

**Joachim Graf**

# Murphys Computergesetze

oder



Wie das Gesetz, daß alles,  
was schief/gehen kann,  
auch schief/geht,  
durch den Computer  
opti\_m\_iert wird.



## Danksagung

Ausdrücklich bedanken möchte ich mich bei der deutschen und internationalen Computerindustrie. Ohne sie hätte es dieses Buch nie gegeben.

## Erklärung

Jede Ähnlichkeit mit lebenden Personen, bestehenden Firmen, existierenden Produkten, verfügbaren Marken oder vorhandene Vertriebsstrategien in diesem Buch ist nicht beabsichtigt, sondern unvermeidlich.

## Widmung

Meiner Frau Gerti sowie Sven, Lars und Manuel für ihre täglichen Anregungen zum Kapitel »Menschen« sowie meinem Partner Daniel Treplin zum Kapitel »Computerfreaks«

## Vorwort

Wohl kein Forschungsergebnis hat mehr zum Verständnis unserer modernen Industrie- und Informationsgesellschaft beigetragen wie das Gesetz von Murphy. Wer sich vor Augen führt, daß alles, was schiefgehen kann, auch schiefgehen wird, der wird von einem tiefen Verständnis für die Welt, das Leben an sich und den gesamten Rest durchdrungen.

Die konkreten Anwendungen von Murphys Gesetz reichen von »A« wie »Atomwirtschaft« (»Je gefährlicher und unsinniger ein Projekt ist, um so heftiger wird es von Bonner Politikern unterstützt«) bis »Z« (»Wenn Du von A bis Z beschreiben willst, fällt entweder zu »A« oder zu »Z« kein Beispiel ein«). Die murphologische Forschung hat eine Vielzahl von Veröffentlichungen hervorgebracht.

Für fast jede Quelle täglicher Pannen - seien es Frauen oder die gegenwärtige Bonner Regierung - es sind murphologische Arbeiten erhältlich. Ein Bereich wurde allerdings bisher aus für den Autor nicht nachvollziehbaren Gründen ausgespart: die elektronische Datenverarbeitung.

Dabei tragen Computer wie kaum ein anderes als Objekt getarntes Subjekt unseres Alltags dazu bei, das Gesetz von Murphy immer wieder aufs neue zu bestätigen. Alle nur denkbaren pannenträchtigen Komponenten sind in einem Computersystem vereint: Zentraleinheit, Monitor, Massenspeicher, Erweiterungskarten, Tastatur und weitere Peripheriegeräte arbeiten hart daran, möglichst unverträglich zueinander zu sein. Wo dies nicht ausreicht, unterstützt die Dreieinheit aus Programmierer, Anwendungsprogramm und Anwender das physikalische Gesetz der Entropie, nach dem die Natur einen Zustand des größtmöglichen Chaos anstrebt.

Der vorliegende Band versucht daher, diese Lücke in der murphologischen Forschung zu schließen.

Dem Autor kam dabei seine jahrelang Beschäftigung mit Computersystemen als Anwender, Programmierer und EDV-Fachjournalist - also: stets als Opfer - zugute.

Sollte der Leser in diesem Buch einen fachlichen Irrtum oder einen Rechtschreibfehler entdecken, so ist auch dies eine Bestätigung von Murphys Gesetz. Denn

1. war der Autor bemüht, solche Fehler zu vermeiden und
2. ist dieses Buch auf einem Computer mit einem Textverarbeitungsprogramm entstanden und wurde mit einem automatischen Rechtschreibprüfungsprogramm korrigiert.

## A. Murphologische Grundlagen

Murphys Gesetz ist die Präzisierung und logische Weiterentwicklung der allgemeinen Entropielehre, nach der alle Teilchen des Universums bestrebt sind, sich in größtmöglicher Unordnung anzuordnen. Die Erkenntnis, daß besagte Teilchen auf dem Weg dorthin Dir mindestens einmal auf die Zehen fallen, führte zu:

### **Murpys Gesetz:**

Wenn etwas schiefgehen kann, dann wird es auch schiefgehen. Mit der Erfindung des Computers versuchte der Mensch zum ersten Mal, unbelebter Materie eine gewisse Intelligenz einzuhauchen.. Ein fataler Entschluß. Denn bis zum heutigen Tage sind Computer zwar weder intelligent noch kreativ. Jedoch Heimtücke, Hinterhältigkeit und Verschlagenheit sind bei ihnen bereits optimal entwickelt. So können wir Murphys Gesetz erweitern durch die

### **erste digitale Ableitung:**

Murphys Gesetz wird durch Computer optimiert.

Da aber moderne Computer heute bereits mehr als die eine Sache gleichzeitig machen können, folgt nahtlos die

### **zweite digitale Ableitung:**

Alles geht auf einmal schief.

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Mit der Erfindung von Checksummen, Korrektur- und Backup-Programmen sowie fehlertoleranten Systemen erschließt sich dem staunenden - zum Objekt degradierten - Menschen die Vielseitigkeit der elektronischen Datenverarbeitung durch die

## **dritte digitale Ableitung:**

Es geht auch schief, wenn es eigentlich nicht schiefgehen kann.

Werten wir den reichhaltigen Erfahrungsschatz von Anwendern, Programmierern, Entwicklern und anderen armen Schweinen aus, können wir nun Murphys Gesetz und seine digitalen Ableitungen auf den elektronischen Alltag gesetzmäßig anwenden:

## **Erste elektronische Anwendung von Murphys Gesetz:**

Bei Computern ist nichts undenkbar, geschweige denn unmöglich - außer dem Wünschenswerten

## **Zweite elektronische Anwendung von Murphys Gesetz:**

In der Welt der EDV enden Pannen nicht, sondern gehen, einander überlappend, ineinander über.

## **Dritte elektronische Anwendung von Murphys Gesetz:**

Computerpannen warten geduldig auf den ungünstigsten Zeitpunkt, um dann erbarmungslos zuzuschlagen.

## **Daniels Korrekturanmerkung zur dritte elektronischen Anwendung:**

Kann denn eine Panne zuschlagen?

## **Joachims Antwort auf Daniels Korrekturanmerkung:**

Wart's nur ab.

## **Vierte elektronische Anwendung von Murphys Gesetz:**

Man kann sich bei Computern auf nichts verlassen. Nicht einmal darauf, daß man sich auf nichts verlassen kann.

## **Fünfte elektronische Anwendung von Murphys Gesetz:**

1. Du kannst niemals einer großen Panne entgehen, indem Du eine kleine produzierst.
2. Im besten Fall wird sich die kleine Panne zur großen hinzugesellen, um diese zu unterstützen.

## **Sechste elektronische Anwendung von Murphys Gesetz:**

Niemand kann sich so viele Pannen vorstellen, wie sie im Inneren eines Computers passieren.

## **Bernhards Seufzer:**

Du kannst Dir aber sicher sein, daß Du jede einzelne am Leibe verspüren wirst.

## **Bernhards Folgerungen:**

1. Wahrlich große Pannen verhalten sich wie Fernsehintendanten: Sie produzieren möglichst viele Wiederholungen
2. Billige Pannen gibt es nicht.
3. War eine Panne doch einmal billig, dann hast Du ihren wahren Umfang noch nicht erkannt.

Da Computer und elektronische Datenverarbeitung zumindest ursprünglich etwas mit Mathematik zu tun hatten, ist die allgemeine Einführung in die Computer-Murphologie nicht vollständig ohne die mathematische Begründung von Murphys Gesetz. Sie wollen jedoch bitte berücksichtigen, daß Mathematik und binäre sowie menschliche Logik einerseits und EDV-Anlagen nichts, aber auch gar nichts miteinander zu tun haben. Selbstverständlich mit der Ausnahme, daß man diesen Satz zu beweisen sucht:

## **Mathematische Begründung von Murphys Gesetz:**

Die exakte mathematische Formel für Murphys Gesetz in der Mathematik lautet:  $1 + 1 = 2$ , wobei "=" ein Symbol mit der Bedeutung "selten, wenn überhaupt" ist.

## **Abweichungstheorem:**

Der Unterschied zwischen digitaler Logik und Murphys Gesetz besteht darin, daß man nach der digitalen Logik eigentlich davon ausgehen müßte, daß immer alles nach der gleichen Methode schiefgeht.

## Binäre Übersetzung des Abweichungstheorems:

Wenn Null besonders groß ist, ist es beinahe so groß, wie ein bißchen Eins.

## B. Täter

Wo immer auf dieser Welt ein Computer und ein Mensch zusammenkommen, gibt es Täter und Opfer. Wer Täter und wer Opfer ist, bestimmt der persönliche Bezugspunkt: Egal, was passiert, man gehört immer zu den Opfern. Schließlich gilt wie überall im Leben auch in der Welt der Computer das

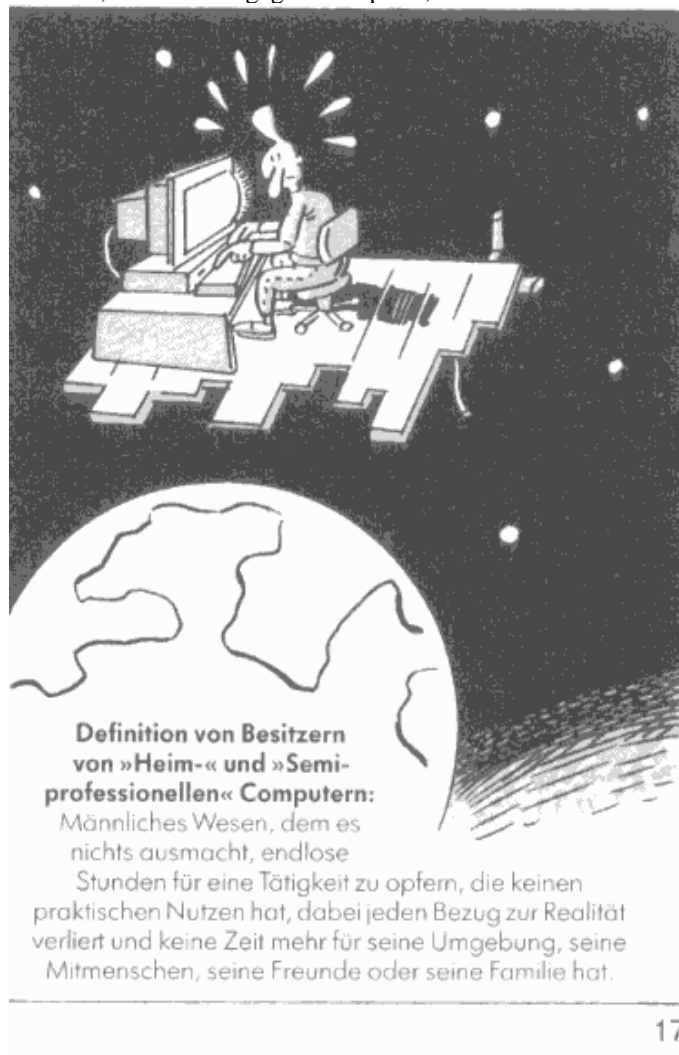
### Allgemeine Opfergesetz:

Egal, auf welcher Seite Du gerade stehst - diese Seite wird verlieren.  
Wenn Du die Seite wechselst, wird sich das Schlachtenglück wenden.

Das allgemeine Opfergesetz auf den EDV-Bereich angewendet und präzisiert, ergibt die:

### Digitale Quartettregel:

1. Bist Du ein Anwender, so wirst Du gegen Computer, Hardwarehersteller, Anwender und Programmierer verlieren.
2. Bist Du ein Hardwarehersteller, so wirst Du gegen Computer, Anwender und Programmierer verlieren.
3. Bist Du ein Programmierer, so wirst Du gegen Computer, Hardwarehersteller und Anwender verlieren.



### Konsequente Folgerungen aus der digitalen Quartettregel:

1. Es kann keine menschlichen Gewinner geben.
2. Der Computer gewinnt immer.

### Erweiterte Folgerungen aus der digitalen Quartettregel:

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Sollte der Computer einmal nicht gewinnen, dann gewinnen Software oder Peripheriegeräte, im besten Fall die Steckdose.

Aufgrund dieser Ableitungen beschäftigen wir uns im folgenden Abschnitt vorwiegend mit den Siegern im allgemeinen Existenzkampf: der Computer- Hard- und Software.

Doch jeder Computerbesitzer weiß, daß nicht allein die Unbillen der Technik auf ihn lauern. Wer nicht das Glück hat, alleine mit seinem Computer Wohnung oder Büro zu teilen, dem drohen weitere Gefahren von sogenannten "Freunden", von Mitbewohnern, Kollegen und Familienangehörigen. Kurz: Die Kombination aus heimtückischer Technik und Computerlaien richtet den maximal möglichen Schaden an.

Kapitel 4 hingegen wird das Allgemeine Opfergesetz auf die Gruppe der Nicht-Computerbesitzer angewendet: Hier galt das analytische Interesse des Autors den Gesetzmäßigkeiten, die - aus der Sicht eines Normalmenschen - bei einem Zusammenprall von diesem mit einem Computerfreak entstehen.

## 1.0 Hardware

Hardware ist der geglückte Versuch, die Fehler der Software vorherzuahnen, vorhandene Irrtümer zu optimieren, zu speichern und mit immer höherer Geschwindigkeit zu produzieren.

Hardware setzt sich zusammen aus Computer, Eingabegerät, Drucker, Massespeicher sowie anderer in Silizium geätzter Heimtücke.

Hardware ist aus der Sicht des Anwenders dazu da, analog der gestellten Aufgabe, einwandfrei und mit höchstmöglicher Geschwindigkeit so viele Fehler hervorzubringen, daß in kürzestmöglicher Zeit eine größtmögliche Anzahl von nicht mehr zu korrigierenden Fehlern auftritt.

Für Programmierer und Hersteller hingegen ist Hardware dazu da, analog der gestellten Aufgabe, einwandfrei und mit höchstmöglicher Geschwindigkeit so viele Fehler hervorzubringen, daß in kürzestmöglicher Zeit eine größtmögliche Anzahl von nicht mehr zu korrigierenden Schäden auftritt.

### **Doppelgesetz von der komplexen Hardware:**

1. Komplexe Systeme neigen zu komplexen Fehlern.
2. Einfache Systeme hingegen neigen zu komplexen Fehlern.

### **Erste Erweiterung:**

Neue Systeme produzieren neue Fehler.

### **Zweite Erweiterung:**

Neue Systeme wiederholen ihre neue Fehler.

### **Dritte Erweiterung:**

Alte Systeme produzieren neue und alte Fehler.

### **Schlußfolgerung:**

1. Komplexe Systeme neigen dazu, ihre eigene Funktion zu behindern.
2. Computer funktionieren nur deshalb, damit sie Fehler produzieren können.
3. Systeme neigen zum Wachstum und werden dadurch anmaßend.

### **Die abteilungsspezifische Ableitung:**

Willst Du in einer Abteilung eine permanente Ausrede für eigene Fehler haben, dann rüste sie mit Computern aus (siehe dazu auch Anhang F: Die besten Ausreden).

### **Erster Gegensatz von digitaler und analoger Logik:**

Menschen, die mit Computern arbeiten, verhalten sich nicht so, wie der Computer verlangt, daß sie sich verhalten sollen.

### **Zweiter Gegensatz von digitaler und analoger Logik:**

Menschen werden sich dann und nur dann vernünftig verhalten, wenn alle anderen Möglichkeiten ausgeschöpft sind.

### **Dritter Gegensatz von digitaler und analoger Logik:**

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Systeme, die idiotensicher sind, werden auch von Idioten bedient

## **Gegensatz von digitaler und digitaler Logik:**

Computer, die mit anderen Computern arbeiten, verhalten sich nicht so, wie anderen Computer verlangen, daß sie sich verhalten sollen.

## **Das Assembly-Axiom:**

Alles, was zusammengesetzt wurde, fällt früher oder später auseinander.

## **Verschärfungen:**

1. Alles fällt früher auseinander.
2. Es sucht sich den ungünstigsten Zeitpunkt dazu aus.
3. Alle unbeseelten Gegenstände können sich gerade soweit bewegen, daß sie einem im Weg stehen.

## **Allgemeine Reparaturgesetze:**

1. Wenn Du das kaputte Teil entdeckt hast, fehlt Dir das entsprechende Werkzeug, um es auszubauen.
2. Wenn Du es ausbauen kannst, muß der Computerhändler an den Hersteller einschicken.
3. Wenn der Händler es auf Lager hat, ist der Austausch unnötig.
4. Die Reparaturkosten können bestimmt werden, indem man den Reparaturkostenvoranschlag und den Preis für ein neues Gerät jeweils mit zwei multipliziert und den höheren der beiden Werte nimmt.
5. Ein von einem Händler geliefertes Teil ist in Deinem Computer nicht lauffähig.
6. Das reparierte Teil ist nach dem erneuten Einbau auch nicht mehr lauffähig.
7. Handelt es sich bei einem Reparaturfall um eine Festplatte, dann wirst Du die darauf gespeicherten Daten nie mehr wiedersehen.

## **Ausnahme vom siebten Reparaturgesetz:**

Du wirst lediglich das Inhaltsverzeichnis der Festplatte wiederherstellen können, um zu sehen, was Dir alles verloren gegangen ist.

## **Das Zeitdilemma der Hersteller ("EISA-Syndrom"):**

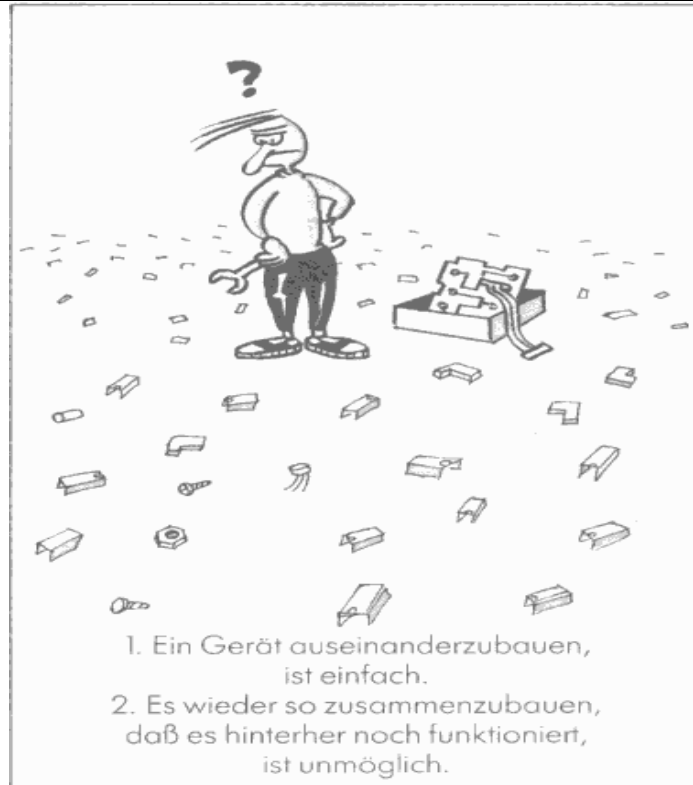
1. Kündigungst Du ein Produkt zu früh an, dann haben es alle längst vergessen, wenn es endlich erscheint.
2. Kündigungst Du es zu spät an, dann ist die Konkurrenz mit dem gleichartigen Produkt schneller.
3. Es ist immer zu früh oder zu spät.

## **Folgen (auch "Jacks Desaster" genannt):**

1. Neuerscheinungen werden immer früher angekündigt, spätestens jedoch sechs Monate vor der Fertigstellung des ersten Prototyps.
2. Produkte erscheinen immer nach dem angekündigten Erscheinungstermin, frühestens jedoch sechs Monate nach der Ankündigung.
3. Bis ein Gerät tatsächlich erhältlich ist, ist es veraltet.
4. Die tatsächliche Markteinführung eines Produktes ist immer nach der Ankündigung der Nachfolgeneration von seiten der Konkurrenz.

## **Logische Konsequenz:**

Irgendwann werden sämtliche Hersteller dazu übergehen, Produkte nur noch anzukündigen, um sich Entwicklungs- und Produktionskosten zu sparen.



25

## Franz' Optionalgesetz:

1. Wenn ein Hersteller sagt, sein Gerät habe diese und jene Option, so heißt das nur, daß der Hersteller bewußt eine wichtige Leiterplatte weggelassen hat, damit man sie extra kaufen muß.
2. Die Erweiterung wird mehr kosten als das eigentliche Gerät.
3. Die Erweiterung wird funktionieren, allerdings nicht bei Dir.
4. Wenn Du Dein Computersystem irgendwann einmal verkaufen willst, wirst die Erweiterung nicht los.

## Udos Bastlererkenntnisse:

1. Ein elektronisches Gerät auseinanderzubauen ist einfach.
2. Es wieder so zusammenzubauen, daß es hinterher noch funktioniert, unmöglich.
3. Dagegen treten hilft nur bei anderen.

## 1.1 Computer

Die Leistungsfähigkeit eines Computers setzt sich zusammen aus seiner Intelligenz - also der Anzahl der fest eingebauten Fehler - , der Geschwindigkeit, mit der er eine größtmögliche Anzahl von Katastrophen produziert, und der Antwortzeit: der Zeit, die der Computer braucht, um sich von Deiner Eingabe zu erholen.

Seit Jahren ist die Computerindustrie mit wachsendem Erfolg dabei, die Leistungsfähigkeit und Fehleranfälligkeit ihrer Systeme zu erhöhen und damit eine immer größere Zahl von Supportingenieuren, Reparaturbetrieben, Installationsfachleuten und Troubleshooting-Experten in Lohn und Brot zu setzen. Diesem wachsendem Markt steht auf der anderen Seite die immer geringer werdende Zahl von Firmen gegenüber, die noch ohne Computer in der Lage sind, positive Bilanzen zu erreichen. Ganz zu schweigen von der noch geringeren Zahl von Firmen, die noch in der Lage sind, trotz Computer positive Bilanzen zu erreichen.

Murphys Gesetz - daß nämlich alles, was schiefgehen kann, auch schiefgeht - wird wie wir festgestellt haben durch den Computer optimiert. Da diese Regel sowohl für das Computersystem an sich als auch für die Beziehung "Computer/Rest der Welt" gilt, sind die Chancen groß, daß es schon in absehbarer Zeit außer Computerherstellern und Computerreparaturbetrieben keine anderen Firmen mehr gibt, bis auch diese aufgrund obligatorischer Computerfehler Konkurs anmelden und eine Gesellschaft übrigbleibt, in der außer Computerfehlern nichts mehr existiert.

# Joachim Graf "Murphys Computergesetze"

Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)

---

## Das endgültige Preis-Gesetz:

Gleichgültig, wie teuer Du ein Computersystem einschätzt. Es wird am Ende stets teurer als erwartet.

**Platts Berechnung des endgültigen Preis-Gesetzes** (auch unter dem Namen "gemeines Erweiterungsfeuer" bekannt):

$$K > (J * (1000 + A/15) + (1,5 * B) + A/20)$$

Dabei sind K die Gesamtkosten nach J Jahren, wenn der Anwender ein jährliches Nettoeinkommen von A besitzt und sich einbildet, sein System würde B Mark kosten.

## Beispiel für Platts Berechnung des endgültigen Preis-Gesetzes:

Ein MS-DOS-Computer, von dem der Verkäufer sagt, er koste 2000 Mark, wird einen Anwender mit einem jährlichen Nettoeinkommen von 36 000 Mark nach einem Jahr also inklusive Software, Schulungen, Computerbüchern und -zeitschriften sowie der dringend erforderlichen Erweiterungen insgesamt mindestens 82000 Mark kosten:

$$(1 * (3400)) + 3000 + 1800$$

## Das Dimensionsmirakel:

Jeder Computer ist zu klein.

### Präzisierungen:

1. Hat er eine ausreichend große Festplatte, dann ist sein Hauptspeicher zu klein.
2. Hat er genügend Hauptspeicher, dann ist die Festplatte zu klein.

## Die MS-DOS-Erweiterung zum Dimensionsmirakel:

Sind Festplatte und Hauptspeicher ausreichend dimensioniert, dann besitzt er ein Betriebssystem, das:

- a) eines von beiden nicht unterstützt.
- b) eine Speicherverteilung benötigt, die das vorhandene Anwendungsprogramm nicht versteht.

## Die physikalische Erweiterung zum Dimensionsmirakel:

1. Auf jeden Fall besitzt Dein Computer einen Steckplatz zu wenig.
2. Das stellst Du erst dann fest, wenn Du Dir eine neue Steckkarte gekauft hast.

## Die BIOS-Erkenntnis:

1. Das BIOS ist immer nur 99prozentig kompatibel.
2. Das von Dir am häufigsten benutzte Programm greift auf das letzte Prozent zu und stürzt unter Datenverlust ab.

## Die Styroporvorhersage:

- a) Je einfacher die Gebrauchsanleitung (beispielsweise: "hier ziehen"), desto schwieriger ist es, alle Einzelteile heil aus dem Verpackungsmaterial zu schälen.
- b) Das wichtigste Teil wird immer mit der Verpackung weggeschmissen.
- c) Du wirst alle Einzelteile der Verpackung nie wieder in den umhüllenden Karton bekommen, um diesen wegräumen zu können.
- d) Sobald der Computer die Verpackung verlassen hat, wird er nie wieder hineinpassen.

## Mengenmäßige Relation der Styroporvorhersage:

Je mehr Verpackungsmaterial zum Einsatz kommt,

- a) um so mehr Teile sind nach dem Auspacken zerstört.
- b) um so mehr Teile sind nicht mehr aufzufinden.

## Die Erkenntnis des Norman Mailer:

Computer sind die intelligentesten Idioten, die es gibt.

## Die dreizehn Bauteil-Gesetze:



# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

1. Ein kostspieliger Chip, geschützt durch eine schnell wirkende Sicherung, wird die Sicherung schützen, indem er zuerst durchbrennt.
2. Bauteile funktionieren solange und wirklich nur solange, bis sie die Eingangskontrolle passiert haben.
3. Nachdem Du bereits alle 16 Halterungsschrauben am Computergehäuse entfernt hast, wirst Du feststellen, daß Du die falsche Abdeckung entfernt hast.
4. Nachdem eine Abdeckung durch 16 Schrauben gesichert wurde, wirst Du feststellen, daß Du ein Anschlußkabel oder eine Steckbrücke vergessen hast.
5. Nachdem der Computer wieder in Betrieb genommen wurde, findest Du die restlichen Teile auf dem Arbeitsplatz.
6. Toleranzen werden sich in eine Richtung zum Zwecke der größten Schwierigkeiten beim Montieren ansammeln.
7. Die Steckplätze oder Chipsockel, in die Du etwas einstecken oder einbauen willst, sind am schlechtesten zugänglich.
8. Für alle komplizierten Montagen brauchst Du drei Hände.
9. Für alle einfachen Montagen brauchst Du vier Hände.
10. Gehäusebohrungen sind einen zehntel Millimeter zu klein.
11. Gehäusebohrungen mit einem richtigen Durchmesser sind an der falschen Stelle.
12. Muttern passen nie auf überzählige Schrauben.
13. Der einzig verfügbare Steckplatz ist für die einzubauende Steckkarte zu kurz.

## **Die Batteriebanalität:**

Der Akku eines Laptops ist eine Minute vor dem nächsten fälligen Speichern leer.

## **Relation zwischen Batterie und Arbeit:**

Je mehr ungespeicherten text man hat, um so schneller ist der Laptop-Akku leer.

## **Das Privatnutzungsaxiom:**

Computer, die tagsüber perfekt funktionieren, versagen nachts, wenn Du ins Büro zurückgehst, um sie für private Geschäfte zu verwenden.

## **Erweiterung:**

Dennoch wird der Host Deine Aktionen mitprotokollieren und Dein Chef ausnahmsweise Zeit finden, diese Protokolle zu lesen.

## **Gesetz vom Kundendienst:**

Computer, die kaputt sind, funktionieren solange der Kundendienst anwesend ist.

## **Jaruks Wartungsgesetz:**

1. Wenn es billiger wäre, einen neuen Computer zu kaufen, besteht die Firma auf Reparatur.
2. Wenn es günstiger wäre, das alte System zu reparieren, besteht die Firma auf dem neuesten Modell.

## **Das Siemenssche Doppelgesetz von der Kompatibilität:**

1. Alle PCs sind kompatibel. Nur sind manche kompatibler als andere.
2. Deiner ist immer um das entscheidende Quäntchen weniger kompatibel.

## **Das Verbesserungsgesetz:**

1. Ein Computermodell heißt "Enhanced", "Advanced" oder "Extended", wenn es dem Hersteller gelungen ist, einen Teil der Fehler zu beheben, die die Vorgängerversion unbrauchbar gemacht hat.
2. Solch ein Computer wird erst dann auf den Markt gebracht, wenn Du die Vorgängerversion gekauft hast.
3. Jeder Hersteller wird in seinen verbesserten Computermodellen genügend Fehler verstecken, damit es noch für mindestens eine wiederum "verbesserte" Version reicht.

## 1.2 Eingabegeräte

Die Computerindustrie bezeichnet Tastatur, Maus, Trackball und Digitalisierbrett gerne als "Benutzerschnittstelle", was unter konsequenter Anwendung von Murphys Computergesetzen eigentlich nichts anderes bedeuten kann, als daß sich der Benutzer geschnitten hat, wenn er meint, er könne mit einem dieser Geräte vernünftig arbeiten.

Während eine Tastatur dazu gemacht wurde, mit unlogischen Cursorbewegungen aus unübersichtlichen Menüs schwerverständliche Befehle auszuwählen, die besser per Maus angewählt werden, dienen Maus und Trackball und Digitalisiertablett dazu mit unlogischen Maus-, Trackball- und Digitalisierstiftbewegungen aus unübersichtlichen Menüs schwerverständliche Befehle auszuwählen die besser per Tastatur angewählt werden und dabei Subjekt und Objekt durch so lange und komplizierte Satzkonstruktionen voneinander trennen, bis es Lektor, Leser und Textverarbeitungsprogramm schwindelig wird.

Wirklich zukunftsorientierte Computerbesitzer warten deshalb auf die ersten funktionsfähigen Spracheingabegeräte. Erst mit diesen wird es möglich sein, die höchste Stufe des Mißverstehens zwischen Computer und Bediener zu erreichen.

### **Die grundlegende Tastaturerkenntnisse:**

1. Deine Tastatur hat immer eine Taste weniger, als Dein bevorzugtes Programm unterstützt.
2. Deine Tastatur hat immer eine Taste zuviel, die kaputt gehen kann und wird.

### **Die grundlegende Mauserkenntnis über die Kompatibilität:**

Kaufst Du Dir eine Mouse-System-kompatible Drei-Tasten-Maus, wirst du Dein Leben lang kein Programm finden, daß die dritte Taste unterstützt. In dem Moment, wo Du auf eine Microsoft-kompatible Zwei-Tasten-Maus umsteigst, wirst Du vorwiegend mit einem Programm arbeiten müssen, das die dritte Taste sinnvoll einsetzt

### **Die grundlegende Digitalisierbretterkenntnis:**

Egal, wie lange oder kurz Du Dein Digitalisierbrett schon hast: Die Folie über den wirklich wichtigen Funktionen ist immer zerkratzt genug, um diese regelmäßig zu verwechseln.

### **Die grundlegende Barcodestifterkenntnis:**

Du wirst mit Deinem Barcodestift alles Unwichtige fehlerfrei einlesen können: Das Etikett Deines Jacketts genauso wie den Preis der Frühstücksflocken. Nur an den Barcodes, die Dein Programm ausgibt, wird er mit der höchstmöglichen Fehlerrate scheitern.

### **Der Trackball/Maus-Unterschied:**

1. Du hast immer so viel Platz für eine Maus auf Deinem Schreibtisch, daß Du die Unannehmlichkeiten eines Trackballeinstzes in Kauf nehmen würdest.
2. Du hast immer weniger Platz auf Deinem Schreibtisch, als Deine Maus braucht.

### **Das Maustreiberaxiom:**

Bei dem einzigen Programm, für das Du Deine Maus brauchen kannst, versagt der Maustreiber.

### **Das Wurstfingerphänomen:**

Tippst Du zwei Tasten gleichzeitig, wird derjenige Buchstabe auf dem Monitor erscheinen, den Du nicht wolltest.

### **Die AT/XT-Relation:**

Wenn Du eine Tastatur besitzt, die in zwei Modi betrieben werden kann, wird sie immer auf den falschen Modus eingestellt sein.

**Gesetz von der hüpfenden Escape-Taste** (auch als "XT-AT-MF2-Divergenz" oder "SHIFT" / '<' und <ESC>-Verblüffung bekannt):

Arbeitest Du abwechselnd an zwei Computern, werden deren Tastaturen die größtmöglichen Layoutabweichungen untereinander aufweisen.

### **Gesetz der Pull-down-Menüs:**

1. Du klickst immer den Menüpunkt daneben an.
2. Stehen zwei falsche Menüpunkte zur Auswahl, wird der Mauszeiger denjenigen aktivieren, dessen Auswirkungen am schwerwiegendsten sind und bei dem die meiste Zeit erforderlich wird, den Ursprungszustand wieder herzustellen.

### **Gregors Seufzer:**

# Joachim Graf "Murphys Computergesetze"

Datei: **Murphy.doc** (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)

---

Es fällt immer die Taste aus, die am häufigsten benutzt wird. Also immer **E** oder die Leertaste, niemals jedoch die **±** oder **œ** Taste.

## Die Return-Ergänzung zu Grgors Seufzer:

Wenn schon die Return-Taste ausfällt, dann ist es die auf der alphanumerischen tastatur, niemals die auf dem Zahlenblock.

### Ausnahme:

Wenn Du überwiegend mit der Maus arbeitest und die Tastatur nur dafür brauchst, um lange Zahlenkolonnen einzutippen, ist es genau umgekehrt.

## Das Schnittstellenphänomen:

Die Maus steckt immer in der Schnittstelle, die der Maustreiber nicht anspricht.

## Gesetz vom "Beenden"- "NEIN"-Doppelschlag:

Wenn Du ausversehen auf die Tastenkombination kommst, die Dein Programm beendet, wirst Du ebenfalls die Taste drücken, die die Frage verneint, ob die gemachten Änderungen abgespeichert werden sollen.

### Beweis:

Die Tasten **B** für "Beenden" und für **V** "Verlassen des Programmes" liegen in unmittelbarer Nachbarschaft der Taste für **n** (für "Speichern - Nein").

## 1.3 Drucker

Ein Drucker zerfällt in den stets verstopften Druckkopf, eine zu kleine Papierzuführung, ein nichtpassendes Kabel, ein leeres Farbband beziehungsweise eine leere Tonerkassette sowie eine den Computer nicht verstehende Elektronik - und dies genau in dem Moment, wenn man ihn anschaltet. Darüber hinaus ist der Drucker die letzte Gelegenheit des Computers, Fehler zu produzieren. Schließlich gilt zur Erfüllung von Murphys Gesetz

### Die Ultima Ratio des Ausdrucks:

1. Wenn alles funktioniert hat, wird der Drucker versagen.
2. Wenn der Drucker nicht versagt, werden die ausgedruckten Ergebnisse falsch sein.
3. Stimmen die Ergebnisse, wirst Du sie nicht entziffern können.
4. Stimmt alles, interessiert sich niemand für Deine Ergebnisse.

Es gibt zwei Kategorien von Druckern, die im nachfolgenden Kapitel gleichberechtigt nebeneinander behandelt werden, da sie keine prinzipiellen Funktionsunterschiede aufweisen: Matrixdrucker, die mit nervenzerfetzender Lautstärke Papier mit unleserlichen Hieroglyphen besudeln, und Laserdrucker, die dies mit nervenzerfetzender Geräuschlosigkeit tun..

### Aldus' Druckgrundsatz:

Ein Ausdruck ist nie vollkommen.

### Die Manuskript-Konkretisierung von Aldus' Druckgrundsatz:

Den Fehler auf einem Ausdruck entdeckst Du erst, wenn Du Dir die Kopie ansiehst und der Brief mit dem Ausdruck bereits im Briefkasten liegt.

### Die Grafik-Konkretisierung von Aldus' Druckgrundsatz:

1. Der Ausdruck paßt nie auf eine Seite.
2. paßt er doch, dann ist die Papierabrißkante so verrutscht, daß der Ausdruck dennoch über die Perforation geht.

### Der mathematische Druckerpapierbeweis:

Bei einem auszudruckendem Text der Länge n Seiten ist die zur Verfügung stehende Restmenge Druckerpapier n-1 Blatt.

### Das Reimer-Syndrom (auch "gemeiner" Etikettenschwindel genannt):

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

Ein Drucker wird so lange anstandslos auf Etiketten drucken, solange Du anwesend bist. In dem Moment, wenn Du den Raum verläßt, bleiben die Etiketten im Papiereinzug kleben.

## **Der Auflösungsreisatz:**

1. "Hohe Auflösung" bedeutet, daß ein Kreis auf dem Bildschirm aussieht wie Stonehenge von oben - nur verschmiert.
2. "Mittlere Auflösung" bedeutet, daß ein Kreis auf dem Bildschirm aussieht wie Stonehenge von oben - nur unscharf.
3. "Niedrige Auflösung" bedeutet, daß ein Kreis auf dem Bildschirm aussieht wie Stonehenge von oben - nur klotziger.

## **Das Epson-Gesetz von den Druckerleistungen:**

1. "Near Letter Quality" bedeutet, daß ein Drucker etwas zu Papier bringt, was so ähnlich wie ein Buchstabe aussieht.
2. "Letter Quality" ist die Schriftart, bei der man nur beim Hinsehen erkennt, daß sie von einem Matrixdrucker stammt.
3. "Draft" ist die Bezeichnung für eine Betriebsart, die:
  - bei schlechtem Farbband das Papier unverändert hinterläßt und
  - bei neuem Farbband etwa drei Millimeter hohe hellgraue Linien aufs Papier bringt.Beides allerdings in höchstmöglicher Geschwindigkeit

## **Petras Druckertreiber-Analyse:**

Gegeben ist ein Anwenderprogramm mit beliebig vielen Disketten, auf denen beliebig viele Druckertreiber sind. Daraus lassen sich folgende Aussagen ableiten:

1. Dein Drucker ist in der Treibersammlung nicht dabei.
2. Ist ein Treiber für Deinen Drucker dabei, dann ist er kaputt.
3. Ist er vorhanden und nicht kaputt, dann arbeitet er nicht mit dem Interface zusammen.
4. Ist er vorhanden, nicht kaputt und arbeitet er mit dem Interface zusammen, dann beherrscht er ausschließlich den altgriechischen Zeichensatz und druckt nur im tibetischen Dreiecksformat von links unten nach rechts oben.

## **Das allgemeine PostScript-Postulat (auch als "DDL-Direktive" oder "PCL-Anweisung" bekannt):**

1. Die benötigte Schrift fehlt.
2. Ist die gewünschte Schrift vorhanden, fehlt der gewünschte Schriftstil.
3. Ist beides vorhanden, druckt der Drucker sie in der falschen Größe an die verkehrte Stelle.
4. In allen anderen Fällen steigt er mit einer Fehlermeldung aus und bricht den Druckvorgang ab.

## **Das spezifische PostScript-Postulat (auch als "Postulat vom kleinsten gemeinsamen Vielfachen" bekannt):**

1. Was der PostscriptStandard ist, definieren Programm und Drucker völlig unterschiedlich.
2. Die einzige Schrift, auf die sich beidbe verständigen können, wird Courier 10 pt sein.

## **Ausnahme des spezifischen PostScript-Postulats:**

Satz 2 gilt nur, wenn Du diese Schrift nicht brauchen kannst. Willst Du Courier 10 pt einsetzen, werden Drucker und Porgramm sich auf die Konvertierung sämtlicher ASCII-Zeichen in die "Symbol"-Schrift einigen.

## **Behmes WYSIWIG-Definition:**

Die französische Übersetzung von WYSIWIG ist »Honni soit qui mal y pense« oder kurz HSQMYP.

## **Rückübersetzung von Behmes WYSIWIG-Definition:**

Die englische Übersetzung von WYSIWIG ist »What You see is what you might get«.

## **Die WYSIWIG-Verschärfung:**

Sobald bei Dir irgendwelche Hoffnungen auftauchen, Du würdest Dich mit Deiner Arbeit einem Ende nähern, lautet die englische Übersetzung von WYSIWYG »What You see is what you never get« oder kurz WYSIWYNG.

## **Charles' Einzugskenntnis:**

1. Kein Einzelblatteinzug kann Einzelblätter, kein Traktor Endlospapier vernünftig einziehen.
2. Darüber hinaus ist der Traktor ausschließlich dazu da, Endlospapier schief einzuziehen.
3. Ein Einzelblatteinzu hingegen ist ausschließlich dazu da, Einzelblätter schief einzuziehen.

## Murphys Widerspruch zu Charles' Einzuserkenntnis:

Der Drucker zieht in dem Moment sauber und exakt ein, wo es nicht erforderlich ist.

## Regel über grafikfähige Drucker:

Ein Kasten, der auf dem Bildschirm folgendermaßen aussieht:



wird auf einem nichtgrafikfähigen Drucker so aussehen:



## Der ASCII/Centronics-Grundsatz von den Druckerstandards:

Das einzige, was zwischen verschiedenen Druckern standardisiert ist, ist das Netzkabel.

### Ableitung für den anglo/britischen Sonderweg:

Selbverständlich gilt dieser Standard nicht für englische Netzstecker und amerikanische Stromstärken.

## 1.4 Massenspeicher

Massenspeicher zerfallen in Disketten-, Band-, Festplatten- oder Winchesterlaufwerke sowie in dem Moment, wo auf ihnen wichtige Daten gespeichert sind.

Der Unterschied zwischen den verschiedenen Laufwerksarten liegt in der Zugriffszeit, worunter in der Computerwelt die Zeitdauer verstanden wird, die ein Laufwerk braucht, um

- Die gesuchte Daten zu finden und
- während dieser Suche alle anderen wichtigen Dateien so zu zerstückeln, daß es weniger Arbeit bedarf, sie neu einzugeben, als sie zu reparieren.

Bei einem Winchesterlaufwerk (also dem Gerät, daß dazu da ist Dateien zu zerschießen) sind Intelligenz, Perfidie und Heimtücke auf einem sogenannten Festplattencontroller untergebracht. Dieser sorgt unter anderem dafür, daß Dateien prinzipiell fünf Minuten vor dem täglichen Backup ruiniert werden oder daß garantiert unwichtige Dateien ((zum Beispiel die README-Datei eines längst gelöschten Programms) von diesem Zerstörungsvorgang auf jeden Fall verschont bleiben.

Band- und Diskettencontroller erfüllen ihre morphologischen Aufgaben, indem sie sicherstellen, daß ausschließlich diejenigen Bänder reißen beziehungsweise diejenigen Disketten nicht mehr lesbar sind, auf denen die einzige aktuelle Datenversion gespeichert ist.

### Die Backup-Prämissen:

- Ein Backup braucht immer eine Diskette mehr, als Du vorrätig hast.
- Ein Backup-Programm wird in dem Moment versagen, wo Du es benötigst.

### Erste Ableitung:

Das Backup-Programm wird bei seinem Versagen mit der zerstörten Dateiversion die einzige noch vorhandene Sicherheitskopie überschreiben.

### Zweite Ableitung:

Wenn Du das Backup zurückspielen willst, wirst du feststellen, daß die einzige Version von RESTORE auf der Platte (und nur dort) war, bevor Du sie formatiert hast.

### Die Disketten-Regel:

Wenn eine Diskette im Laufwerk klemmt, wende Gewalt an. Wenn das Laufwerk dabei kaputtgeht, hätte es sowieso erneuert werden müssen.

### Der Lesefehlerlehrsatz:

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Ein Lesefehler tritt nur bei der Datei auf, die Du noch unbedingt brauchst und von der Du keine Kopie hast.

## **Gesetz von der Geheimhaltung der Dateien:**

Wenn eine Information vertraulich ist, bleibt sie aus Versehen als Datei auf der Diskette. Ein eventuell vorhandener Verschlüsselungsmechanismus wurde vergessen.

## **Erweiterung des Dateigeheimhaltungsgesetzes:**

Sie wird zufällig von demjenigen gefunden und gelesen werden, vor dem sie verborgen wurde.

## **Seagates Formatier-Axiom:**

Du wirst eine Diskette, die sich bereits im Laufwerk befindet auf jeden Fall mit der falschen Aufzeichnungsdichte formatieren.

## **Präzisierungen:**

1. Eine HD-Diskette wirst Du mit 360K Byte formatieren.
2. Eine DD-Diskette wirst Du mit 1,2 MByte formatieren.

## **Folgerung:**

Daß Du eine 360 KByte-Diskette mit 1,2 MByte formatiert hast, wirst Du erst dann bemerken, wenn Du auf diese Diskette die einzige Version einer lebenswichtigen Datei gespeichert hast und sie daraufhin sämtliche Daten verloren hat.

## **Götz'sche Erkenntnis der allgegenwärtigen Unsicherheit:**

Erst, wenn Du - beispielsweise beim Formatieren - **J** auf die Frage des Programmes: "Sind Sie sicher?" geantwortet hast, fällt Dir auf, daß Du Dir ganz und gar nicht sicher bist.

## **Götz'sche verschärfte Erkenntnis der allgegenwärtigen Unsicherheit:**

Wenn Du danach die Disketten kontrollierst, bist Du Dir sicher: Du hast soeben Deine wichtigste Datei gelöscht.

**Grundregel von der Unentrinnbarkeit klebriger Flüssigkeit** (auch der "Cola-und-süßer-Kaffe-Lehrsatz" genannt):

**Du wirst eine Kaffeetasse oder ein Colaglas auf Deinem Schreibtisch nur dann umwerfen, wenn noch Flüssigkeit drin ist.**

1. Die auslaufende Flüssigkeit bahnt sich mit gnadenloser Präzision den Weg zu der wichtigsten Diskette auf dem Schreibtisch.
2. Ihr Weg dorthin führt über die einzigen Ausdrucke und schriftlichen Entwürfe, von denen weder eine Kopie noch eine Datei existiert.
3. Die Flecken werden auf diesen Aufzeichnungen die wichtigsten Stellen für immer unlesbar machen.
4. An der Zieldiskette angekommen, wird die Flüssigkeit in die Schreib-Leseöffnung fließen.
5. Stehen mehrere Disketten zur Auswahl, wird die Flüssigkeit in die Diskette fließen, auf der mehr Dateien sind, deren Inhalt nichtwiederbeschaffbar ist.
6. Die Diskette ist anschließend nicht mehr zu lesen.
7. Sind auf der Diskette trotzdem noch einzelne Dateien zu lesen, dann die unwichtigen (zum Beispiel die Textdatei mit dem Einkaufszettel von letzter Woche).
8. Ein Restaurationsprogramm wird alle Sektoren der Diskette wieder lesbar machen. Mit Ausnahme des alles entscheidenden.

**Der CHKDSK-Grundsatz** (oder "Lehrsatz von der Vergeblichkeit menschlichen Strebens"):

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

- a) Wenn Du ein CHKDSK über Deine Festplatte machst, wird das Programm keinen einzigen unzusammenhängenden Block finden.
- b) Verzichtest Du auf CHKDSK, dann hast Du Tausende davon.

## **Die "Speeddisk"-Erweiterung des CHKDSK-Grundsatzes (bekannt unter dem Namen "Nortons Dilemma"):**

Du wirst die unzusammenhängenden Blöcke erst dann bemerken, wenn Du Speeddisk oder ein anderes Tool startest, das diese Blöcke nicht alleine beseitigen kann, aber die Gelegenheit beim Schopfe packt und Deine Platte ruiniert.

## **Das 3,5-Zoll\_Müll-Axiom:**

Die Plastiktütchen, in denen 3,5-Zoll Disketten verpackt sind, dienen einzig dazu, den Müllberg zu vergrößern.

## **Die Diskettenboxen-Gesetze:**

1. Du bekommst eine Diskette leichter in die Schachtel hinein als wieder heraus.
2. Die Plastiktrenner in der Diskettenschachtel dienen dazu, Dir den Blick auf die gesuchte Diskette zu versperren.
3. Ansonsten klappen die gesuchten Disketten nach vorne.
4. Eine Diskette ist nie in dem Fach, in dem Du sie suchst.
5. Du wirst den Schlüssel nie vermissen. Nur dann, wenn Du die Box versehentlich abschließt.
6. Diskettenboxen lassen sich nicht stapeln.
7. Sie werden Dir so lange das Gegenteil beweisen, bis Du eine von ihnen berührst. Dann werden alle zugleich umfallen und ihren Inhalt auf dem Fußboden verstreuen..

## **Satz von der gesundheitsfördernden Wirkung von Diskettenboxen:**

Wenn Du eine Diskettenbox abschließt, um sie zu transportieren, wird das Schloß aufspringen. Dabei werden sämtliche Disketten herausfallen und sich im größtmöglichen Abstand voneinander im Zimmer verteilen.

## **Die Schreibschutz-Etiketten-Sentenzen:**

1. Schreibschutzetiketten kleben nicht richtig.
2. Schreibschutzetiketten wandern an die unzugänglichste Stelle des Diskettenlaufwerkes.
3. Schreibschutzetiketten gehen nicht wieder ab.
4. Schreibschutzetiketten fehlen dann, wenn Du aus Versehen eine wichtige Diskette formatierst.
5. Schreibschutzetiketten kleben dann auf der richtigen Datendiskette, wenn Du speichern willst und das Anwenderprogramm diesen Fehler nicht abfangen kann.

## **Erweiterung der fünften Schreibschutz-Etiketten-Sentenz:**

In diesem Fall wird das Programm unter Datenverlust abstürzen.

## **Manfreds Amiga-Axiom:**

1. Egal, was für eine Diskette im Laufwerk ist - es ist nie die Systemdiskette.
2. Es ist nur dann eine Systemdiskette, wenn sich auf Deiner bootfähigen Festplatte eine neuere Betriebssystemversion befindet.

## **Der de-Lates-Seufzer:**

Auf einer leeren Diskette kannst Du lange suchen.

## **Murphys Erweiterung des de-Lates-Seufzer:**

Du kannst auch auf einer vollen Diskette lange suchen.

## 2. Software

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Software ist der geglückte Versuch, die Fehler der Computerhardware zu optimieren und durch Weiterentwicklung neue Fehler hinzuzufügen.

Software setzt sich zusammen aus Betriebssystemen sowie Textverarbeitungs-, Dateiverwaltungs-, Grafik-, Tabellenkalkulations- und Telekommunikationsprogrammen, die wiederum allesamt nichts anderes sind als verschiedene Erscheinungsformen von in kompilierte Programmzeilen gegossene Hinterhältigkeit.

Die einzige Aufgabe, die Computersoftware aus der Sicht der Sicht des Anwenders erfüllt - neben der Fähigkeit, die Konten der Hersteller zu füllen - ist, reibungslos, aufgabengemäß und mit hohem Datendurchsatz so zu versagen, daß in kürzestmöglicher Zeit der größtmögliche Schaden angerichtet wird. Was wiederum für Softwarehersteller bedeutet, daß der finanzielle und personelle Aufwand für die Programmentwicklung und -marketing im negativen Verhältnis zu den benötigten Supportausgaben steht. Die einzige Software, die keinen Support benötigt, ist folglich diejenige, für die schon vor Markteinführung eine zu große Supportabteilung eingerichtet wurde.

Aus Anwendersicht stellt sich allerdings

## **Die Morschhauser Frage:**

Support? Was ist das?

Eine Frage übrigens, die bis heute noch von niemandem beantwortet werden konnte.

## **Grundsatz zum Verständnis der Softwareindustrie:**

Alle großen Softwareentwicklungen wurden aufgrund gravierender Programmfehler verwirklicht.

### **Erste Folgerung aus dem Softwaregrundsatz:**

Jedes Programm hat Fehler.

### **Zweite Folgerung aus dem Softwaregrundsatz:**

Jedes Programm hat immer einen Fehler mehr.

### **Dritte Folgerung aus dem Softwaregrundsatz:**

Die Beseitigung eines Fehlers ruft mindestens zwei neue hervor.

### **Persönliche Ableitung aus dem Softwaregrundsatz:**

Wenn die Fehler sich bemerkbar machen, dann bei Dir.

### **Erster logischer Umkehrschluß aus der Existenz von Utilities:**

Die Tatsache, daß es sehr viele kleine nützliche Programme gibt, beweist, daß es auch sehr viele große unnütze Programme gibt.

### **Zweiter logischer Umkehrschluß aus der Existenz von Utilities:**

Die Tatsache, daß es sehr viele große, unnütze Programme gibt, beweist, daß es noch mehr kleinere unnütze Programme gibt, da es je mehr kleine als große Programme gibt.

### **Boris Folgerung:**

Die Effektivität eines Programmes ist umgekehrt proportional zum Verkaufspreis.

### **Die Expertenerkenntnis:**

Expertensysteme erkennt man daran, daß sie auf die Wissensengabe »eine Rose riecht besser als ein Kohlkopf« den Schluß ziehen, Rosen werden auch eine bessere Suppe ergeben.

### **Hans Matthöfers Widerspruch zur Expertenerkenntnis:**

Künstliche Intelligenz ist besser als natürliche Dummheit.

### **Gertis Widerlegung des Matthöfers-Widerspruch:**

Jedes Programm, von dem seine Programmierer behaupten, es besitze die Eigenschaft künstlicher Intelligenz, hat in seinem Datenbestand so viel natürliche Dummheit angereichert, daß es beliebig viele intelligent klingende und zutiefst falsche Antworten geben kann.

### **Heinis Erkenntnisse über Computerspiele:**

1. Es fehlt Dir immer ein Punkt zum neuen Highscore.



# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

2. Hast Du ein Spiel so lange gespielt, bis Dich niemand mehr schlagen kann, wird ein Freund es zum ersten Mal spielen und problemlos Platz eins der Highscoreliste erreichen.
3. Bist Du der Beste, wird sich niemand dafür interessieren.

## **Manfreds Erkenntnisse der Computerspiele:**

1. Das Computerspiel, das Du Dir gekauft hast, ist nicht den Preis wert, den es gekostet hat.
2. Kein Computerspiel ist so gut, wie das Bild auf der Verpackung.
3. Eine gute Kritik in einer Spielzeitschrift besagt nur, daß das Spiel dem Redakteur gefallen hat. Du wirst immer einen anderen Geschmack haben, als er.
4. Das merkst Du aber erst nach dem Kauf.
5. Dein größter Rivale wird ein besseres Spiel haben als Du.

## **Postulat der Multifunktionalität:**

Je weniger Funktionen ein Programm hat, um so perfekter wird es sie ausführen.

## **Folgerungen aus dem Postulat der Multifunktionalität:**

1. Je perfekter ein Programm Funktionen ausführt, um so gründlicher wird es versagen.
2. Jedes andere Programm wird genau so gründlich versagen.

## **Ableitung für integrierte Pakete:**

1. Je mehr Funktionen auf der Rückseite der Verpackung angepriesen werden, um so weniger wird das Programm beherrschen.
2. Wenn draufsteht »für alle Anwendungen«, ist es für niemanden brauchbar.
3. Alle Funktionen, wegen denen Du das Programm gekauft hast, müssen als Spezialmodule extra bezahlt werden.

## **Doppeltes Gesetz von der Erfolgs-/Entwicklungs-Relation:**

1. Je mehr Entwicklungsarbeit in ein Programm gesteckt wird, um so weniger Leute werden es kaufen.
2. Je weniger Entwicklungsarbeit in ein Programm gesteckt wird, um so mehr Leute werden es kaufen und die darin enthaltenen Fehler entdecken.

## **Erste Folgerung aus der Erfolgs-/Entwicklungs-Relation:**

Je mehr Fehler in einem Programm stecken, um so schlechter ist der Support.

## **Zweite Folgerung aus der Erfolgs-/Entwicklungs-Relation:**

Nur das Programm, das niemand braucht, läuft fehlerfrei.

## **Allgemeiner Löschrundsatz:**

Beim Löschen von überflüssigen BAK- und SIK- oder sonstige von Programmen angelegten Sicherheitsdateien verschwindet unter Garantie eine, die man zehn Minuten später dringend braucht.

## **Erweiterung:**

Dieser Satz gilt nicht nur für Sicherheitskopien, sondern für alle Dateien.

## **Das Originalprogramm-Dilemma:**

Wenn Du ganz dringend ein Programm brauchst, ist es ausgeliehen und der Entleiher telefonisch nicht erreichbar.

## **Die Raubkopie-Erweiterung:**

Ist der Entleiher telefonisch erreichbar, ist das Programm eine Raubkopie, die er aus versehen von seiner Festplatte gelöscht hat.

## **die Problem-Prämissen:**

1. Wenn man dringend die Software-Hotline braucht, ist Dein Telefon kaputt oder die Nummer besetzt..
2. Wenn Du das Freizeichen bekommst, ist Wochenende und folglich keiner zu erreichen.
3. Kommt das Freizeichen und es ist ein Werktag, dann ist an diesem Tag die Hotline nicht besetzt.
4. In jedem anderen Fall ist der einzige Mensch, der Dir bei Deiner Frage weiterhelfen kann, krank oder im Urlaub.

## **Allgemeines Dringlichkeitsaxiom:**

# Joachim Graf "Murphys Computergesetze"

Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)

---

Alle Dinge werden unter Druck schlimmer.

## Mathematische Definition des allgemeinen Dringlichkeitsaxiom:

$$D = (P^Z/K) * (C + A + N)$$

Wie man sehen kann, kann die Dringlichkeit D jederzeit beliebig groß werden. P ist die Anzahl der Probleme, Z der Kehrwert der verbleibenden Zeit und K die Kompetenz des mit der Lösung des Softwareproblems Betrauten (es gilt  $0 < K < 1$ ). Die Anzahl der Probleme P ist eine Unbekannte, für die generell gilt  $P > 0$ . Sollte P bekannt sein so gilt  $P = P + 1$ .

Der Wert  $P^Z/K$  ist mit der Summe  $C + N + A$  malzunehmen, wobei C die Anzahl der betroffenen Computer, A die Zahl der verschiedenen Anwenderprogramme in der Firma und N die Zahl der Computer ist, die in mindestens einhundert Meter Umkreis um den mit der Lösung des Softwareproblems Betrauten herum in einem Netzwerk zusammengeschlossen sind.

## Gertis Erweiterung des allgemeinen Dringlichkeitsaxiom:

Alle Dinge werden auch ohne Druck schlimmer.

## Die Großrechnergleichung:

Laufen zwei Prozesse gleichzeitig, wird der unwichtigere dem wichtigen die Rechnerzeit wegnehmen.

## Die grundlegende Viren-Erkenntnis:

Du bekommst eine Computervirus genau dann, wenn Du denkst, Du hättest keinen.

## Die terminierte Viren-Erkenntnis:

Du bekommst eine Computervirus genau dann, wenn Du ihn am wenigsten gebrauchen kannst.

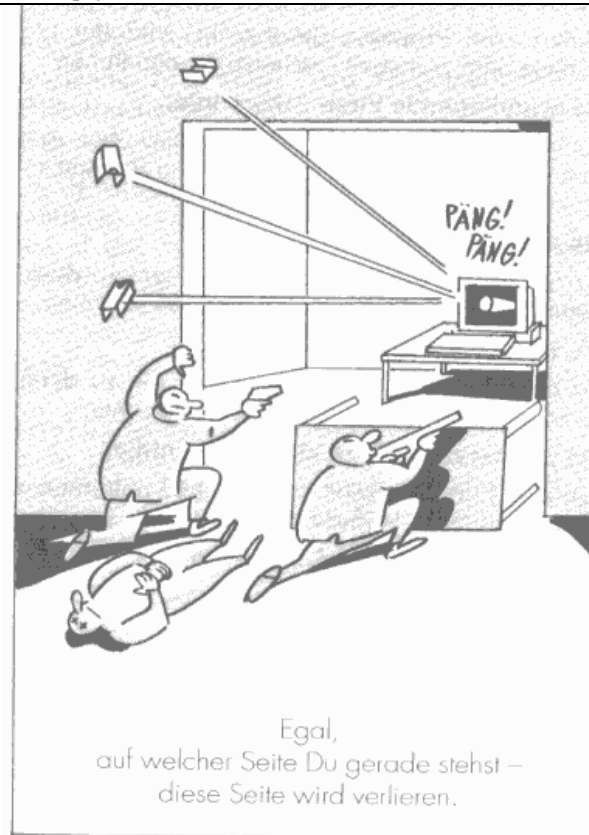
## Die partnerschaftliche Viren-Erkenntnis:

Garantiert ungefährliche und leicht zu entfernende Viren bekommen immer nur die anderen.

## Die geografische Viren-Erkenntnis:

Du bekommst immer die Sorte von Computerviren

- von denen die Experten sagen, daß sie in unserem Land noch nicht aufgetreten sind
- für die es noch kein Virensuchprogramm gibt
- die eine neue Qualität in Heimtücke Destruktivität aufweisen.



### **Die quantifizierte Viren-Erkenntnis:**

Du hast immer einen Virus mehr als Du glaubst.

### **Die qualifizierte Viren-Erkenntnis:**

Der Virus, der Deinen Computer befällt, verseucht nur die Dateien, von denen Du kein Backup hast.

### **Die erweiterte qualifizierte Viren-Erkenntnis:**

Der Virus hat immer einen Arbeitsplatzrechner mehr befallen, als Du glaubst.

### **Die rekursive Viren-Erkenntnis:**

Ein eingesetztes Virensuchprogramm wird ausschließlich diejenigen Viren erkennen und bekämpfen, die Du nicht hast. Den Virus, der Deinen Computer befallen hat, wird es ungeschoren lassen.

### **Erweiterungen zur rekursive Viren-Erkenntnis:**

1. Dies wirst Du erst feststelle, wenn Du das Virenprogramm gekauft hast.
2. Du wirst der einzige sein, dessen frisch erworbenes Virensuchprogramm mit Viren verseucht ist.
3. Dies wirst Du erst feststellen, wenn Du es gestartet hast.

### **Die ergänzte rekursive Viren-Erkenntnis:**

Ein Virensuchprogramm, auf Deinem Computer eingesetzt, wird bei seinem Suchlauf alle Text- und Programmdateien irreparabel zerstören und lediglich den Virus übrig lassen.

### **Folgerung aus der ergänzten rekursive Viren-Erkenntnis:**

Das einzige, was am Ende auf Deinem Computer noch lauffähig ist, wird der Virus sein.

## 2.1 Textverarbeitung

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc* (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)

Textverarbeitungsprogramme sind der geglückte Versuch, einem nichtsahnenden Menschen 1500 Mark für das versprechen abzuknöpfen, er könne mit rund 300 leicht zu merkendne Befehlen Schreibmaschine und Tippex ersetzen und hätte trotz des Programms noch Zeit, sich auf das zu konzetrieren, was er eigentlich schreiben will.

Neben der eigentlichen Arbeit des Schreibens kann der Anwender mit einer Textverarbeitung noch eine Menge nützlicher Dinge tun. Das ist zuallererst und im wesentlichen die Erlernung der Bedienung des Textverarbeitungsprogrammes selber. Hat er daneben noch ein wenig Zeit, kann er gegen die Hilfe des Textverarbeitungsprogrammes zusätzliche Arbeiten erledigen lassen.

Zum Beispiel die *Arbeit mit Textbausteinen* (eine Art Floskelgenerator zum Einfügen unnötiger Textpassagen an Stellen, an die sie nicht passen), die *Verwaltung von Kopf- und Fußnoten* (die automatisch dorthin plaziert werden, wo sie den Autor stören und der Leser sie nicht findet) oder die *Serienbrieffunktion* (mit der man veraltete Adressen in einen überflüssigen Brief automatisch so einfügen kann, daß die Hälfte der Adressen falsch ausgedruckt wird und so die Rücklaufquote durch die Post mit dem Vermerk »unbekannt« über neunzig Prozent liegt. Die anderen zehn Prozent werden von den Empfängern weggeschmissen.)

## **Theas Absturz/Speicher-Relation:**

Ein Computer stürzt nur ab, wenn der Text lange nicht gespeichert wurde.

## **Unverrückbare Gesetze der Textverarbeitung** (auch »WordStar-Axiome« genannt):

1. Wenn Du ein Wort löschen willst, verschwindet garantiert die gesamte Zeile.
2. Wenn Du eine Zeile löschen willst, verschwindet garantiert ein ganzer Absatz.
3. Wenn Du einen Absatz löschen willst, verschwindet der ganze Text.
4. Wenn Du den ganzenText löschen willst, passiert gar nichts.



## **Schlußfolgerung aus dem WordStar-Axiomen:**

Keiner dieser Vorgänge ist rückholbar.

**Frankes Gesetz vom Word-Wechselbad** (auch unter dem Namen »Microsofts AErger-Gesetz« bekannt):

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Wenn Du mit der Funktion »Suche-Ersetzen« arbeitest, wirst Du dann vergessen, die Groß/Kleinschreibung einzuschalten, wenn der zu suchende Text aus einem und der zu ersetzende aus zwei Buchstaben besteht. Ein »Ärger« wird folglich immer ein »Aerger«, nie in ein »Aerger« konvertiert.

## **Michaels Einsicht in das Wesen der Tastaturbelegung:**

Wenn Du Dich an die Tastaturbelegung eines Programmes gewöhnt hast, erscheint eine neue Version, in der alles komplett umgestellt ist.

## **Das Gesetz der hohen Zahl** (auch als »WordStar-Rivalitätsgesetz« oder »Fünf-Null-Syndrom« bezeichnet):

1. Wenn eine neue Version einer Textverarbeitung auf den Markt erscheint, wird sie die höchste aller existierenden Programme aufweisen.
2. Zwei Wochen später werden auch alle anderen Programme diese Versionsnummer tragen.

## **Liesels Erkenntnis:**

Diese Versionsnummern sagen nichts über die Zahl der möglichen Verbesserungen gegenüber den alten Versionen aus.

## **Marcel's Verschärfung von Liesels Erkenntnis:**

1. Je teurer der Update auf die neue Version, um so geringer der Nutzen.
2. Je höher die Versionsnummer, um so größer der Speicherverbrauch.
3. Das Feature, auf das Du seit Jahren wartest, wird auch in der neuen Version nicht implementiert sein.

## **Axiom vom hinterhältigen Druckfehler:**

Du findest den entscheidenden Fehler in deinem Text erst, wenn Du den Text ausgedruckt und die Datei auf der Diskette gelöscht hast.

## **Die unveränderlichen Regeln beim Textformatieren in Desktop-Publishing Programmen:**

1. Wenn Du einen Text positionieren willst, paßt die letzte Zeile nicht mehr auf die Seite.
2. Formatierst Du den Text um die kleinstmögliche Einheit kleiner, ist der Text zu kurz.

## **Das Klick-and-Move-DTP-Gesetz** (auch als »Ja-mach-nur-einen-Plan«-Dilemma bekannt):

- a) Klickst Du auf einer layouteten Seite auf einen Kasten, um ihn zu verschieben, wirst Du den ihn umgebenden Text bewegen.
- b) Klickst du auf den Text, um ihn zu verschieben, wirst du lediglich den Kasten bewegen.
- c) Willst Du beides verschieben, bewegen sich die Hilfslinien.
- d) Keiner dieser Vorgänge ist rückholbar, die »Rücknahme«-Funktion wird nicht funktionieren.
- e) Ist einer der Vorgänge rückholbar, so wird sich irgend etwas um einen Punkt verschieben und das gesamte Layout ruinieren.

## 2.2 Dateiverwaltung

Dateiverwaltungsprogramme und Datenbanken sind der geglückte Versuch, einem nichtsahnenden Menschen 1500 Mark für das Versprechen abzuknöpfen, er könne mit rund 300 leicht zu merkenden Befehlen seinen Zettelkasten ersetzen und hätte trotz des Programms noch Zeit, sich an das zu erinnern, was er ursprünglich mit einem Dateiverwaltungsprogramm wollte.

Dateiverwaltungsprogramme und Datenbanken sind, wie der Name sagt, in erster Linie Verwaltungsprogramme, mit denen man Dinge auf die lange Bank schieben kann. Wie bei Bürokratien üblich, steht das, was man vorne hineinsteckt (also die Datensätze), das, was hinten herauskommt (also die sortierten Datensätze, Listen oder ähnliches), und die Zeit dazwischen in keinem Verhältnis.

Obwohl in allgemeiner Form prinzipiell für alle programmarten zutreffend, gilt für Dateiverwaltungsprogramme ganz besonders das

## **Gesetz des Dauerverhältnisses:**

Egal wie die Verhältnisse in Deiner Datenbank aussehen oder sich verändern - alles dauert immer länger als nach Deiner pessimistischsten Prognose.

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Kein Programmstart entwickelt schließlich so viel Ausflüchte (Programmiererslang: »Fehlermeldungen«), um das, was man eingegeben hat, nicht wieder herausrücken zu müssen, wie datenbanken und Dateiverwaltungsprogramme.

## **Das Zorro-Gesetz:**

1. Egal, wie groß Du eine Maske anlegst. Sie hat immer zu wenig Felder.
2. Wenn dennoch genug Felder definiert sind, sind diese zu kurz.

## **Andis Erkenntnis von der diebischen Dateiverwaltung:**

Wenn Du Dich darüber ärgerst, daß Du in Deinem Dateiverwaltungsprogramm die Feldlängen nicht verlängern kannst: sei beruhigt. Dein Programm wird auf jeden Fall in der Lage sein, die Feldlängen willkürlich zu verkürzen, und dabei die nun überflüssigen Daten wegwerfen.

## **Bobs Datensatzregeln:**

- a) ein Dateiverwaltungsprogramm zerstört nur den Datensatz, den Du dringend benötigst.
- b) Wenn kein Backup existiert, werden alle Datensätze zerstört.
- c) Existiert ein Backup, ist der Datensatz auch dort verschwunden.

## **Gesetz von der Minimierung der Informationsmenge:**

1. Wenn Du eine datenbank mit n Datensätzen anlegen willst, kann das einzige verfügbare Dateiverwaltungsprogramm bestenfalls n-1 Datensätze verwalten.
2. Das wird Dir von dem Programm allerdings erst dann mitgeteilt, wenn Du n-2 Datensätze bereits eingegeben hast.

## **Ashton Tates Gesetz vom Zeitverhalten in Datenbanken:**

Egal, wie Deine Datenbankstruktur aussieht: Das von Dir benutzte Datenbankprogramm wird beim Sortieren oder Suchen das schlechteste Zeitverhalten an den Tag legen. Liegen beispielsweise Deine Datensätze völlig unsortiert vor, sucht Dein Programm lediglich die sortierten datensätze in einer annehmbaren Zeit heraus.

## **Tonis Erweiterungen des Adreß-Axioms:**

1. Den Zettel, auf dem die einzugebende Adresse steht, wirst Du nicht mehr finden.
2. Der erste Kollege, den Du nach der Adresse fragst, wird sie Dir hohnlachend auswendig sagen.
3. Sekunden später wirst Du den zettel mit der Adresse finden.
4. Ist eine Adresse doch in einer Datenbank gespeichert, hat sie sich inzwischen geändert oder die telefonnummer stimmt nicht mehr.

## 2.3 Grafikprogramme

Grafikprogramme sind der geglückte Versuch, einem nichtsahnenden Menschen 1500 Mark für das Versprechen abzuknöpfen, er könne mit rund 300 leicht zu merkenden Befehlen Bleistift, Lineal und Radiergummi ersetzen und hätte trotz des Programms noch Zeit, sich an das zu erinnern, was er ursprünglich eigentlich zeichnen wollte.

In der Kategorie unbrauchbare Grafikprogramme fallen für die professionelle Arbeiten unbrauchbare Malprogramme, für künstlerische Arbeiten unbrauchbare Zeichenprogramme sowie für jeden klar denkenden Menschen unbrauchbare CAD-Programme.

Malprogramme geben Farben auf dem Bildschirm richtig, auf dem Drucker falsch und auf dem Schwarzweiß-Drucker gar nicht wieder. Zeichen- und CAD-Programme hingegen geben einfarbige Linien auf Farb- und Schwarzweiß-Drucker gar nicht wieder. Alle drei Programmarten geben jedoch auf jeden Fall das, was auf dem Bildschirm schwarz beziehungsweise weiß war auf dem Ausgabegerät schwarz beziehungsweise weiß wieder.

Seite 74

## C. Opfer

### 1. Programmierer

# Joachim Graf "Murphys Computergesetze"

*Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)*

---

Programmierer sind (trotz anderslautender Gerüchte): Menschen, die zu nachtschlafender Zeit mit völlig untauglichen Entwicklungspaketen für nicht zusammenpassende Konglomerate fehlerverseuchter Hardware versuchen, im Auftrag von unfähigen Bedienern deren einander widersprechende Anforderungen in Programme umzusetzen, die am Schluß niemand verwendet.

Programmierer zerfallen in zwei Kategorien. Die eine Sorte versagt bei dem Versuch, für viel zu wenig Geld mit viel zu viel Aufwand die logischen Irrtümer von Programmiersprachen, die Fehler von Compilern und die in Silizium gegossenen Ungereimtheiten der Hardwareentwickler so gegeneinander auszuspielen, daß das Computersystem am Schluß wenigstens hin und wieder das tut, was man von ihm erwartet. Die andere Sorte tut dies ganz umsonst.

Generell ist das Denken eines Programmierers logisch (»IF 1=2 CALL mainprogramm«), stets strukturiert (»ON hunger GOSUB Aldi ELSE RETURN«) und von keinerlei Vorurteilen beeinträchtigt. Obwohl es einzelne Vertreter dieses Berufszweiges geben soll, die dem Vorurteil nachhängen, daß ein Computer dazu geschaffen wurde, dem Menschen zu dienen. Anstatt umgekehrt.

Oder, wie es der berühmte angloamerikanische Schriftsteller Wilhelm D. Base Shakespeare sagte:

## **2b .or .not 2b.**

Obwohl es schwierig ist, Murphys Computergesetze aus Sicht des Programmierers zu schildern (schließlich ist ein Programmierer das im Grunde völlig überflüssige Glied der Kette Marketingabteilung - Werbeabteilung - Programmierer - Vertriebsabteilung - Anwender - Supportabteilung - Updateabteilung), soll auf den folgenden Seiten der Versuch dazu unternommen werden. Auch wenn sich Softwarehäuser und Anwender seit Jahren darüber einig sind, daß ihr Leben ohne die überbezahlten Programmierer und deren Einwände über die Machbarkeit bestimmter Programmanforderungen sehr viel leichter wäre.

Namhafte Hersteller sind deswegen mit wachsenden Erfolg seit geraumer Zeit dazu übergegangen, ihre Software mittels »CASE« (»Coputer Aided Software Engineeering«) direkt entwickeln zu lassen, weil letztendlich nur ein Computer Programme so schreiben kann, daß andere Computer sie auch in der richtigen Art mißverstehen können.

## **Lükes Grundlage der Programmierung:**

Es wird nicht funktionieren.

### **erste Ableitung:**

Funktioniert es doch, dann hat es jemand anderes geschrieben.

### **zweite Ableitung:**

Fluchen ist die einzige Sprache, die Programmierer perfekt beherrschen.

### **Schlußfolgerung:**

Ein Computer wird das tun, was Du programmierst - nicht, was Du willst.

## **Doppelregel für Hobbyprogrammierer:**

1. Führst Du ein selbstprogrammiertes Programm vor, dann stößt Du beim ersten Mal auf einen offensichtlichen Fehler.
2. Gravierende Fehler sind von Dir nicht reproduzierbar. Sie werden allerdings von jedem bemerkt, der außer Dir Dein Programm startet.

## **Axels Erkenntnis vom Debugging:**

Nichts verbessert ein Programm so sehr wie das Fehlen von Kontrollroutinen.

## **Axels erweiterte Erkenntnis:**

Wenn Debugging der Vorgang ist, Fehler aus einem Programm auszubauen, dann ist Programmieren der Vorgang, Fehler einzubauen.

## **Axels Folgerung:**

Wenn Du nicht weißt, was Du tust - mach es elegant.

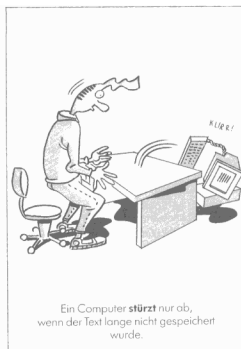
# Joachim Graf "Murphys Computergesetze"

Datei: **Murphy.doc** (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)

Seite 107



Die Fertigstellung eines Programms  
braucht immer doppelt so lang,  
wie geplant.



Ein Computer **stürzt** nur ob,  
wenn der Text lange nicht gespeichert  
wurde.



# Joachim Graf "Murphys Computergesetze"

Datei: Murphy.doc (Erstelldatum: 23.04.93 13:39 Uhr Letzte Aktualisierung: 30.06.97 21:50)



115