Software auf möglichst vielen PCs zu haben, weil ihre Priorität auf Gewinn liegt. Linux hat solche Ziele nicht, weil Linux frei ist. Es hat andere Prioritäten.

Um das zu verstehen, muss man [FOSS](#)  verstehen. Es ist sehr verständlich, dass Linux-Anfänger es noch nicht verstehen - sie kennen es noch nicht. Sie sind gewöhnt, an proprietäre Software zu denken. Lassen Sie es mich daher erklären:

Typische FOSS Software wird von jemandem erstellt, der herumschaut, keine existierende Software findet, die er mag, und daher seine eigene schreibt. Weil er so ein netter Kerl ist, schmeißt er anschließend den Quelltext ins Internet und sagt: "Bedient euch". Er kann das machen, weil es ihn nichts kostet, die Software zu kopieren, so dass es ihn nicht mehr kostet, der ganzen Welt seine Software zu geben, als wenn er sie nur für sich selbst benutzen würde. Er erleidet keinen Nachteil dadurch, seine Software wegzugeben.

Wichtig ist aber: Er **profitiert** auch nicht davon, seine Software zu veröffentlichen. Ob sie von einer

Person oder eine Milliarde benutzt wird, macht für den Entwickler keinen Unterschied. Natürlich wird er Befriedigung dadurch bekommen, dass er ein populäres Produkt erschaffen hat: Die Anzahl der Benutzer kann ein netter Ego-Schub sein; ein Weg, wenn Sie so wollen, "of keeping score". Aber es bringt ihm kein Geld: Es ist FOSS.

Wenn die Software ein Erfolg ist, interessieren sich andere Leute dafür und helfen, sie zu verbessern. Das ist der größte Vorteil von FOSS: Jeder Benutzer ist ein potenzieller Entwickler. Jeder kann mitmachen und seinen Teil dazu beitragen, damit die Software besser arbeitet, mehr macht, weniger Fehler hat. Es ist großartig, wenn eine Software eine Gemeinschaft von Entwicklern anzieht. Es ist großartig **für die Software**. Es verbessert die Software. Es bereichert den Entwickler aber nicht. Es beansprucht nur mehr von seiner Zeit.

FOSS ist das genaue Gegenteil von Software wie Windows: FOSS dreht sich nur um die Software. Es geht nicht um die Anzahl der Endbenutzer. Software, die zwar gut funktioniert, aber nur wenige Benutzer hat, ist für die kommerzielle Software-Welt ein Flop, aber für FOSS ein Gewinn.

FOSS dreht sich um hochqualitative Software, Software, die Dinge TUN kann. Wenn Sie daran interessiert sind, sie zu benutzen, wird von Ihnen auch erwartet, dass Sie Zeit investieren, **wie** sie zu benutzen ist. Software wurde von Leuten geschrieben, die viel von ihrer Freizeit ohne persönlichen Gewinn investiert haben, und Ihnen kostenlos zur Verfügung gestellt. Das Geringste, was Sie tun können, um diesen Beitrag zurückzuzahlen ist, ein klein bisschen Ihrer eigenen Zeit zu investieren, bevor Sie sich beschweren, dass etwas nicht so funktioniert wie ähnliche Windows-Software.

*"Ah, jetzt habe ich dich." sagt der Anfänger süffisant "Es gibt Linux-Projekte mit dem Ziel, Windows zu ersetzen, und nicht nur eine Alternative zu sein."*

Es ist leicht zu erkennen, woher diese Idee kommt. KDE und Gnome zum Beispiel bieten eine Desktopumgebung an, die deutlich Windows-ähnlicher ist als typische Linux-Windowmanager und die CLI. Linspire ist eine Distribution, die nahezu vollständig auf der Idee basiert, Linux Windows-ähnlich zu machen.

Trotzdem - paradoxerweise - stützen diese Projekte **meinen** Punkt mehr als den des Anfängers.

Warum? Weil diese Projekte normale FOSS-Projekte sind, die sich nur darum drehen, Software besser zu machen. Der einzige Unterschied ist, dass eine der Definitionen von Qualität bei diesen Projekten ist *"Wie einfach kann ein Windows-Nutzer es verwenden?"*.

Sobald Sie das einbeziehen, können Sie nur zustimmen, dass dies 100% typische Linux-Projekte sind mit dem einzigen Ziel, Software besser zu machen. Diese Projekte werden von überdurchschnittlich selbstlosen Linux-Entwicklern betrieben: Sie erschaffen keine Programme für ihren eigenen Nutzen, weil sie Linux sehr gut kennen. Stattdessen programmieren sie Software nur für den Nutzen anderer Leute: Software, die den Übergang von Windows zu Linux einfacher macht.

Diese Entwickler haben erkannt, dass es Windows-Nutzer gibt, die zu Linux wechseln möchten, und sie haben große Anstrengungen hineingesteckt, um eine Linux-Umgebung zu erzeugen, die Windows-Nutzer angenehm vertraut finden. Aber sie haben das **nicht** getan, um zu versuchen, Windows zu ersetzen, obwohl das Endresultat diesen Eindruck vermitteln könnte. Das Endziel macht den Unterschied: Das Ziel ist nicht die Erschaffung eines Windows-Ersatzes, das Ziel ist, den Übergang von Windows-Benutzern zu Linux zu vereinfachen.

Feindseligkeit aus der Community gegenüber diesen Projekten ist nicht ungewöhnlich. Manchmal aus rationalen, verständlichen Gründen (*"KDE ist ein Ressourcenfresser, benutze daher Fluxbox"*), aber manches scheint ein irrationales, feindliches, *"Windows-ähnliche Software ist schlecht"*

Verhalten zu sein. In Wirklichkeit ist dies aber kein Anti-MS oder Anti-Windows-Verhalten. Stattdessen ist es die viel verständlichere Abneigung gegen das, was man nicht versteht.

Der 'typische' Linux-Nutzer ist ein Hobbyist: Er benutzt Computer, weil Computer Spaß machen, Programmierung Spaß macht, Hacking Spaß macht. Und Linux ist ein weitaus besseres Betriebssystem für einen Hobbyhacker: Er kann es nach Belieben bis zur untersten Ebene auseinander nehmen und wieder zusammensetzen.

Anmerkung von Felix Schwarz: Der Begriff 'Hacker'/'Hacking' sollte hier verstanden werden als 'eine Person, die sich daran erfreut, ein tiefgreifendes Verständnis der internen Arbeitsweise eines technischen Systems zu erlangen' (Quelle: [Wikipedia](#)). Ähnlich liest sich die [Definition des Chaos Computer Clubs](#). Demgegenüber steht der 'Cracker', der sich (häufig destruktiv) mit der Ausnutzung von Sicherheitslücken beschäftigt.

Allerdings gibt es zurzeit unter jenen, die zu Linux kommen, einen großen Anteil an solchen, die weder Hobbyisten noch Hacker sind. Sie wollen einen Computer, der "einfach funktioniert", einen Computer, der wie Windows funktioniert. Sie sind nicht daran interessiert, Linux zu konfigurieren, damit es so funktioniert, wie sie es wollen; sie wollen, dass es sofort nach dem Anschalten funktioniert.

Und das ist auch vollkommen OK, aber aus der Perspektive eines typischen Linux-Nutzers ist das wie jemand, der ein Legoauto haben will, das bereits aufgebaut und zusammengeklebt geliefert wird, damit nichts auseinander fallen kann. Ihrem Verständnis ist das total fremd. Der einzige Weg, wie sie reagieren können, ist mit einem erstaunten "*Warum könnte irgendjemand das wollen?*"

Es ist erstaunlich. Wenn Sie ein vorgefertigtes Modellauto haben wollen, kaufen Sie sich ein Spielzeugauto. Wenn Sie ein Auto haben wollen, das Sie selbst aufbauen und auseinander nehmen können, kaufen Sie Lego. Warum sollte jemand ein Legoauto wollen, das nur als gewöhnliches Spielzeugauto verwendet werden kann? Der ganze Witz an Lego ist, dass man Spaß dabei haben kann, es selbst zusammenzubauen!

So reagiert ein typischer Linux-Nutzer auf die "*Warum kann es nicht einfach funktionieren?*"-Abteilung: "*Wenn Sie etwas wollen, das einfach funktioniert, verwenden Sie Windows. Wenn Sie es hacken wollen, verwenden Sie Linux. Warum wollen Sie zu Linux wechseln, wenn Sie kein Interesse daran haben, Vorteile aus seiner Open-Source-Herkunft zu ziehen?*"

Die Antwort ist normalerweise, dass sie nicht wirklich zu Linux wechseln wollen. Sie wollen nur weg von Windows: Sie rennen weg vor den ganzen Viren, sie fliehen vor der Malware, sie streben danach, frei zu sein von den Beschränkungen, wie sie ihre bezahlte Software verwenden dürfen, sie versuchen, den Fängen des E.U.L.A. zu entkommen. Sie versuchen nicht, sich **in** Linux einzufinden, sie versuchen, von Windows **wegzukommen**. Linux ist schlicht und einfach die bekannteste Alternative.

**Kleine Werbeunterbrechung** (Felix Schwarz): Ein Freund von mir hat mit seiner Band Ende Mai 2006 eine Platte herausgebracht. Im Gegensatz zu den meisten CDs steht die Musik unter einer freien Lizenz (CC Commons) und kann auch in Tauschbörsen verbreitet werden. Und wem die Musik gefällt, sollte sich auch die CD kaufen. Das beste dabei: Die Künstler erhalten 100% der Einnahmen, haben aber auch die komplette Produktion vorfinanziert.

KICK ASS ROCK: [Shearer](#)

Früher oder später...

Sie könnten denken, *"Okay, das erklärt, warum Entwickler keine bewussten Anstrengungen unternehmen, damit ihre Software so funktioniert wie Windows. Aber sicherlich könnte Linux-Software eine GUI haben, die für Windowsbenutzer nutzerfreundlich ist, ohne mit den FOSS-Prinzipien in Konflikt zu geraten?"*

Es gibt ein paar Gründe, warum das nicht der Fall ist.

**Erstens:** Glauben Sie wirklich, dass jemand, der eine Software schreibt, ihr absichtlich eine [lausige Benutzerschnittstelle](#)  gibt?

Wenn jemand einen großen Batzen seiner Freizeit dafür opfert, um eine Software zu schreiben, wird er die Benutzerschnittstelle (UI) so gut wie möglich machen. Die UI ist ein sehr wichtiger Teil der Software: Es ist sinnlos, Funktionalität zu haben, auf die man nicht mittels einer UI zugreifen kann. Vielleicht wissen Sie es nicht, aber es gibt **immer** einen Grund, warum die UI so funktioniert wie sie es tut. Der Grund? Weil es die beste Oberfläche ist, die der Programmierer schaffen konnte.

Bevor Sie darauf bestehen, dass eine Windows-ähnliche Oberfläche die Software besser machen würde, behalten diesen Punkt im Hinterkopf: *Der Schöpfer dieser Software, ein Programmierer, der per Definition **weit** mehr als Sie über diese Software weiß, stimmt nicht mit Ihnen überein.* Er könnte daneben liegen, aber [die Chancen stehen schlecht](#). 

**Zweitens:** Es GIBT bereits nette, Windowsnutzer-freundliche GUI Oberflächen. Mir fällt spontan keine Funktion ein, die man nicht über eine GUI kontrollieren könnte, unabhängig davon auf welcher Ebene. Sie können den Kernel kompilieren (make xconfig), Ihre Firewall einstellen (fwbuilder), Ihre Festplatte partitionieren (qtparted) ... alles da, schön, interaktiv, intuitiv und benutzerfreundlich.

Aber der "Veröffentlichungszyklus" von Linux ist nicht wie bei Windows. Sie bekommen nicht das fertige, hochglanzpolierte Paket von Anfang an. GUIs bringen Komplexität, aber keine zusätzliche Funktionalität zur Software. Ein Entwickler setzt sich nicht hin und entwirft eine schöne GUI, die nichts tut. Er setzt sich hin und schreibt eine Software, die das tut, was er benötigt.

Das erste, was eine Software tut, ist, dass sie von der Kommandozeile aus benutzbar ist. Es wird vermutlich alle möglichen Aufrufoptionen und vielleicht eine längere Konfigurationsdatei haben. So fängt es an, weil diese Funktionalität notwendig ist. Alles andere kommt später. Und selbst wenn die Software eine nette GUI hat, ist es wichtig sich zu erinnern, dass es normalerweise immer noch komplett über die Kommandozeile und die Konfigurationsdateien gesteuert werden kann.

- Dies ist so, weil das CLI viele Vorteile bietet: Das CLI ist universell. Jedes Linux-System hat ein CLI. Jedes Programm kann von der Kommandozeile aus ausgeführt werden. Es ist einfach, Software auf anderen Rechnern per Kommandozeile zu steuern. Nichts davon stimmt für die GUI: Einige Linux-Rechner haben kein X11 Windowsystem installiert, manche Software hat kein GUI, manche Software ist nicht von GUI-Menüs aus verfügbar. Meist ist es nicht einfach oder praktisch, ein GUI-Werkzeug auf einem anderen Rechner zu benutzen.

Und zuletzt können verschiedene GUI-Oberflächen existieren, die dieselbe Aufgabe erledigen, und niemand kann sagen, welche Sie installiert haben.

Erinnern Sie sich also, wenn Sie fragen "*Wie mache ich...?*", wird Ihnen meistens gesagt, wie Sie es mit Hilfe der Kommandozeile erledigen können. Das **heißt nicht**, dass es **nur** über die Kommandozeile gemacht werden kann. Es zeigt einfach die relative Wichtigkeit, die die GUI während der Entwicklung eines Softwareprojekts hat verglichen mit der CLI

- Windows ist absolut GUI-zentriert. Es ist ein GUI-basiertes Betriebssystem mit einer lausigen Kommandozeile (die aber bald verbessert wird). Es gibt eigentlich keine Nicht-GUI Windows-Software. Das führt dazu, dass Leute meinen, eine GUI sei ein vitaler und integraler Teil einer Software. Aber unter Linux wird Software veröffentlicht, sobald sie funktionsfähig ist. Erst nachdem sie stabil und relativ fehlerfrei sowie einiges an Funktionalität zu bieten hat, ist es sinnvoll, eine GUI hinzuzufügen.  
Versuchen Sie, Software ohne eine hilfreiche GUI als Vorschau ("sneak preview") anstatt als fertiges Produkt zu sehen. FOSS ist sehr selten nur "fertig", sie wird immer verbessert. Mit der Zeit wird sie benutzerfreundlich gemacht. Aber meistens ist es wichtiger, die Software **funktionsfähig** zu machen anstatt sie nur äußerlich zu verschönern. Seien Sie froh, dass Sie die Funktionalität lange vor allen Warmduschern bekommen haben, die eine gute GUI brauchen, anstatt die Software von morgen bereits heute zu verlangen. FOSS ist mehr ein Weg als ein Ziel.

Das letzte, was Sie im Kopf behalten sollten, ist, dass GUIs für Software oft separate Programme sein werden. Sie könnten sogar komplett unabhängig vom originalen Programm, von völlig anderen Entwicklern programmiert werden. Wenn Sie eine GUI wollen, ist es nicht unwahrscheinlich, dass es zwei getrennte Installationen sein werden, statt alles in einem Stück zu haben.

Das bedeutet zugegebenerweise, dass es einen zusätzlichen Schritt gibt, um das schwer fassbare "Windows-mäßige" GUI-Verhalten zu bekommen, aber sollte nicht davon ablenken, dass Sie jetzt sofort fast alles über eine schöne Windows-ähnliche Oberfläche machen können. Merken Sie sich nur: Eine GUI ist normalerweise der LETZTE Schritt, nicht der erste. Linux bevorzugt das Aussehen nicht gegenüber der Funktion.

**Drittens:** Linux wurde ganz bewusst für den gut informierten, erfahrenen Nutzer gebaut statt für den ungebildeten Anfänger. Aus zwei Gründen:

- Unwissenheit mag ein Segen sein, aber nur für kurze Zeit. Wissen ist ewig. Es mag Tage, Wochen oder Monate dauern, um Ihren Wissenstand vom "Linux-Anfänger" zum "durchschnittlichen Linux-Benutzer" zu heben, aber wenn es einmal da ist, haben Sie noch viele Jahre der Linuxnutzung vor sich.  
Viel Code zu erzeugen, um Software einfacher für Neulinge zu machen, wäre so als ob man immer Stützräder an Fahrrädern anbringen würde. Es wäre einfacher für den Start, aber danach?  
Ich bin mir sicher, dass Sie jetzt kein Fahrrad mit Stützrädern kaufen würden. Und zwar nicht, weil Sie ein "anti-user-friendly" Freak sind. Nein, einfach weil es für Sie nutzlos wäre und auch nutzlos für alle außer Anfänger, Stützräder würden Sie nur stören.
- Unabhängig wie gut Software ist, sie ist immer nur so gut wie der Benutzer. Die sicherste Tür der Welt ist kein Hindernis für Diebe, wenn Sie das Fenster offen, die Tür unverschlossen oder die Schlüssel im Schloss lassen. Selbst mit dem besten Motor der Welt würden Sie nicht weit kommen, wenn Sie Diesel anstatt Benzin einfüllen.  
Linux legt **alle** Macht in die Hände der Benutzer. Das beinhaltet die Macht, es kaputt zu machen. Keiner gewinnt in dieser Situation. Der einzige Weg, dass Linux weiterhin gut arbeitet, ist, genug zu lernen, um zu wissen, was Sie tun. Wenn man es einem Nutzer einfacher macht,

an Funktionen herumzudoktern, die er nicht versteht, ist es nur wahrscheinlicher, dass er zufällig etwas kaputt machen wird.

**Viertens:** Wo haben Sie im Text oberhalb einen Weg gesehen, dass FOSS überhaupt davon profitieren würde, wenn es viele typische Windowsnutzer anziehen würde?

Nehmen Sie sich Zeit, lesen Sie es noch einmal, wenn Sie mögen. Ich werde warten.

Die Richtschnur von Linux und FOSS ist, "gute Software zu machen". Es geht **nicht** um "Windows-ersetzende Software herstellen". Das einzige, das eine Horde typischer Windowsnutzer zu Linux beitragen wird, sind Beschwerden. Über was werden sie sich beschweren? *"Es funktioniert nicht wie Windows."*

Nein, tut es nicht. Wenn es wie Windows funktionieren würde, würde Linux besch\*\*\*\* sein. Es wäre eine minderwertige Kopie, die niemand verwenden würde. Der Grund, warum Leute Linux so leidenschaftlich mögen, ist, dass es **nicht** so funktioniert wie Windows. Es **nimmt nicht an**, dass Sie ein ewig ungebildeter Anfänger sind, es **versteckt nicht** seine Abläufe und Mechanismen vor Ihnen.

Windows chauffiert sie herum; Linux gibt Ihnen die Schlüssel in die Hand und setzt Sie auf den Fahrersitz. Wenn Sie nicht fahren können, ist das Ihr Problem. Und Ihr Fehler. Viele Leute werden Ihnen helfen, wenn Sie fragen. Und wenn Sie einen Vorschlag machen, wie etwa einen Tempomaten einzubauen, könnten Sie auch irgendwohin kommen: Es belässt die Kontrolle beim Fahrer, aber nimmt ihm ein paar Dinge ab. Aber sie werden sehr schnell abgefertigt, wenn Sie versuchen, jeden zu überzeugen, dass Linux wirklich, wirklich einen Chauffeur braucht.

*"Aber Linux würde so viel etablierter!",* schreit der Anfänger.

Das könnte sein. Aber wie viele Linux-Entwickler würden profitieren, wenn Linux etablierter würde? Linux ist frei im Sinne von Freibier. Keiner der Leute, die Linux erschaffen haben, profitiert davon, dass es eine größere Benutzerzahl hat. Keiner der Leute in den Linux-Foren profitiert davon, dass Linux eine größere Benutzerzahl hat. Linux' Ziel ist **nicht** "eine große Benutzerzahl zu bekommen" - das ist das Ziel proprietärer Software.

**Anmerkung von Felix Schwarz:** Erfahrenen Anhängern freier Software wird folgende Aussage bekannt vorkommen: "Freie Software muss frei sein im Sinne von Freiheit und **nicht** wie Freibier." Der Kern freier Software liegt also in der Freiheit, nicht darin, dass sie u.U. kostenlos abgegeben wird. Im vorangegangenen Absatz wird aber von monetären Anreizen gesprochen und de facto wird die meiste freie Software kostenlos angeboten. Insofern ist Linux **auch** frei im Sinne von Freibier. Da dieser Begriff im Originalartikel genauso benutzt wurde, möchte ich die Begrifflichkeit hier nicht ändern. Ich persönlich hätte allerdings aber auch eine andere Formulierung wie z.B. "kostenlos" verwendet, um falsche Assoziationen zu vermeiden.

Linux' Ziel ist es, ein wirklich gutes Betriebssystem zu schaffen. Die Entwickler sind beschäftigt mit dem Hinzufügen neuer Funktionen, dem Entfernen von Fehlern und der Verbesserung bestehender Implementierungen. Sie beschäftigen sich nicht damit, Plakatwerbung aufzuhängen, wie gut ihr Zeug ist. Das sollte Ihnen etwas darüber sagen, wo ihre Prioritäten liegen.

Und schauen Sie, was dieser Ansatz mit Linux' Benutzerbasis getan hat: Es hat sie vergrößert. Linux begann winzig und ist riesig geworden. Der Grund, warum Linux so großen Anklang gefunden hat?

Weil der Fokus immer auf der Qualität lag. Die Benutzer, die von Linux angezogen wurden, wollten Freiheit und Qualität, die nur FOSS Ihnen geben kann. Linux wurde groß, weil es sich nicht darum kümmerte, wie groß es wurde. Entwickler waren nur damit beschäftigt, dass es funktioniert und zwar gut funktioniert, und so zogen sie Benutzer an, die ein Betriebssystem wollten, das funktioniert und zwar gut funktioniert.

Wenn man all das plötzlich wegwerfen würde und sich statt dessen darauf konzentrierte, sich darauf zu konzentrieren, dass Linux Windows ersetzt, würde dies genau die Sache zerstören, die Linux zu dem gemacht hat, was es heute ist. Es gibt da draußen Firmen, die Linux' Wachstum sehen und daran verdienen möchten. Sie sind frustriert über die GPL, die es ihnen sehr schwer macht, Linux zu Microsoft-Preisen zu verkaufen. *"Linux wird sterben, wenn es so offen bleibt,"* sagen sie, *"weil niemand Geld damit machen kann."*

Sie erkennen nicht, dass mit einem proprietären Linux die Gans geschlachtet würde, die die goldenen Eier legt. Linux wurde groß, weil es FOSS war und niemand versuchte, es zu einem Windows-Ersatz zu machen. Linux gedeiht, weil es Windows von einer Seite angreift, wo Microsoft es niemals besiegen kann: Offenheit und Qualität.

Für die meisten Windowsnutzer ist Linux eine schlechtere Windowskopie. Es hat weniger offensichtliche Funktionalität, weniger Integration und viel mehr Komplexität. Diese Art von Nutzern sieht Linux als ein schlechtes Betriebssystem. Und das ist richtig: Es erfüllt nicht ihre Bedürfnisse. Ihre Bedürfnisse sind ein Betriebssystem, das sehr einfach zu verwenden ist und alles tut, ohne dass sie etwas lernen müssten.

Windows wurde erschaffen für nicht technikkundige Nutzer. Unter diesen Nutzern ist die Wahrnehmung weit verbreitet, dass Linux schwer zu benutzen sei. Das ist nicht der Fall, aber es ist ein verständliches Missverständnis.

Linux ist tatsächlich erfreulich einfach zu benutzen. Wirklich. Es ist wirklich einfach zu benutzen. Der Grund, warum es nicht so wahrgenommen wird? Weil der Ausdruck *"einfach zu benutzen"* so weit deformiert wurde. Im alltäglichen Sprachgebrauch heißt *"einfach zu benutzen"* heute *"es ist einfach etwas zu tun, ohne vorher zu wissen, wie es zu machen ist"*. Aber das ist nicht wirklich *"einfach zu benutzen"*, oder? Das ist *"es ist einfach herauszufinden"*. Das ist wie der Unterschied zwischen:

- Ein Tresor mit einem Zettel darüber, auf dem steht *"Sie öffnen diesen Tresor, indem Sie das Rad zur 32 drehen, dann 64, dann 18, dann 9 und dann den Schlüssel umdrehen und die Klinke nach unten drücken."*

und

- Ein Auto kann geöffnet werden, indem man auf der Fernbedienung den "Öffnen"-Knopf drückt.

Es ist weit einfacher, das Auto zu öffnen, richtig? Ein Knopf irgendwo in der Nähe des Autos im Gegensatz zu sehr spezifischen Drehungen. Aber es ist einfacher für jemanden, der nicht weiß, wie beide zu öffnen sind, den Tresor zu öffnen als das Auto: Der Tresor besitzt klare Instruktionen, während das Auto nur Knöpfe hat, die nicht mal am Auto direkt zu finden sind.

Mit Linux ist es genauso. Es ist einfach zu benutzen, wenn Sie wissen **wie**. Es ist einfach zu **benutzen**, aber nicht immer einfach zu **lernen**. Nur wenn Sie willens sind, Zeit in das Lernen von Linux zu investieren, werden Sie es einfach finden. Je weiter Sie eine Aufgabe in einfache Schritte zerlegen, umso mehr Schritte müssen Sie machen, um diese Aufgabe zu bewerkstelligen.

Nehmen wir als wirklich einfaches Beispiel, diese beliebige Übung: Sie wollen fünf Zeilen (Absätze)

aus der Mitte eines Textdokuments ans Ende verschieben.

In MS Word, MS WordPad oder MS Notepad, alles "benutzerfreundliche" Windows Texteditoren, der schnellste Weg, dies zu tun, ist:

- Strg-Shift-Runter
- Strg-Shift-Runter
- Strg-Shift-Runter
- Strg-Shift-Runter
- Strg-Shift-Runter
- Strg-X
- Strg-Ende
- Strg-V

(Angenommen, Sie verwenden die Tastatur, ansonsten diverse Mausklicks und ein verlässliches Autoscrolling, um den Text nach unten zu ziehen.)

Mit vi allerdings machen Sie:

- d5d
- Shift-g
- p

(Oder wenn Sie vi **wirklich** gut kennen, wird ":1,5m\$" auch funktionieren!)

**Anmerkung von Felix Schwarz:** Verschiedene Leser haben berichtet, dass die obige vi-Befehlsfolge ":1,5m\$" nicht korrekt ist. Das angegebene Kommando würde die Zeilen 1-5 an das Textende verschieben. Richtig wäre hier ":",.+4m\$" (Karsten W. erklärt: ". steht für die aktuelle Zeile, .+4 entsprechend für aktuelle +4, also steht :,.+4 für 5 Zeilen mitten im Dokument."). Den gleichen Effekt erhält man auch mit ":",+4m\$". Vielen Dank an Jürgen B., Karsten W. und Sebastian Schaezel.

Vi, der so "benutzer-unfreundlich" ist, wie es nur geht, schlägt Microsofts Produkte um Längen. Warum? Weil vi für *Funktionalität* entworfen wurde, während Microsoft versucht, "benutzerfreundlich" zu sein. Microsoft bricht alles in einfache Schritte runter und daher braucht es weit mehr Schritte, um die gleiche Aufgabe zu erfüllen.

Dadurch ist vi deutlich schneller und einfacher zu benutzen für fast alle Aufgaben zum Ändern von Text. Zumindest so lange, wie Sie wissen, wie es zu benutzen ist. Wenn Sie nicht wissen, dass "d5d" heißt "Packe fünf Textzeilen in den Puffer und lösche sie aus dem Text", werden Sie Schwierigkeiten haben, vi zum Funktionieren zu bringen. Aber WENN Sie es wissen, dann kommen Sie mit vi am schnellsten zum Ziel.

Wenn irgendein Anfänger sieht, wie schnell und einfach ein erfahrener vi-Benutzer Dinge tun kann, wird er bereitwillig zustimmen, dass vi Word für das Ändern von Text überlegen ist. Anschließend versucht er es selbst. Er startet vi und erhält einen Bildschirm voller "~"s und wenn er tippt, erscheint nichts auf dem Schirm.

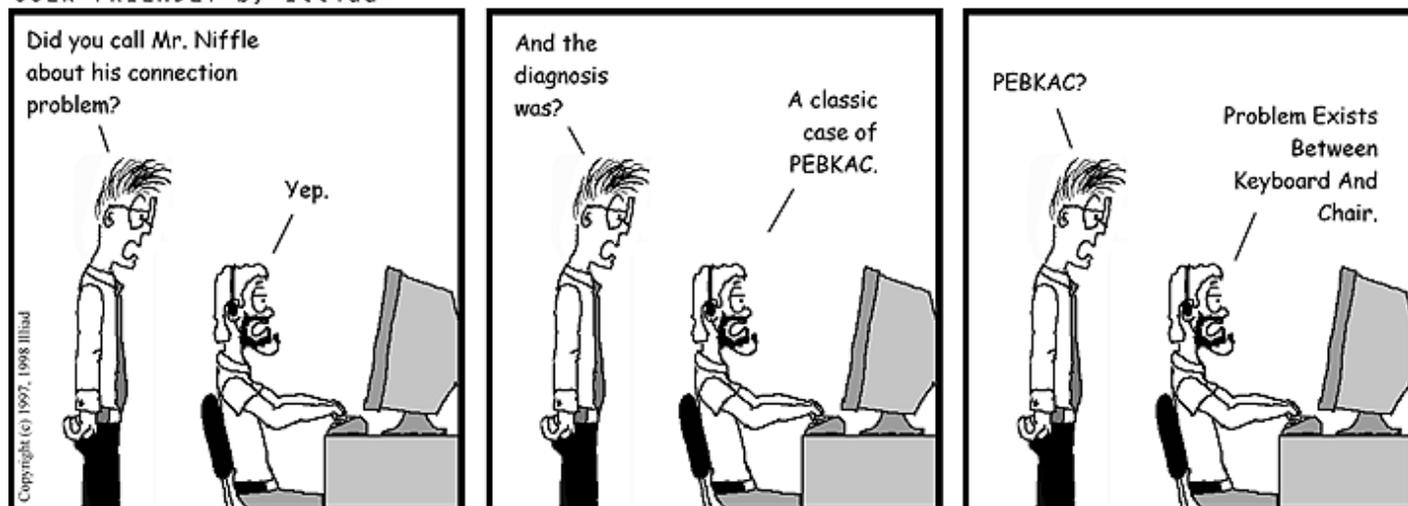
Er findet etwas heraus über die Texteingabe- und Kommando-Modi und versucht, vi mit einem begrenzten Wissen seiner Funktionen zu verwenden. Er kämpft, weil es so viele Dinge gibt, die er lernen muss, bevor er vi zum Funktionieren bringen kann. Dann beschwert er sich "vi wäre so viel

*besser, wenn es so einfach zu benutzen wäre wie Word!"*

Aber das tatsächliche Problem ist *"Ich weiß nicht, wie man vi benutzt und ich habe keine Lust, es zu lernen."* Aber das würde heißen, dass es sein Problem wäre, daher wirft er es der Software vor. Ungeachtet der vielen tausend Leute, die vi ohne jedes Problem nutzen: Es ist zu schwer zu benutzen und muss geändert werden!

Und glauben Sie mir, wenn er einen Texteditor schaffen könnte, der so *"benutzerfreundlich"* wie Word und so funktional wie vi ist, würde er nur mit Beifall empfangen werden. Vermutlich würde er sogar den Nobelpreis für Herausragende Intelligenz bekommen, weil niemand bisher dazu in der Lage war. Aber nur Jammern darüber, dass vi schwer zu benutzen ist, wird mit Hohn begegnet, weil es kein Problem mit vi ist, sondern ein Problem mit ihm.

USER FRIENDLY by Illiad



[Von UserFriendly.org](http://UserFriendly.org) Copyright © 2004 J.D. "Illiad" Frazer.

Es ist, als ob man den Pinsel von Leonardo da Vinci kaufen und sich dann darüber beschweren würde, dass man immer noch nicht malen kann. Der Pinsel war es nicht, der die Mona Lisa gemacht hat, es war die Fähigkeit des Künstlers. Der Pinsel ist ein Werkzeug, das auf die Fähigkeiten des Benutzers angewiesen ist. Es gibt keinen anderen Weg, diese Fähigkeiten zu bekommen außer Übung.

Das gleiche gilt für vi. Das gleiche gilt für viele Linux-Programme, über die neue Benutzer sich beschweren, dass sie "zu schwer" oder "nicht intuitiv genug" sind, ganz gleich ob sie über einen Texteditor, einen Paketmanager oder die Kommandozeile selbst sprechen.

Bevor Sie darauf beharren, dass etwas in Linux verbessert werden muss, gibt es eine wichtige Frage: *"Haben erfahrene Benutzer ein Problem damit?"*

Wenn die Antwort *"Nein"* lautet, liegt das Problem bei Ihnen. Wenn andere Leute es erfolgreich verwenden, warum können Sie das nicht? Haben Sie sich die Zeit genommen, um die Benutzung zu lernen? Oder haben Sie erwartet, dass es für Sie funktioniert, direkt nach dem Wort "Start"?

*"Benutzerfreundlichkeit"* und *"reine Funktionalität"* schließen sich aus. All die kleinen Knöpfe und Drop-Down-Menüs, die wichtig sind, um eine Software einfach benutzbar zu machen, sind nur Hindernisse, die dem erfahrenen Benutzer im Weg sind. Ein anderes Beispiel soll diesen Unterschied verdeutlichen: Man kann einen Weg von A nach B finden, indem man entweder Karte und Kompass benutzt oder einfach den Straßenschildern folgt. Letztlich wird man auch immer zum Ziel kommen, indem man den Schildern folgt, aber es wird doppelt so lange dauern, als wenn man



Das ist, was Linux **ist**. Das ist, worum es geht. Leute migrieren zu Linux, weil Sie Viren, BSODs ["Blue Screen of Death", der blaue Bildschirm, wenn Windows komplett abgestürzt ist, Anm. von Felix Schwarz] und Spyware leid sind. Das ist verständlich. Aber diese Leute **wollen nicht** Linux. Sie wollen eigentlich nur Windows ohne die Fehler. Sie wollen nicht wirklich Linux. Warum sollte Linux also **sie** wollen?

Aber wenn sie Linux wegen der Viren und Spyware **ausprobieren** und **dann** entscheiden, dass sie die Idee eines Betriebssystems lieben, das **sie** kontrollieren... Dann wollen Sie Linux um seiner selbst willen. Und das ist der Punkt, an dem Linux auch sie will.

Bevor Sie entscheiden, ob Sie zu Linux wechseln wollen, fragen Sie sich "*Warum will ich wechseln?*"

Wenn die Antwort ist "*Ich will ein Betriebssystem, das die ganze Macht in die Hände des Benutzers legt und von ihm erwartet, dass er weiß, wie sie zu gebrauchen ist.*": Holen Sie sich Linux. Sie werden eine große Menge Zeit und Mühe investieren, bevor Sie bekommen, was Sie wollen, aber dann werden Sie schließlich belohnt mit einem Computer, der genauso so arbeitet, wie Sie es wollen.

### **ABER...**

Wenn die Antwort lautet "*Ich will Windows ohne die Probleme*": Machen Sie eine saubere Installation von Windows XP SP2, installieren Sie eine gute Firewall, installieren Sie ein gutes Anti-Viren-Programm, benutzen Sie niemals den IE zum Browsen im Web, aktualisieren Sie regelmäßig Ihr System, starten Sie nach jeder Software-Installation neu und lesen Sie etwas über gute Sicherheitsregeln. Ich habe Windows selbst benutzt von 3.1 über 95, 98, NT und XP und hatte **nicht einmal** einen Virus oder Spyware oder wurde gehackt. Windows **kann** ein sicheres und stabiles Betriebssystem sein, aber es ist auf **Sie** angewiesen, damit es so bleibt.

Wenn die Antwort lautet "*Ich will einen Ersatz für Windows ohne die Probleme*": Kaufen Sie einen Apple Mac. Ich habe wundervolle Dinge über MacOS X "Tiger" gehört und sie haben Hardware, die gut aussieht. Es wird Sie einen neuen Computer kosten, aber damit werden Sie bekommen, was Sie wollen.

In beiden Fällen **wechseln Sie nicht** zu Linux. Sie werden enttäuscht sein sowohl von der Software als auch von der Community. **Linux ist nicht Windows.**

---

Wenn Sie Anmerkungen zu diesem Artikel haben - gut oder schlecht - [schreiben Sie mir eine E-Mail](#) (Anm. des Übersetzers: Anmerkungen zur deutschen Übersetzung bitte an [Felix Schwarz](#)) oder benutzen Sie [mein Blog](#).



This work is copyright and belongs to Dominic Humphries. It may be redistributed under a [Creative Commons License](#). This page's URL **must** be supplied in attribution.

**Nachwort von Felix Schwarz:** In der Tat ist diese Übersetzung ein gutes Beispiel für den klassischen Open Source-Entwicklungsprozess. Eines Tages im Januar 2006 hatte ich nachmittags mal 3-4 Stunden Zeit und habe diese genutzt, um eine Übersetzung ins Deutsche anzufertigen. Natürlich war sie nicht perfekt (sie ist es immer noch nicht) und enthielt diverse Fehler, aber man konnte sie lesen. Ich stellte die deutsche Version online und Dominic Humphries verlinkte sie. Im Laufe der Wochen lasen mehrere Tausend Besucher den Artikel und ab und an bekam ich kurze oder längere E-Mails mit Anmerkungen und Verbesserungsvorschlägen, die ich

dann auch eingebaut habe.

So wurde dieser Artikel im Laufe der Zeit immer wieder verbessert und der Dank muss auch an die unten aufgeführten Personen gehen, die mit ihren hilfreichen Kommentaren dazu beigetragen haben.

## Änderungen

- 11.02.2012** kleinere Verbesserungen der Übersetzung (Dank an Daniel Hommrich)
- 22.06.2008** kleinere Verbesserungen der Übersetzung (Dank an Karlheinz, Patrik Fimml und Heiko Hartmann), noch eine weitere vi-Ergänzung (Dank an Sebastian Schaeztl)
- 20.06.2008** Hinweis auf die zweite Version des Originalartikels (Dank an Jeremy Boy, Nikolaus Klepp, Matthias Kretz für die Hinweise)
- 08.12.2007** kleine Grammatikfehler ausgebessert (Dank an fuesika und Wilhelm Kweton), tote Links korrigiert (Dank an Marek Macioschek, David Sieg und Daniel Schmid für die Hinweise)
- 14.10.2007** kleine Satzkorrekturen (Dank an Thomas Fiedler und Edgar Hoffmann), Erklärungen englischer Akronyme wie CLI und BSOD auf Anregung von Thomas Fiedler, weitere Anmerkung zu vi-Befehlen hinzugefügt.
- 01.10.2007** Link auf deutsche Lizenzseite, korrigierter Link auf Original-Blog (Dank an Tobias Hobmeier)
- 26.09.2007** diverse Fehlerbeseitigungen und Verbesserungen der Übersetzung (Dank an Heiko Kempe, Ralph Jud, Ute Köster und Miriam Kuethé)
- 23.09.2007** Verbesserung der Übersetzung "gutaussehend" (Dank an Sebastian Linke)
- 26.08.2007** Verbesserung der Übersetzung: Maschine -> Motor (Dank an Nico)
- 03.07.2007** bessere Formulierung bei Motorrad-"Schieflage" (Dank an Uwe Walter)
- 11.06.2007** weitere Fehlerbeseitigungen (Dank an Dominik Geyer und Wolfgang Seebacher), Anmerkung zu Hackern und Crackern auf Anregung von Dennis Brakhane hinzugefügt.
- 24.01.2007** richtiger vi-Befehl (Dank an Jürgen B., Karsten W.), bessere Formulierung (Dank an Peter Henning), Anmerkung zu Freibier
- 28.11.2006** diverse Korrekturen (Dank an Bruno Holliger, Thomas Schneider, Sebastian Restel)
- 27.08.2006** "Stück Software" ist Denglisch (Dank an Danny Tank)
- 18.08.2006** Korrekturen im Satzbau (Dank an Jon Kowal)
- 05.08.2006** ein paar Fehler beseitigt (Dank an Norman Steinbach)
- 25.07.2006** diverse Rechtschreibfehler beseitigt (Dank an Jacqueline Dürr und insbesondere Andreas Drop)
- 17.06.2006** Tippfehler verbessert (Dank Sebastian Wagner)
- 13.06.2006** kleine Rechtschreibkorrekturen (Dank an anonymous und Timo Taglieber)
- 05.06.2006** kleiner Rechtschreibfehler beseitigt
- 04.06.2006** kleiner Rechtschreibfehler beseitigt (Dank an Robert).
- 03.06.2006** kleinere Rechtschreibfehler beseitigt (Dank an Stefan Schoenemann), Link auf Originalartikel angepasst.
- 17.05.2006** diverse englische Originalpassagen gelöscht, kleine Satzänderung auf Grund eines Hinweises von Christian Rumpf)
- 12.04.2006** zahlreiche Verbesserungen der Übersetzung (großen Dank an David Wührer)
- 12.04.2006** kleine Satzänderung (Dank an Edward von Flottwell)
- 22.03.2006** Link auf deutsche Wikipedia geändert (Dank an anonymous)
- 15.01.2006** kleine Korrekturen und Verbesserungen (Dank an Jessica Campbell)

**12.01.2006**

initiale Übersetzung