

Markt&Technik

Joachim Graf

Murphys Windows-Gesetze o d e r

Wie das durch den Computer
optimierte Gesetz,
daß alles,
was schiefgehen kann,
auch schiefgeht,
von grafischen
Benutzeroberflächen
zur unnachahmlichen
Vollkommenheit
ausgebaut wird.



Murphys Windows-Gesetze

Murphys Windows-Gesetze

oder:

**Wie das durch den Computer
optimierte Gesetz, daß alles, was
schiefgehen kann, auch schiefgeht,
von grafischen Benutzeroberflächen
zur ungeahnten Vollkommenheit
ausgebaut wird**

*von Joachim Graf
mit Illustrationen von Rolf Boyke*

Kompatibel zu Windows 3.1

Graf, Joachim:

Murphys Windows-Gesetze oder wie das durch den Computer optimierte Gesetz, daß alles, was schiefgehen kann, auch schiefgeht, von grafischen Benutzeroberflächen zur unnachahmlichen Vollkommenheit ausgebaut wird / Joachim Graf. – Haar bei München : Markt-und-Technik-Verl., 1992
ISBN 3-87791-364-4

Die Informationen im vorliegenden Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungs vorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.
Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

95 94 93 92

Bestell-Nr. 91364

© 1992 by Markt&Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, D-8013 Haar bei München/Germany
Alle Rechte vorbehalten

Einbandgestaltung: Grafikdesign Heinz Rauner
Dieses Buch wurde mit Desktop-Publishing-Programmen erstellt,
auf der Linotronic 300 belichtet und
auf chlorfrei gebleichtem Papier gedruckt
Druck und Verarbeitung: Schoder, Gersthofen
Printed in Germany

Inhaltsverzeichnis

Danksagung	7
Erklärung	7
Widmung	7
Vorwort	8
A. Murphyologische Grundlagen	11
B. Die Potenzierung der in Programmzeilen gegossenen Heimtücke: von DOS zu Windows	20
C. Wie alles, was schiefgeht, in Vierkanal-Stereo schiefeht: Die Multimedia-Extensions von Windows	30
D. Der murphyologische Update: Windows 3.1	34
E. Das Alphabet der Pannen: Windows- Anwendungen von A bis Z	47
F. Die Optimierung der in Programmzeilen gegossenen, in Silizium geätzten Heimtücke: Windows und die Hardware	99
G. Die zwei Seiten der Medaille: Programmierer, Anwender und Windows als feixender Rand, auf dem sie stehenbleibt	120
H. Zitate	136

I. Erkenntnisse	141
J. Definitionen	150
K. Index	159
Schlußergebnis	160

Danksagung

Ausdrücklich bedanken möchte ich mich bei den Erfindern der grafischen Benutzeroberfläche, den Ingenieuren der Palo Alto Research Center von Xerox, sowie bei allen, die ihre Ideen abgekupfert haben. Ohne sie hätte es dieses Buch nie gegeben.

Erklärung

Jegliche Ähnlichkeiten mit lebenden oder toten Personen, bestehenden oder bankrotten Firmen, verfügbaren und angekündigten Produkten sowie anderen Plagen der Computeranwender sind vom Autor zwar nicht ausdrücklich beabsichtigt, aufgrund der Materie jedoch vermutlich unvermeidlich.

Widmung

Allen Menschen, die trotz anderer Erfahrungen nach wie vor der unverbrüchlichen Hoffnung sind, daß grafische Benutzeroberflächen die Arbeit am Computer vereinfachen.

Vorwort

In seinem Buch »Murphys Computergesetze« hat der Autor zum ersten Mal in der Geschichte der elektronischen Datenverarbeitung den Beweis angetreten, daß kein Forschungsergebnis mehr zum Verständnis unserer modernen Industrie- und Informationsgesellschaft beigetragen hat, wie

Murphys Gesetz:

Wenn etwas schiefgehen kann, dann wird es auch schiefgehen.

Ausgehend von der Tatsache, daß Computer zwar weder intelligent noch kreativ sind, jedoch Heimtücke, Hinterhältigkeit und Verschlagenheit bei ihnen bereits optimal entwickelt sind, kann jeder Hard- und Softwareentwickler, jeder Programmierer und jeder Anwender – kurz: jedes Computeropfer – Murphys Gesetz erweitern durch die

Erste digitale Ableitung:

Murphys Gesetz wird durch Computer optimiert.

In den Zeiten fortschrittlicher Multitasking-Betriebssysteme und Multi-User-Installationen sind moderne Computer heute bereits in der Lage, mehr als eine Sache gleichzeitig zu machen (beispielsweise unter Datenverlust abzustürzen und gleichzeitig die einzig existierende Sicherheitskopie auf sämtlichen ange-

schlossenen Festplatten und Diskettenlaufwerken zu löschen), ergibt sich daraus bekanntermaßen die

Zweite digitale Ableitung:

Alles geht auf einmal schief.

Die Flexibilität der in Silizium und Programmzeilen gegossenen Heimtücke erschließt sich darüberhinaus durch die

Dritte digitale Ableitung:

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

Doch mit dem massiven Siegeszug der PCs war der Leidensweg von Anwendern, Programmierern, Entwicklern und anderen ähnlich armen Schweinen längst nicht beendet. Grafische Benutzeroberflächen begannen sich weltweit über die Monitore auszubreiten wie Schimmel über feuchte Kellerwände. Ausgerüstet mit Maus und Beruhigungspillen leidet die weltweite Computergemeinde nun zusätzlich unter der

Vierten digitalen Ableitung (auch »Erste Ableitung der ersten digitalen Ableitung«) von Murphys Gesetz:

Das Gesetz, daß alles, was schiefgehen kann, auch schiefgeht, wird durch den Computer optimiert und von grafischen Benutzeroberflächen zur ungeahnten Vollkommenheit ausgebaut.

Bei dem hier vorliegenden zweiten Band über murphyologische Computerforschung konnte der Autor wiederum auf seine jahrelange Beschäftigung mit

grafischen Benutzeroberflächen als Anwender, Programmierer und EDV-Fachjournalist – also: stets als Opfer – zurückgreifen.

Für den computertechnisch erfahrenen Leser sollten fachliche Irrtümer und Rechtschreibfehler, die er in diesem Buch entdeckt, nicht verwundern. Denn sie sind selbstverständlich eine Bestätigung für die im vorliegenden Buch vertretenen Thesen. Wurde das Buch doch mit einem unter Windows arbeitenden Textverarbeitungsprogramm geschrieben und mit einem windowsbasierenden automatischen Rechtschreibprüfprogramm korrigiert.

A. Murphyologische Grundlagen

Auch wenn sich das vorliegende Grundlagenwerk schwerpunktmäßig mit dem von Microsoft zu verantwortenden Heimtückenoptimierer »Windows« beschäftigt, so muß doch festgehalten werden, daß es für den Computerbesitzer keine Möglichkeit gibt, potentiellen Pannen, dräuenden Datenverlusten und anstehenden Abstürzen zu entkommen, indem man beispielsweise auf System 7, Open Look, Motif, New Wave oder Comfodesk ausweicht. Schließlich gilt überall in der EDV-Welt das

Gesetz von der plattformübergreifenden Panne:

Der einzige Unterschied zwischen verschiedenen Computersystemen besteht darin, daß der Anwender unterschiedliche Dinge tun muß, um dieselben Pannen zu produzieren.

Es ist also egal, ob der Anwender die Löschen-Taste drücken oder ein Dateisymbol mit der Maus auf das Papierkorb- beziehungsweise auf das Shredder-Icon ziehen muß, wenn er eine garantiert nicht mehr benötigte Datei löschen will. Er wird in jedem Fall feststellen (und dies selbstverständlich zu spät), daß sein System statt dessen die einzige Kopie eines dringend benötigten Textes unwiederbringlich vernichtet hat. Jedoch gilt zur Beruhigung aller die

Erste Erweiterung des Gesetzes von der plattformübergreifenden Panne:

Wenn Du bei verschiedenen Computersystemen dasselbe tust, wirst Du unterschiedliche Pannen produzieren.

Zweite Erweiterung des Gesetzes von der plattformübergreifenden Panne:

Wenn Du bei verschiedenen Computersystemen dasselbe tust, um jemand anderem zu beweisen, daß unterschiedliche Pannen dabei herauskommen, wird alles so lange völlig normal und fehlerfrei funktionieren, solange dieser anwesend ist.

Michaels Erweiterung:

... aber keine Sekunde länger.

Unterschiede bei den grafischen Benutzeroberflächen verschiedener Computersysteme bestehen, dieser Erkenntnis folgend, einzig darin, daß kleine, leistungsschwache Computersysteme große nichtbehebbarer Fehler produzieren, während leistungsstarke Computersysteme hingegen große nichtbehebbarer Fehler produzieren. Tröstlich immerhin, daß zwischen den grafischen Benutzeroberflächen verschiedener Computersysteme dennoch wegweisende Unterschiede bestehen:

Ausnahme von dem Gesetz der plattformübergreifenden Panne:

Fehlerbehandlungsroutinen verschiedener Computersysteme sind so unterschiedlich, daß Du Erfahrungen, die Du auf dem ersten System gemacht hast, nicht auf dem zweiten verwenden kannst.

Erste Erweiterung:

Du wirst sie auch nicht auf dem ersten verwenden können.

Zweite Erweiterung:

Eine untrügliche Fehlerbekämpfungsroutine auf dem ersten Computersystem ist auf einem zweiten System die einzige todsichere Methode, Deine Daten endgültig und unwiederbringlich zu vernichten.

Dritte Erweiterung:

Auf dem ersten auch.

Jegliche Art von Computern tragen bekanntlich als als Objekte getarnte Subjekte unseres Alltags dazu bei, das Gesetz von Murphy immer wieder auf's neue zu bestätigen. Mit der Einführung von grafischen Benutzeroberflächen wurden nicht nur alle nur denkbaren pannen-trächtigen Komponenten, die in einem Computersystem vereint sind – wie Zentraleinheit, Monitor, Massenspeicher, Erweiterungskarten, Tastatur, Anwender und sonstige Peripheriegeräten – um eine weitere Komponente ergänzt, die in trauter Einigkeit mit allen anderen daran arbeitet, möglichst unverträglich zueinander zu sein. Nein, darüberhinaus ist die

Generelle Erkenntnis über grafische Benutzeroberflächen:

Grafische Benutzeroberflächen sind der geglückte Versuch, die Fehler von Computerhardware, Peripheriegeräten, Entwicklungsumgebungen, Programmiersprachen, Anwendungsprogrammen und Tools zu optimieren und durch Weiterentwicklung neue und verheerendere Fehler hinzuzufügen.

So ergänzen grafische Benutzeroberflächen das physikalische Gesetz der Entropie (nach dem bekannterweise alle Teilchen des Universums nicht nur bestrebt sind, sich in größtmöglicher Unordnung anzuordnen, sondern Dir darüberhinaus auf dem Weg dorthin mindestens einmal auf die Zehen fallen) um eine neue Erkenntnis. Nämlich, daß schmerzende Zehen auch dann weh tun, wenn das sie verursachende Chaos interaktiv und benutzerfreundlich ist.

So gelten in der Welt der grafischen Benutzeroberflächen zwar prinzipiell dieselben Gesetze wie auch sonst bei Computern, so beispielsweise die

Erste elektronische Anwendung von Murphys Gesetz:

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

Doch je mehr bunte Dinge auf dem Bildschirm zu sehen sind, umso mehr Chancen hat eine Panne, sich zu verstecken. Daraus folgt die

Erste interaktive Anwendung von Murphys Gesetz:

Hinter jedem geöffneten Fenster Deiner Benutzeroberfläche wartet eine Computer-Panne auf den ungünstigsten Zeitpunkt, um dann erbarmungslos zuzuschlagen.

Bei genauerer Betrachtung kann die erste interaktive Anwendung von Murphys Gesetz problemlos erweitert werden durch die

Zweite interaktive Anwendung von Murphys Gesetz:

Auch hinter jedem geschlossenen Fenster Deiner Benutzeroberfläche wartet eine Computer-Panne auf den ungünstigsten Zeitpunkt, um dann erbarmungslos zuzuschlagen.

sowie die

Erweiterung der zweiten interaktiven Anwendung von Murphys Gesetz:

Dieses Gesetz gilt selbstverständlich auch für Mauszeiger, Pop-up-, Pull-down- und andere Menüs, sowie für alles, was sonst auf dem Bildschirm zu sehen oder nicht zu sehen ist.

Die Menü-Konkretisierung:

1. Jedes Pop-up-, Pull-down- oder sonstige Auswahlmenü ist zu klein.
2. Besitzt ein Auswahlmenü »n« Felder, so steht der gewünschte Menüpunkt an Position »n+1«.
3. Du mußt folglich stets auf den Rollbalken klicken, bevor Du den gewünschten Menüpunkt anklicken kannst.

Die Monitor-Konkretisierung:

1. Jeder Monitor ist zu klein.
2. Mit einem größeren Monitor steigt die Zahl der verborgenen Pannen.
3. Mit einem größeren Monitor steigt auch die Zahl der offensichtlichen Pannen.

Da in der Welt grafischer Benutzeroberflächen Pannen bekanntermaßen nicht enden, sondern einander überlappend ineinander übergehen, hat Microsoft seine Windows-Fehlerverwaltungs- und Optimierungsroutine in Windows 3.1 erweitert: Statt dem unter Windows 3.0 häufig auftretenden »Nicht behebbarer Fehler im Anwendungsprogramm« (englisch laut »bay area computer currents« vom 9.9.91: »UAE: unreasonable application error«) meldet Windows 3.1 nun bekanntlich das leichtverständliche »Dieses Programm hat die Systemintegrität verletzt«. Während bei Windows 3.0 danach sämtliche Daten verloren sind, sind bei Windows 3.1 danach sämtliche Daten verloren:

Doktor-Watson-Dreisatz (oder »Verallgemeinerung der interaktiven Anwendungen von Murphys Gesetz«):

1. Du kannst niemals einer Panne entgehen, indem Du sie zu vermeiden versuchst oder probierst, ihr auf die Spur zu kommen.
2. Im besten Fall wird ein installiertes Wächterprogramm versagen und mit seinem Versagen die bereits vorhandene Panne unterstützen.

3. Der einzige Unterschied beim Einsatz eines Wächterprogramms ist der, daß Du vor dem Absturz dreimal »OK« anklicken mußt.

Da sich bekanntlicherweise niemand so viele Pannen vorstellen kann, wie im Inneren eines Computers passieren, gilt

Die Pannengleichung bei grafischen Benutzeroberflächen:

Bei Arbeiten mit grafischen Benutzeroberflächen sei »N« die Zahl der entdeckten und »M« die Zahl der noch unentdeckten, aber vermuteten Fehler. Für die Gesamtsumme »P« der tatsächlichen Pannen gilt dann

$$P = N * M + 1$$

In Deinem Computersystem verbirgt sich also immer mindestens ein Fehler mehr, als in Deinen schlimmsten Alpträumen vermutet. Wie überall im Bereich der exakten mathematisch geprägten Naturwissenschaft nimmt »=« dabei die Bedeutung von »'höchstens', wenn es für Dich vorteilhaft und 'mindestens', wenn es für Dich schlecht aussieht, mein Junge« an. (Wobei der Autor an dieser Stelle nicht auf die geschlechtsspezifischen Implikationen der Definition eingehen kann. Unter murphyologischen Gesichtspunkten ist es nämlich nicht so, daß Frauen unbegabter für Mathematik sind. Tatsächlich ist es so, daß dies auch für Männer gilt und daß jegliche Hard- und Software beides stets gnadenlos ausnützt, wobei sie dabei Satz, Inhalt, Subjekt und Objekt durch so lange und komplizierte Satzkonstruk-

tionen durcheinanderwürfelt, bis es Lektor, Leser und Autor schwindelig wird. Was wiederum beweist: Auch für murphyologische Fachwerke gilt das

Graf'sche Klammergesetz (auch »Fadenphilosophie« genannt):

Eine zur Verdeutlichung eingefügte Klammer in einer Erklärung führt bei Leser wie Autor zum endgültigen Verlust des Roten Fadens.

Klammer zu, Punkt.



Speicher ist das, was grafische Benutzeroberflächen zum Absturz bringt, wenn man vergißt, sich genügend viel davon anzuschaffen.

B. Die Potenzierung der in Programmzeilen gegossenen Heimtücke: von DOS zu Windows

Die von Wilhelm Tore in die Welt gesetzte These, mit Windows würde der Traum eines jeden Computeranwenders – »Information at your fingertips« (murphyologisch ins Deutsche übersetzt etwa: »wasserfeste Druckerfarbe an den Händen«) – verwirklicht, führt in die Irre:

Bills grundlegender Irrtum:

Ersetze den kryptischen, langsamen und komplizierten Kommandozeileneditor von DOS (bei dem man beispielsweise zum Kopieren von Dateien auf eine Diskette zehn Tastendrucke braucht: »COPY . A: Return«) durch ein mit der Maus gesteuertes grafisches Benutzerinterface (Befehlsfolge hier: Doppelklick auf das Icon des Programmanagers, Doppelklick auf das Icon des Dateimanagers, der Reihe nach Klicken auf die einzelnen Verzeichnisicons, der Reihe nach Klicken auf die gewünschten Dateien, Klicken auf Menü »DATEI«, Klicken auf »KOPIEREN«, Klicken auf das Icon des Diskettenicons, Klicken auf »OK«, Klicken auf das Menü »DATEI«, Klicken auf »BEENDEN«). Die Benutzer werden dich dafür lieben.

Verallgemeinerung von Bills grundlegendem Irrtum:

Ein Computer wird durch grafische Benutzeroberflächen nicht einfacher zu bedienen, es wird nur immer einfacher, Fehler zu machen.

Jedoch ist der Fortschritt nicht mehr aufzuhalten. Konservative Kleingeister, die sich den Segnungen grafischer Benutzeroberflächen verschließen wollen, bekommen die volle murphyologische Wucht der freien Marktwirtschaft zu spüren. Im Gegensatz zur Planwirtschaft (in der man bekanntlich nur das bekommt, was von anderer Stelle als sinnvoll erachtet wird), hat der Computerbesitzer in der Hard- und Softwarewelt die Freiheit der Wahl (er wird nämlich nur das bekommen, was von anderer Stelle als sinnvoll erachtet wird):

Erstes Postulat der Wahlfreiheit (auch »Löwenzahn-Gesetz« genannt):

Egal, welches Anwendungsprogramm Du hauptsächlich einsetzt: Wenn Du Windows nicht verwenden willst, wird es nur in einer Windows-Version verfügbar sein.

Wie jedes murphyologische Gesetz kann auch das erste Postulat der Wahlfreiheit nicht dadurch ausgetrickst werden, indem man beispielsweise das Betriebssystem und/oder bevorzugte Anwendungsprogramm wechselt:

Zweites Postulat der Wahlfreiheit

Wenn Du Dein bevorzugtes Anwendungsprogramm unter Windows verwenden willst, wird es auf absehbare Zeit ausschließlich als DOS-Version verfügbar sein.

Erste Ergänzung des zweiten Postulats der Wahlfreiheit

Als DOS-Task unter Windows wird es so lange fehlerfrei arbeiten, bis es unter Datenverlust abstürzen kann.

Zweite Ergänzung des zweiten Postulats der Wahlfreiheit:

Auch jedes Windows-Programm wird nur so lange fehlerfrei arbeiten, bis es unter Datenverlust abstürzen kann.

Gorbatschow-Erweiterung des Löwenzahn-Gesetzes (auch »Winword 2.0-Erweiterung« genannt):

Wenn Du Dich endlich entschließt, von der DOS- auf die Windows-Version Deines bevorzugten Anwendungsprogramms umzusteigen, so sind inzwischen so viele neue Windows-Versionen Deines Anwendungsprogramms erschienen, daß in der aktuellen Windows-Version kein Importfilter mehr für die Daten der DOS-Version existiert.

Verallgemeinertes Postulat der Wahlfreiheit:

Die einzige Art, mit Löwenzahn oder anderem Unkraut im Garten umzugehen: Mach' es zu Deiner Lieblingsblume.

Erstes Multitasking-Gesetz:

Laufen zwei Prozesse gleichzeitig, wird der Unwichtige dem Wichtigen Rechenzeit wegnehmen.

Zweites Multitasking-Gesetz:

Der Absturz eines Multitasking-Tasks wird

- nur dann, wenn es der Anwender nicht kontrollieren kann
 - unter dem maximal möglichen Datenverlust sowie
 - zum ungünstigsten Zeitpunkt
- erfolgen.

Erste Erweiterung des zweiten Multitasking-Gesetzes:

Besonders gerne stürzen darum Hintergrund-Tasks ab.

Anatols Widerlegung der ersten Erweiterung des zweiten Multitasking-Gesetzes:

Vordergrund-Tasks stürzen genauso gerne ab.

Zweite Erweiterung des zweiten Multitasking-Gesetzes:

Jeder Task wird beim Absturz alle anderen Tasks sowie Benutzeroberfläche und Betriebssystem mitnehmen.

Drittes Multitasking-Gesetz (oder »Echtzeit-Erweiterung des Multitasking-Gesetzes«):

Jeder Task wird dabei mit dem Absturz so lange warten, bis sich in allen anderen Tasks genügend nicht-gespeicherte und unwiederbringliche Daten angesammelt haben.

Viertes Multitasking-Gesetz:

Sofern in irgendeinem Task ein Auto-Save installiert ist, wird das Speichern über den Cache-Controller erfolgen, was dazu führt, daß die Daten beim Absturz nicht auf die Festplatte geschrieben werden.

Ergänzungen zum vierten Multitasking-Gesetz:

1. Ist kein Cache-Controller installiert, wird der Absturz während des Speichervorgangs erfolgen, was zum Verlust aller, inklusive der bereits früher geschriebenen Daten führt.
2. In jedem anderen Fall ist anschließend die Festplatte kaputt.

Der Unterschied zwischen tastaturgesteuerten Programmen und mausgesteuerten grafischen Benutzeroberflächen ist dem murphyologisch geschulten Computeropfer dabei stets bewußt. Gilt beispielsweise bei DOS:

Das Kommandozeilen-Editor-Gesetz (auch als »DOSKEY-« oder »DOSEDIT-Gesetz« bekannt):

Der Kommandozeileneditor funktioniert nur dann nicht, wenn man mehrmals hintereinander eine so lange Zeile eintippen muß, daß er sich lohnen würde.

so ist Anwender wie Programmierer unter grafischen Benutzeroberflächen einer gänzlich anderen Gesetzmäßigkeit ausgeliefert:

Das Makro-Rekorder-Gesetz:

Der Makro-Rekorder funktioniert nur dann nicht, wenn man mehrmals hintereinander so lange und umständliche Pull- und Klickoperationen durchführen muß, daß er sich lohnen würde.

Solche Beispiele sind beliebig fortführbar. Gilt unter DOS (unabhängig ob mit dem Präfix »DOMES-«, »MS-« oder »DR-«):

Der CHKDSK-Grundsatz (oder »Lehrsatz von der Vergeblichkeit menschlichen Strebens«):

- A) Wenn Du einen CHKDSK über Deine Festplatte machst, wird das Programm keinen einzigen unzusammenhängenden Block finden.
- B) Verzichtest Du auf den CHKDSK, dann hast Du tausende davon.

Die »Speeddisk«-Erweiterung des CHKDSK-Grundsatz (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 Schopf packt und Deine Platte ruiniert.

so kann der Windows-Anwender sich verlassen auf

Die Windows-Erweiterung des CHKDSK-Grundsatz:

Ein Windows-Festplatten-Hilfsprogramm wird wahlweise

- erklären, daß es nur unter DOS arbeitet,
- die unzusammenhängenden Blöcke nicht bemerken und die Gelegenheit beim Schopf packen und Deine Platte ruinieren,

- die unzusammenhängenden Blöcke bemerken, in einer Alertbox die Alternativen »Weitermachen« und »Ignorieren« anbieten, und dann die Gelegenheit beim Schopf zu packen und bei jeder der gewählten Alternativen Deine Platte ruinieren.

Verallgemeinerungen für alle Windows-Applikationen:

1. Jede Windows-Anwendung weiß, was gut für Dich ist und was Du wirklich brauchst.
2. Es wird nie das sein, von dem DU denkst, daß es gut für Dich ist und was Du wirklich brauchst.

Die Zubehör-Konkretisierung (auch »Uhren-Setup-Gesetz« genannt):

Du wirst in den Menüs der Windows-Zubehör-Anwendungen nie die Dinge finden, die Du dort suchst. Im Setup-Menü der Uhr wirst Du beispielsweise alles einstellen können – nur nicht die Zeit.

Grunderkenntnis der grafischen Benutzeroberflächen:

Es gibt nur zwei unverrückbare Erkenntnisse im Leben:

1. Grafische Benutzeroberflächen machen das Bedienen von Computern einfacher.
2. Die Erde ist eine Scheibe.

Die Windows-Weisheiten der wahnsinnigen Wartezeit:

1. Wenn Du denkst, jetzt hast Du lange genug gewartet – dann verwandelt sich der Mauszeiger in eine Sanduhr.

2. Wenn Du denkst, Du hast noch nicht genug gewartet, ist Dein Rechner abgestürzt.
3. Wenn Du denkst, Dein Rechner sei abgestürzt und Du bootest neu, war er es nicht.
4. Wenn Du denkst, Dein Rechner sei nicht abgestürzt, ist er es doch.
5. Du hoffst immer vergebens.

Das fünffache Sanduhr-Phänomen (in Macintosh-Kreisen auch als »Armbanduhr-Wunder« bekannt):

1. Du bootest Deinen Rechner nach langer Wartezeit vor der Sanduhr just eine halbe Sekunde, bevor er mit seiner Aktion doch noch fertig wird.
2. Die Wiederherstellung des Zustands, in dem sich Dein Computer vor dem Booten befand, dauert mindestens zehnmal solange wie der Bootvorgang.
3. Dieser Faktor ist unabhängig von der Länge des Bootvorgangs (merke: je kürzer, desto länger dauert die Wiederherstellung; je länger, desto länger dauert die Wiederherstellung)
4. Wenn Du Dir den Bootvorgang sparen willst, ist der Computer unwiederbringlich abgestürzt.
5. Alle Alternativen bedingen den Verlust unwiederbringlicher Daten mit sich.

Joachims Philosophie:

Den Computer neu booten geht schneller als erst einen genialen Trick zu versuchen um anschließend aus- und wieder anschalten zu müssen.

Die Windows-Wiederlegung von Joachims Philosophie:

Du darfst nicht booten.

Die ultimative Definition einer grafischen und interaktiven Oberfläche:

Dreck 'n drop



*Entwickle ein Produkt. Bis Du es fertiggestellt hast,
ist es veraltet.*

C. Wie alles, was schiefgeht, in Vierkanal-Stereo schiefgeht: Die Multimedia-Extensions von Windows

Die Multimedia-Extensions sind der geglückte Versuch, einer langsamen und fehlerhaften grafischen Benutzeroberfläche durch die Implementation von einer zu langsamen CD-ROM-Schnittstelle und schlechten Audiofähigkeiten einen nicht ausgereiften Multimedia-Standard hinzuzufügen, der die Absturzhäufigkeit des Gesamtsystems drastisch erhöht:

Microsofts Multimedia-Definition:

Du weißt, daß es Multimedia ist, wenn Windows unter ohrenbetäubendem Lärm aus dem angeschlossenen Lautsprecher abstürzt.

Um diesen Vorzug allgemeingültig zu machen, wurde der Standard eines Multimedia-PCs aus der Taufe gehoben:

Die MPC-Erweiterung von Microsofts Multimedia-Definition:

Du weißt, daß es ein Multimedia-Standard ist, wenn die verschiedenen Komponenten, die Du wegen ihrer MPC-Kompatibilität für viel Geld gekauft hast, zueinander, zum Gesamtsystem sowie jeweils zu sich selber inkompatibel sind.

Die Ton-Tragödie:

Egal, wie Du die Multimedia-Extensions installierst: Du wirst nie alle zur Verfügung stehenden Datentypen mit dem Media-Player spielen können:

- Wenn Du WAV-Dateien spielen kannst, wirst Du keine Audio-Dateien spielen können.
- Wenn Du Audio-Dateien abspielen kannst, wird Deine Soundkarte allergisch auf MIDI-Dateien reagieren.
- Wenn Du MIDI-Dateien spielen kannst, wirst Du Dir keine WAV-Dateien anhören können.

Grafische Verallgemeinerung der Ton-Tragödie:

Du wirst auf jeden Fall nie in der Lage sein, dir MMM-Dateien anzusehen.

Das CD-ROM-Dilemma:

Wenn Du ein CD-ROM-Laufwerk installiert hast, wird jedes Programm, mit dem Du es ansprichst, versuchen, auf dieses Laufwerk schreibend zuzugreifen und abstürzen.

Regeln von der CD-ROM-Installation:

1. Du hast keinen freien Steckplatz für Deinen SCSI-Adapter mehr.
2. Hast Du noch einen freien Steckplatz, dann ist der SCSI-Stecker zu breit, als das Du ihn einstecken könntest.

3. Wenn Du zwei Steckkarten tauschen willst, um Dein CD-ROM-Laufwerk zu installieren, dann befindet sich an der einzigen Steckkarte, die Du ausbauen kannst, ein festgeschraubtes Kabel.
4. Die Schraube ist angerostet.
5. Das einzusteckende SCSI-Kabel ist immer möglichst unzugänglich angebracht.



Nach dem Ende einer langen Wartezeit wirst Du glücklich sein, wenn auf dem Bildschirm endlich die Sanduhr erscheint – und Du richtig warten darfst.

D. Das murphyologische Update: Windows 3.1

Wie bei jedem Softwareprodukt müssen auch bei Windows regelmäßige Updates sein, um den Anwender in Atem zu halten und nicht das Risiko einzugehen, er könnte sich an eine bestimmte Art von Arbeit gewöhnen:

Erstes generelles Update-Gesetz:

Wenn Du Dich an eine bestimmte Art des Arbeitens mit einem Programm gewöhnt hast, erscheint eine neue Version, die völlig anders bedient werden muß.

Zweites generelles Update-Gesetz:

Eine neue Version eines Programms erscheint auch dann, wenn Du Dich endlich dazu aufgerafft hast, Dir für die Fehler der alten Version Work-arounds zu schreiben.

Logische Erweiterung des zweiten generellen Update-Gesetzes:

Zwar sind die Fehler der alten in der neuen Version nicht behoben, jedoch äußern sie sich auf andere Art, weswegen die Work-arounds der alten Version nicht mehr funktionieren.

Thomas' Erkenntnis vom Fortschritt:

Der einzige Fortschritt von Windows 3.1 gegenüber Windows 3.0, den der Anwender bemerkt, ist der, daß die Zahl der Inkompatibilitäten höher ist.

Norberts Ergänzung:

... und daß Du deshalb mehr Geld für Updates ausgeben mußt.

Dagegen argumentiert Microsoft, daß Windows 3.1 gegenüber seiner Vorgängerversion eine ganz enorme Verbesserung im Geschwindigkeitsverhalten besäße. Auch hier ist der murphyologische Beweis dieser These sehr leicht anzutreten: Weist beispielsweise unter Windows 3.0 ein 25-MHz-PC mit 80386SX-Prozessor und 2 Mbyte Hauptspeicher mit einem beliebigen Windows-Anwendungsprogramm das gleiche Zeitverhalten auf wie ein 4,77-MHz-PC mit 512 Kbyte Hauptspeicher mit einem gleichartigen DOS-Anwendungsprogramm, so erreicht man dieses Zeitverhalten unter Windows 3.1 bereits mit einem 50-MHz-80486er mit acht Mbyte Hauptspeicher.

Windows-Erweiterungen des Dimensionsmirakels:

1. Du hast zuwenig Hauptspeicher.
2. Hat Du genügend Hauptspeicher, dann ist Deine Festplatte zu klein, um die dank grafischer Benutzeroberflächen enorm angeschwollenen Programme aufzunehmen.

Die NT- und OS/2-Ergänzung des Dimensionsmirakels:

Wenn Du Dein Computersystem auf die richtige Größe erweitert hast, wird eine neue Betriebssystem- oder Oberflächenversion auf den Markt kommen, für die Hauptspeicher, Festplatte oder beides wiederum zu klein ist.

Allgemeiner Grundsatz der Windows-Speicherverwaltung:

Du hast wahlweise zuwenig oder zuviel Speicher

Konkretisierungen des allgemeinen Grundsatzes der Windows-Speicherverwaltung:

1. Weniger als 16 Mbyte Hauptspeicher führen dazu, daß Deine Anwendungen unerträglich langsam arbeiten.
2. Genau 16 Mbyte Hauptspeicher führen dazu, daß Deine Anwendungen unerträglich langsam arbeiten.
3. Mehr als 16 Mbyte Hauptspeicher kann Windows nicht verwalten.

John Dvoraks Frage über das Verhältnis von Speicherplatz, Prozessorleistung und Arbeitsgeschwindigkeit:

»Warum verwenden wir nicht alle einfach GeoWorks, nur um es Ihnen mal zu zeigen?«

Ikas Seufzer:

Das einzige, was unter Windows 3.1 schneller geht, sind die Abstürze.

Bernhards generelle Update-Erfahrung:

Das Update funktioniert ohne teure Hardware-Erweiterung nicht.

Verschärfung von genereller Bernhards Update-Erfahrung:

Es funktioniert auch mit teurer Hardware-Erweiterung nicht.

Verschärfungen der Verschärfung:

1. Mit der teuren Hardware-Erweiterung funktioniert auch die alte Version nicht mehr.
2. Baust Du die teure Hardware-Erweiterung aus, wirst Du feststellen, daß Du die alte Version bereits gelöscht hast.
3. Der einzige Mensch, der im technischen Support des Herstellers bei Deinem Problem helfen könnte, hat soeben einen sechswöchigen Urlaub angetreten.
4. Nach Ablauf der sechs Wochen ist er für acht Wochen auf Schulung.
5. Rufst Du danach nochmal an, hat er die Firma verlassen.
6. In jedem anderen Fall wird er Dir auch nicht weiterhelfen können.

Um die Zahl der Inkompatibilitäten möglichst hoch zu halten, müssen nach Murphy besonders grundlegende Betriebssystem- und Benutzeroberflächen-Updates in möglichst kurzer Folge aufeinanderfolgen. Denn auch hier gilt:

Das Zeitdilemma der Hersteller (inzwischen nur noch »NT-Problem« genannt):

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

Folgen (Auch als »Jack's Desaster« bekannt):

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

Logische Konsequenz:

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

Was sich aus der Sicht der Softwareentwickler katastrophal für Umsatz und Profit auswirkt, ist für den Anwender katastrophal für seine Investitionssicherheit:

Anwendererkenntnis von der Updateankündigung (bekannt als »»3.1«-»Win 32«-»Windows NT«- Dreisprung«:

1. Wenn Du unter hohem Zeit- und Kostenaufwand ein Update auf die aktuelle Betriebssystemversion durchgeführt hast, wird der Hersteller eine neue Version ankündigen.
2. Gleichzeitig wird ein Aufruf an alle Entwickler und Softwarehäuser ergehen, doch keine Software mehr für Deine längst veraltete Betriebssystemversion zu entwickeln, sondern sofort auf die neue Version umzusteigen.
3. Das Softwarehaus, wegen dessen Updateankündigung Du die neue Betriebssystemversion installiert hast, wird daraufhin seine Ankündigung zurückziehen.
4. Die Punkte 1 bis 4 sind rekursiv.

OS/2-Erweiterung der Anwendererkenntnis von der Updateankündigung:

1. Auf dem konkurrierenden Betriebssystem kommt das von Dir dringend benötigte Programm pünktlich auf den Markt.
2. Es ist egal, für welche Version Du Dich entscheidest: Das benötigte Programm gibt es immer nur für die andere Plattform.

Hat sich der Computerbesitzer schließlich für eine bestimmte Betriebssystemversion entschieden, so greift

Die Windows-Paradoxie (auch »Gesetz der neueren Version des Setup-Programms« genannt):

Um Windows 3.1 zu installieren, mußt Du Windows 3.1 bereits installiert haben.

Das Setup-Phänomen (auch »US-Tastaturlayout-Gesetz« genannt):

Du wirst zwar stets in der Lage sein, diejenigen Windows-Komponenten einzustellen, die es unmöglich machen, Windows ohne Absturz zu starten. Doch es wird Dir nicht gelingen, diese Einstellungen rückgängig zu machen, selbst wenn Du es rechtzeitig bemerkst.

Folgerung aus dem Setup-Phänomen:

Egal was Du einstellst und wie Du das tust: Du wirst Windows neu installieren müssen.

Grundlegende Installations-Beobachtung:

Die Windows-Vorgaben werden auf Deinem Computersystem stets das schlechtestmögliche Ergebnis hervorrufen.

Konkretisierungen der grundlegenden Installations-Beobachtung:

1. Die Farben werden so installiert sein, daß Du nichts erkennen kannst.
2. Die Schriften werden so installiert sein, daß Du nichts lesen kannst.
3. Die Schnittstellen werden so installiert sein, daß Du die Maus nicht bewegen kannst.

4. Das Desktop wird so installiert sein, daß Du nichts findest.
5. Die Tastatur wird so installiert sein, daß Du nur kryptische Zeichen tippen kannst.
6. Der Drucker wird so installiert sein, daß er nichts druckt.
7. Du wirst bestenfalls als Ländereinstellung »Altgriechisch« bekommen.
8. Datum und Uhrzeit werden das falsche Format haben und außerdem die Systemzeit des kaputtgegangenen Computers auf dem Schreibtisch daneben verwenden.
9. Das von Dir verwendete Netzwerk wird Windows nicht unterstützen.
10. Die einzig möglichen Einstellungen für den Virtuellen Speicher werden 1024 Byte oder 16 Gbyte sein.
11. Du wirst nur dann Systemklänge installiert haben, wenn Du keine willst.
12. Im anderen Fall wird Windows schweigen.
13. Die mitgelieferten Treiber Deiner Hardware wird Windows nicht finden.

**Erweiterungen der grundlegenden Installations-
Beobachtung (auch »Windows-3.0-DOS-Shell-
Gesetze« genannt):**

1. Windows 3.1 wird bei seiner Installation nur diejenigen Treiber und Zusatzprogramme von Windows 3.0

überschreiben, die auf Deinem Computersystem bisher ihre Arbeit klaglos und weitgehend fehlerfrei verrichtet haben und sie durch fehlerhafte und mit Deinem Computersystem nicht zusammenarbeitende ersetzen.

2. Die bislang fehlerhaften oder mit Deinem Computersystem nicht zusammenarbeitenden Treiber und Zusatzprogramme bleiben erhalten.

Verallgemeinerung der grundlegenden Installations-Beobachtung:

Dein »Windows-kompatibles« Programm wird mit jeder Sekunde seiner Existenz mehr inkompatibel. Wo das nicht hilft, hilft Microsoft.

Gesetz von der komplizierten Windows-Installation auf Notebooks:

1. Die Betriebssystemversion, die für Dein Notebook im richtigen Diskettenformat vorliegt, ist zu alt für das gewünschte Anwendungsprogramm.
2. Hast du die richtige Windows-Version, dann läuft es trotzdem nicht.
3. Hast du alles richtig gemacht, läuft es auch nicht.

Fazit:

Du mußt immer diejenige Software installieren, die am kompliziertesten zu beschaffen oder am schwierigsten zu konvertieren ist. Dann läuft sie trotzdem nicht.

Erstes Gesetz von der Speicherung von benutzerdefinierten Einstellungen:

Ein Programm wird immer nur die Einstellungen speichern können, die unnötig sind.

Zweites Gesetz von der Speicherung von benutzerdefinierten Einstellungen:

Ein Programm wird immer nur die Einstellung speichern wollen, die es nicht soll.

Drittes Gesetz von der Speicherung von benutzerdefinierten Einstellungen:

Ein Programm wird immer die Einstellungen nicht speichern können, die notwendig sind und bei dem der Aufwand, die Einstellungen einzurichten, maximal groß ist.

Das Speicher-Paradoxon:

Wenn Du den Auswahlpunkt »Einstellungen nach dem Beenden nicht speichern« einstellst, wird diese Einstellung gespeichert.

Folgerung aus dem Speicher-Paradoxon:

Der Auswahlpunkt »Einstellungen nach dem Beenden speichern« wird immer falsch eingestellt sein.

Konkretisierungen:

1. Wenn Du aus Versehen Fenster und Icons verschiebst, wird Windows das entstandenen Chaos speichern und dadurch Deine mühsam entwickelte Oberfläche zerstören.

2. Wenn Du Fenster und Icons sorgfältig anordnest, wird Windows nicht speichern und dadurch Deine mühsam entwickelte Oberfläche zerstören.
3. Ist der Menüpunkt korrekt auf Speichern eingestellt, wird Windows ohne zu speichern abstürzen.

Gesetz vom unsichtbaren Icon:

1. »Automatisch anordnen« der Icons ist nur dann angewählt, wenn Du so viele Icons im Programmmanager-Fenster hast, daß sie nicht alle sichtbar hineinpassen.
2. Das von Dir gewünschte Programm wird sich außerhalb des geöffneten Fensters verstecken.

Folgerung aus dem Gesetz vom unsichtbaren Icon:

Du wirst das von Dir gesuchte Programm nie finden.

Allgemeine Erweiterung:

Auch Programmgruppen können sich vor Dir verstecken.

Das DOS-Icon-Desaster:

Windows wird immer dann ohne Veranlassung auf ein Standard-DOS-Icon zurückschalten, wenn Du Dich an das Aussehen Deiner Icons gewöhnt hast.

Das Sysedit-Gesetz:

Regeln oder Hilfsprogramme, die die Arbeit mit Windows spürbar erleichtern, stehen nicht im Handbuch.

Erweiterung:

Stehen die Hilfsprogramme doch im Handbuch, hast Du sie nicht installiert.

Erste Erkenntnis von der Unsinnigkeit der Autostart-Gruppe:

Nur die Programme, die eine maximal mögliche Ladezeit aufweisen und die Du nicht brauchst, rutschen in Deine Autostart-Gruppe.

Zweite Erkenntnis von der Unsinnigkeit der Autostart-Gruppe:

Du wirst sie genausowenig wieder daraus entfernen können, wie es Dir möglich ist, Programme, die nach dem Windows-Start tatsächlich aufgerufen werden sollen, dort so zu installieren, daß sie richtig gestartet werden.



*Jede wirklich sichere Fehlerbekämpfungsmethode
wird todsicher Deine Daten endgültig vernichten.*

E. Das Alphabet der Pannen: Windows-Anwendungen von A bis Z

Windows-Anwendungsprogramme setzen sich aus Textverarbeitungs-, Dateiverwaltungs-, Grafik-, Tabellenkalkulations- und Telekommunikationsprogrammen zusammen sowie mit den berechtigten Forderungen des Anwender widerstrebend bis ablehnend auseinander. Die einzige Aufgabe von in kompilierte Programmzeilen gegossener Hinterhältigkeit (kurz »Programm« genannt) ist es bekanntlich, reibungslos, aufgabengemäß und mit hohem Datendurchsatz so zu versagen, daß in kürzestmöglicher Zeit bei allen Beteiligten und Unbeteiligten der größtmögliche Schaden angerichtet wird.

Das Verhältnis zwischen der Absturzhäufigkeit von DOS- und von Windowsprogrammen kann in einer exakten Formel ausgedrückt werden. Ist »D« die Anzahl der Abstürze eines beliebigen DOS-Programms pro Stunde, so nimmt der vergleichbare Wert der Windows-Version »W« stets folgenden Wert an:

$$W = (D * D * I) @ U$$

wobei $U \geq \infty$

wobei »I« stets den Wert von »nimm irgendeine Zahl, sofern sie nur groß genug ist und verdopple sie«

annimmt. »U« ist dabei der Ungünstigkeitswert des Zeitpunkts, zu dem der Absturz eintritt (Mathematiker, die nun das Argument anführen, daß es keinen Wert geben kann, der größer ist als unendlich, mögen zwar in einem logischen Universum recht haben, nicht aber in einem Murphyologischen, in dem wir bekanntlich alle leben).

Dies bedeutet beispielsweise für die Hersteller und Anbieter von Windows-Software, daß der finanzielle und personelle Aufwand für Programmentwicklung und -marketing in einem stets negativen Verhältnis zu den benötigten Supportausgaben steht. Die einzige Windows-Software, die keinen Support benötigt, ist logischerweise diejenige, für die schon vor Markteinführung eine zu große und zu teure Supportabteilung eingerichtet wurde.

Aus Anwendersicht stellt sich allerdings

Die Windows-Erweiterung zur Morschhauser-Frage:

Support für Windows-Software? Was ist das?

Die Microsoft-Antwort auf die Windows-Erweiterung zur Morschhauser-Frage:

Dasselbe wie Support für Windows.

Eben darum konnte die Morschhauser-Frage bis heute noch von niemandem beantwortet werden:

Konkretisierungen der Morschhauser-Frage:

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

Da alle großen Entwicklungen von Windows-Software aufgrund gravierender Programmfehler verwirklicht wurden, folgt daraus mit murphyologischer Notwendigkeit, daß jede Windows-Software Fehler hat:

Zweite Folgerung aus dem Windows-Softwaregrundsatz:

Jedes Programm hat immer einen Fehler mehr.

Dritte Folgerung aus dem Windows-Softwaregrundsatz:

Die Beseitigung eines Fehlers ruft mindestens zwei neue hervor.

Persönliche Ableitungen aus dem Windows-Softwaregrundsatz:

1. Wenn die Fehler sich bemerkbar machen, dann bei Dir.
2. Egal was Du tust, Dein Windows-Programm bringt eine Fehlermeldung.

Erster logischer Umkehrschluß aus der Existenz von Utilities:

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

Zweiter logischer Umkehrschluß aus der Existenz von Utilities:

Die Tatsache, daß es sehr viele große, unnütze Windows-Programme gibt, beweist, daß es noch mehr kleine unnütze Windows-Programme gibt, da es ja mehr kleine als große Programme gibt.

Dieters Ergänzung:

Alle unnützen Programme hast Du gekauft.

Boris' Folgerung:

Die Effektivität eines Programms ist umgekehrt proportional zum Verkaufspreis.

Benjis Weissagungen über Fehlermeldungen:

1. Dein Windows-Programm wird nur dann mit einer Fehlermeldung abstürzen, wenn Du kein Handbuch greifbar hast.

2. Hast Du ein Handbuch greifbar, dann sind die Fehlermeldungen nicht erklärt.
3. Wenn Fehlermeldungen erklärt werden, dann alle bis auf eine: diejenige, die bei Dir aufgetreten ist.

Peters allgemeine Erkenntnisse über Windows-Software:

1. Das Windows-Programm, das Du Dir kaufst, ist nicht den Preis wert, den es gekostet hat.
2. Kein Windows-Programm ist so gut, wie das Bild auf der Verpackung.
3. Eine gute Kritik in einer Windowszeitschrift besagt nur, daß das Programm dem Redakteur gefallen hat. Du wirst immer einen anderen Geschmack haben als er.
4. Du wirst erst nach dem Kauf merken, daß das Programm für ein Problem entwickelt wurde, das es entweder nicht gibt oder das Du nicht hast.

Postulat der Multifunktionalität:

Je weniger Funktionen ein Programm hat, umso perfekter verstehen sich seine Fehler untereinander.

Folgerungen aus dem Postulat der Multifunktionalität:

1. Je mehr Funktionen ein Programm hat, umso mehr Möglichkeiten zum Versagen besitzt es.

Die Paintbrush-Präzisierung:

1. Die Undo-Funktion funktioniert nur, solange Du sie nicht brauchst.
2. Im besten Fall nimmt sie alle Aktionen zurück, die Du seit Programmstart durchgeführt hast.

Der entscheidende Unterschied zwischen Windows-Programmen und DOS-Programmen ist, daß Du auf jeden Fall die falsche Version hast und ein Update auf die andere brauchst:

Marcel's sechs Windows-Verschärfungen:

1. Je teurer das Update von einem DOS-Programm auf die Windows-Version, umso geringer der Nutzen.
2. Je teurer das Update von einem Windows-Programm auf die neue Version, umso geringer der Nutzen.
3. Je teurer das Update von einem Windows-3.0-Programm auf die Windows-3.1-Version, umso geringer der Nutzen.
4. Je teurer das Update von einem Windows-3.1-Programm auf die neue Windows-3.1-Version, umso geringer der Nutzen.
5. Je höher die Versionsnummer, umso größer der Speicherverbrauch.
6. Das Feature, auf das Du seit Jahren wartest, wird auch in der neuen Version nicht implementiert sein.
7. Der Fehler, über den Du dich seit jeher ärgerst, wird auch in der neuen Version noch vorhanden sein.

8. Auf jeden Fall werden in der neuen Version Menüs und Shortcuts komplett umgestellt sein.

Das 357-Desaster:

Wenn Du durch den Kauf teurer Zusatzhardware erreicht hast, daß jeder Deiner Verkäufer dasselbe Windows-Verkäuferunterstützungsprogramm einsetzen kann, wirst Du feststellen, daß es für die Zwecke Deiner Firma völlig unbrauchbar ist.

Das ABC-Flowchart-Argument:

Erst wenn das mit Deinem Windows-Programm entwickelte Ablaufdiagramm als Ausdruck auf dem Schreibtisch Deines Chefs liegt, wirst Du feststellen, daß Du vergessen hast, die entscheidenden Teile vor dem Ausdruck wieder einzublenden.

Das Access-SQL-Axiom:

Dein Management-Informationssystem wird stets

- in den richtigen Datenbanken nach den falschen Daten suchen,
- in den falschen Datenbanken nach den richtigen Daten suchen,
- in den falschen Datenbanken nach den falschen Daten suchen.

Ansonsten wird es zwar in den richtigen Datenbanken nach den richtigen Daten suchen, diese jedoch falsch oder gar nicht ausgeben.

In jedem anderen Fall sind die gefundenen Daten für Dein Problem irrelevant.

Die Afterdark-Aversion:

Dein Windows-Bildschirmschoner wird sich bevorzugt dann einschalten, wenn Dein Grafikprogramm gerade dabei ist, eine komplexe Zeichnung aufzubauen. Dann wirst Du gezwungen sein, die Maus zu bewegen, um den Zeichnungsaufbau erneut zu verfolgen.

Die Adimens-Agonie:

Programme, die von GEM auf Windows portiert wurden, werden stets sämtliche GEM-Probleme haben, die sie in Windows-Geschwindigkeit offenlegen.

Die Ami-Pro-Anwendungen:

Es gibt fünf Arten, mit einer grafisch orientierten Textverarbeitung mehr zu leisten, als mit einer herkömmlichen Schreibmaschine:

- Du kannst mit Textbausteinen arbeiten, also automatisch fehlerhafte, unnötige und sinnentstellende Textpassagen an Stellen einfügen, an die sie nicht passen.
- Du kannst mit Kopf- und Fußnoten arbeiten, also automatisch fehlerhafte unnötige und sinnentstellende Textpassagen an unterschiedliche Stellen auf unterschiedlichen Seiten plazieren, wo sie Dich als Autor stören und der Leser sie nicht findet.
- Du kannst mit der Serienbrieffunktion arbeiten, also automatisch veraltete Adressen in einen fehlergespickten Brief automatisch so einfügen lassen, daß die Hälfte der Adressen falsch ausgedruckt wird,

die Post vierzig Prozent mit dem Vermerk »unbekannt« zurückschickt und die anderen zehn Prozent von den Empfängern weggeschmissen werden.

- Du kannst mit der Graphik- und Layoutfunktion arbeiten, also automatisch falsche Grafiken in niedriger Auflösung so in Deinen Text integrieren, daß man die wichtigen Textstellen nicht mehr lesen kann und nur offensichtliche Irrtümer und sinnentstellende Tippfehler deutlich hervorgehoben zu lesen sind.
- Du kannst mit Deiner Schreibmaschine schneller tippen.

Die Approach-Abartigkeit:

Deine Windows-Datenbank wird stets darauf optimiert sein, das, was Du eingegeben hast, nicht wieder heraussücken zu müssen.

Die Animation-Works-Allegorie:

»Drag and drop« ist die Methode, etwas sehr langsam und mühevoll über den Bildschirm zu bewegen. Animation auch.

Die Arts&Letters-Ableitung:

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

Die Authorware-Anarchie:

Entwickle ein Autorensystem, mit dem Deine Programmierer interaktiv und intuitiv programmieren können und du wirst feststellen, daß Programmierer weder aktiv noch intuitiv sind.

Konkrete Umsetzungen der Authorware-Anarchie:

1. Dein Windows-Autorensystem wird Programme generieren, die mehr Fehler haben als es selbst.
2. Dein Windows-Autorensystem hat viele Fehler.

Die Beckertools-Belastungen:

1. Dein Windows-Kopiertool wird stets auf das falsche Laufwerk beziehungsweise in das falsche Unterverzeichnis kopieren.
2. Es wird Dich dabei nicht informieren.
3. Laufwerksicons werden Dir nicht zur Verfügung stehen.
4. Du wirst eine Diskette, die sich bereits im Laufwerk befindet auf jeden Fall mit der falschen Aufzeichnungsdichte formatieren.
5. Bei einer HD-Diskette wirst Du auf »360 Kbyte« klicken.
6. Bei einer DD-Diskette wirst Du auf »1,2 Mbyte« klicken.
7. Daß Du falsch 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.

Die Bookshelf-Begründung:

Deine multimediale Windows-Lexikonsoftware wird Dir zu einem von Dir gesuchten Stichwort stets

- einen unverständlichen Text
- einen unverständlichen Text und eine Grafik in falscher Auflösung
- einen unverständlichen Text, eine Grafik in falscher Auflösung und eine rucklige Animation ausgeben.

In jedem anderen Fall wird es das Stichwort nicht finden oder abstürzen.

Multilexikalische Erweiterung der Bookshelf-Begründung:

In einer multimedialen Windowslexikon-Sammlung wird der von Dir gesuchte Begriff immer im letzten der von Dir durchsuchten Lexika sein.

Im anderen Fall befindet er sich auf einer Erweiterungs-CD, die Du nicht hast.

Das Charisma-Chaos (Leitsatz):

Du wirst Deine Daten nie in einer vernünftigen Form mit Deinem Präsentationsprogramm aufbereiten können.

Das Charisma-Chaos (Konkretisierungen):

1. Bei einem senkrechten Balkendiagramm werden die Balkenbezeichnungen zu lang sein, um sie komplett unter die Balken zu bekommen.
2. Bei einem waagrechten Balkendiagramm werden die Balkenbezeichnungen zu lang sein, um Balken und Bezeichnungen komplett auf die Seite zu bekommen.

3. Bei einem Tortendiagramm ist immer ein Kuchenstück zu klein, als daß Du die Bezeichnungen auf die Kuchenstücke setzen kannst.
4. Wenn Du versuchst, die Bezeichnungen der Kuchenstücke außen zu plazieren, wirst Du immer mehrere kleine Kuchenstücke haben, so daß sich die Bezeichnungen unleserlich überlappen.
5. Bei einem Liniendiagramm wird immer ein Wert so groß sein, daß die Unterschiede bei den anderen nicht mehr erkennbar sind.
6. Bei einem Area-Chart wird der unwichtigste Wert immer unten liegen und sich am stärksten verändern, so daß die Veränderungen der anderen nicht mehr erkennbar sind.
7. Für ein Textchart hast Du immer zuviel Text.
8. Bei einem Korrelationsdiagramm wird die Korrelation bei Dir nie sichtbar.

Erweiterungen des Charisma-Chaos:

1. Zwei verknüpfte und überlagerte Charts wirst Du nie wieder voneinander trennen können.
2. Du wirst bei jedem Fehler beide Charts neu aufbauen müssen.
3. Die wirklich interessanten Stellen beider Charts werden sich in jedem Fall überlagern.

Die Chartbuilder-Chronologie (auch

»Verallgemeinerung des Charisma-Chaos« genannt):

Wenn Du das Präsentationsprogramm wechselt, werden sich Deine Probleme nicht lösen, sondern es werden zu den bekannten nur verheerendere, unbekannte und schwer zu bemerkende hinzukommen.

Der Cirrus/Access-Clou:

Egal wie groß und standardisiert ein Marktsegment ist, Microsoft kann es umdefinieren.

Erweiterung des Cirrus/Access-Clous (oder »Borland-Erkenntnis« genannt):

Wenn Du davon Schaden hast, wird Microsoft es tun.

Der Cobra-Adress-Chose:

Der Datenaustausch zwischen Deiner Windows-Textverarbeitung und Deiner Windows-Adressenverwaltung wird nur dann funktionieren, wenn die übertragene Adresse unvollständig oder falsch ist.

Die Color-Lab-Clownerie:

Dein Windows-Bildbearbeitungsprogramm wird Farben auf dem Bildschirm stets richtig, auf dem Farbdrucker stets falsch und auf dem Schwarzweiß-Drucker stets gar nicht wiedergeben.

Erweiterung der Color-Lab-Clownerie für Zeichen- und CAD-Programme:

Dein Windows-Zeichen- und Dein Windows-CAD-Programm hingegen werden Farben auf dem Bildschirm stets richtig und auf Farb- und Schwarzweiß-Druckern stets gar nicht wiedergeben.

Verallgemeinerung der Color-Lab-Clownerie:

Alle drei Programmarten geben jedoch auf jeden Fall das, was auf dem Bildschirm schwarz beziehungsweise weiß war, auf dem Ausgabegerät schwarz beziehungsweise schwarz wieder.

Der Comfo-Desk-Coup:

Jeder grafischer Betriebssystemaufsatz auf Windows wird Windows-Fehler dergestalt optimieren, daß es Dir gelingt, in wesentlich kürzerer Zeit wesentlich mehr und wesentlich schwerwiegendere Pannen zu produzieren, ohne dabei neue Pannen unterdrücken zu können.

Die Concerto-Document-Crux:

Dein Windows-Dokumentenverwaltungssystem wird Die von Dir verwendeten Dokumente

- nicht in der nötigen Geschwindigkeit speichern
- an den falschen Plätzen ablegen
- fehlerhaft indizieren
- falsche Querverweise aufbauen
- mit den falschen Dokumenten verknüpfen
- nicht wieder ausgeben.

In jedem anderen Fall wird es Dir nicht möglich sein, gefundene Dokumente auf dem Bildschirm wieder darzustellen oder auszudrucken.

Die Connect-Chance:

Eine sich selbst adressierende E-Mail wird sich an »E-Mail« adressieren.

Erste Erweiterung der Connect-Chance:

Jede wirklich wichtige Nachricht, die über ein Mailbox-netz verschickt wird, wird eine übereifrige Programm-routine fehlerhaft als Rekursion entlarven und abfangen.

Zweite Erweiterung der Connect-Chance:

Hast Du aber völlig übermüdet eine unsinnige, lächerliche und falsche Nachricht eingetippt, wird sie Stunden später im gesamten Netz zu lesen sein.

Dritte Erweiterung der Connect-Chance (Auch »Die Grundregel von der Übertragungsgeschwindigkeit« genannt):

1. Jede zu hohe Baudrate führt zu »)%'!/_&y\«.
2. Jede andere Baudrate führt zu »)%'!/_&y\«.
3. Hingegen wird die Standardeinstellung Deines Windows-Terminalprogramms zu »)%'!/_&y\« führen.

Pekuniäre Umsetzung der dritten Erweiterung der Connect-Chance:

1. Windows wird dabei zu verhindern wissen, daß Dein Modem beziehungsweise Dein Terminalprogramm die Verbindung unterbricht.
2. Dies wird vorzugsweise dann geschehen, wenn Du eine Auslandsverbindung aufgebaut hast.
3. Ist dies nicht der Fall, befindest Du Dich in einer 300-Dollar-pro-Stunde-Onlinedatenbank.

4. Im Regelfall wirst Du über eine Auslandsverbindung eine 300-Dollar-pro-Stunde-Onlinedatenbank angewählt haben.

Der Corel-Draw-Comme-il-faut (Grundlegende Verallgemeinerung):

Jedes vektororientierte Windows-Zeichenprogramm ist so anständig zu sich selbst, daß es einer importierten Bitmapgrafik mit tiefer Verachtung begegnet.

Der Corel-Draw-Comme-il-faut (Konkretisierungen):

1. Du wirst immer nur die Bitmap-Grafik oder die Vektorgrafikelemente auf dem Bildschirm zu sehen bekommen.
2. Die Bitmap-Grafik wird immer so skaliert sein, daß Du sowohl auf dem Monitor als auch auf Drucker und Belichter Moirees zu sehen bekommst.
3. Die wichtigsten Funktionen Deines Vektorgrafikprogramms wirst Du für die Bitmap-Grafik nicht anwenden können.
4. Wenn Du versuchst, die Bitmap-Grafik zu vektorisieren, wird eine vorhandene Auto-Trace-Funktion wahlweise zu viel oder zuwenig glätten.
5. Du erhältst deshalb alternativ stets entweder ein mit dem Bitmap-Original unähnliches, nur fünf Beziérpunkte umfassendes oder ein mit dem Bitmap-Original unähnliches, fünftausend Beziérpunkte umfassendes, jede Pixeltreppe nachfahrendes und nicht mehr zu bearbeitendes Vektorgrafikobjekt.

Die CPAntivirus-Contrainjektionen:

1. Das Einzige, wofür sich Deine heranwachsenden Kinder an Deinem Computer interessieren, sind die zahlreichen Raubkopien von Windows-Spielen, die sie während Deiner Abwesenheit auf die Festplatte kopieren.
2. Diese Tätigkeit wird jeden existierenden Virus im Umkreis von 400 Kilometern auf Deinen Computer befördern.
3. Du hast immer einen Virus mehr, als Du meinst.
4. Du hast immer einen Virus mehr, als Dein Antivirenprogramm meint.
5. Du hast immer einen Virus mehr, als Dein Antivirenprogramm findet.

Erweiterungen zu den CPAntivirus-Contrainjektionen:

1. Dein Antivirenprogramm wird, wenn es etwas findet, nur den harmlosesten Deiner Viren finden.
2. Wenn Du keinen harmlosen Virus hast, findet Dein Antivirenprogramm die Suchstrings in seiner eigenen Konfigurationsdatei.
3. Du hast nie einen harmlosen Virus.

Die Crosstalk-Camouflage:

1. Windows und Dein Windows-Terminalprogramm werden sich nie darüber verständigen können, wer von beiden gerade auf die serielle Schnittstelle Deines Computers zugreifen darf.

2. Du wirst stets dann eine Systemfehlermeldung mit anschließendem Verbindungsabbruch und Löschen des Empfangspuffers bekommen, wenn Du aus einer teuren Online-Datenbank oder Mailbox bereits 95 Prozent eines sehr langen Dokuments heruntergeladen hast.

Das da-Vinci-Desaster:

Eine elektronische Mitteilung, die Du über Dein Windows-E-Mail-Programm verschickst, wird wahlweise

- über das falsche WAN geroutet
- in das falsche LAN geroutet
- an den falschen Server geschickt
- beim falschen Empfänger ablegt.

In jedem anderen Fall wird der Empfänger nie in sein elektronisches Postfach schauen.

Pekuniäre Erweiterung des da-Vinci-Desasters (auch als »UUCP-Gesetz bekannt):

Je länger eine elektronische Mitteilung ist, umso größer ist die Anzahl der Stationen, über die die Mitteilung geroutet wird und umso länger kreist die falsch adressierte Mitteilung im Netz und verursacht Übertragungsgebühren, die letztlich Dir in Rechnung gestellt werden.

Die Data-Bridge-Deutung:

Die Übernahme von Daten aus DOS-Datenbanken in Windows-Applikationen bedeutet, daß die sinnvollen Daten herausgefiltert werden und Du nur noch die

nichtdruckbaren Steuerzeichen und die falschen Formatierungen in Deine Applikation übernimmst.

Die dbFast-Definition:

xBase = uBase

Konkrete Anwendungen der dbFast-Definition:

1. dBase-kompatible Datenbanksysteme sind es nicht.
2. Jeder kann Dich vom Gegenteil überzeugen. Mit Ausnahme Deines Quellcodes.

Das dreifache dbFast-Dilemma (kurzgefaßt: »2b .OR. .NOT. 2b«):

1. Du hast immer ein ENDIF zuviel.
2. Ansonsten hast Du eines zuwenig.
3. Stimmt die Zahl der ENDIFs, dann ist ein IF zuviel oder zuwenig.

Erweiterung des dreifachen dbFast-Dilemmas:

Dieses gilt auch für jegliche andere Schleifenkonstruktion.

Das DCA-Irma-Debet:

Jede 3270-Host-Emulation unter Windows wird die Unzulänglichkeiten von 3270-Terminals mit den inhärenten Windows-Fehlern kombinieren.

Das Design/IDEF-Debakel:

Dein Datenmodellierungsprogramm wird immer wahlweise die falschen Daten oder das falsche Datenmodell verwenden. In der Regel beides.

Die Design/OA-Deviation:

Jede Datenbank-Frontend-Software unter Windows wird bestrebt sein, Deine Daten in möglichst unübersichtlicher Art und Weise auf dem Bildschirm darzustellen.

Das Designer-Drama:

Wenn Du ein Windows-Programm auf Deine zweite Festplatte D: installieren willst, wird das Installationsprogramm erst feststellen, daß auf C: zuwenig Speicherplatz ist, bevor Du auswählen kannst, ob es auf C: oder D: installieren soll.

Folgerung aus dem Designer-Drama:

Du wirst immer gezwungen sein, mehrere Mbyte auf Deinem C-Laufwerk zu löschen, bevor Du auf D: installieren kannst.

Die DOS-Cockpit-Dämonie:

Deine Windows-Hilfsprogrammsammlung wird das Tool, daß Du benötigst, nicht beinhalten.

Der Draw-Plus-Dauerbrenner:

Jedes Programm ist bestrebt, die zeitaufwendigste Funktion (zum Beispiel Bildaufbau bei einer langsamen Grafikkarte, Neuberechnung einer Mandelbrotmenge mit zwanzig Iterationen oder ähnliches) so häufig wie irgendmöglich durchzuführen, egal ob es sinnvoll ist oder nicht.

Der DrawCel-Diabolismus:

Jedes Zeichenprogramm, daß auf die Daten Deiner Tabellenkalkulation zugreift, wird die Daten dergestalt

falsch auswerten und darstellen, daß das Ergebnis plausibel aussieht und zu den falschen Schlußfolgerungen und Aktivitäten führt. Richtig dargestellte Daten führen hingegen zu den falschen Schlußfolgerungen und Aktivitäten.

Die EQ!-Erweiterungen:

1. Der automatische Syntax-Checker in Deinem Formelgenerator wird erfolgreich verhindern, daß Du den von Dir gewünschten Formelteil entwickelst.
2. Einen nicht ersichtlichen Formelfehler wird der Syntax-Checker nicht bemerken.

Die Excel-Exaltiertheit (Leitsatz):

Wenn Du Dich endlich entschlossen hast, ein Tabellenkalkulationsprogramm anzuschaffen, wirst Du bemerken, daß Du damit mit viel zu großem Aufwand viel zu viele Zahlen in unübersichtliche Formeln verschlüsseln mußt, um hinterher das festzustellen, was Du vorher ohnehin bereits wußtest.

Präzisierungen der Excel-Exaltiertheit:

1. Ein Fehler in einer Formel entsteht nur dann, wenn die fehlerhaften Ergebnisse plausibel sind.
2. Wenn Du mit Einheiten rechnen mußt, wird Geschwindigkeit in amerikanischen Seemeilen per 19 Tagen ausgegeben werden.
3. Geldwerte werden stets dann in der falschen Währung angegeben, wenn die Einheit nicht mit ausgegeben wird.

4. Bevor eine Tabellenkalkulation auch nur näherungsweise brauchbare Ergebnisse liefern kann, wird das Programm seinen Zufallszahlengenerator einschalten.

Dreidimensionale Erweiterung der Excel-Exaltiertheit:

Der einzige Effekt, den die Dreidimensionalität Deines Windows-Spreadsheet hat, ist die Tatsache, daß sich Fehler jetzt zusätzlich noch hinter anderen Objekten verstecken können.

Die Extra!-Einsicht:

Wenn Du versuchst, mit Deinem Windows-Dateitransferprogramm in einem heterogenen Netzwerk Dateien zwischen Rechnern mit verschiedenen Betriebssystemen auszutauschen, wird das einzige, was auf den Zielrechnern ankommt der falsch konvertierte Dateiname sein.

Die Facelift-Font-Folgerung

Es gibt nur drei Arten von Windows-Fonts:

1. Diejenigen, die zu groß sind, um komplett auf Deinem Bildschirm dargestellt werden zu können.
2. Diejenigen, die zu klein sind, als daß Du sie auf Deinem Monitor mit Deiner Grafikkarte noch lesen kannst.
3. Diejenigen, die Dein Drucker nicht druckt.

Die Faxit-Folter:

Faxen aus Windows heraus ist der gelungene Versuch, ein falsch formatiertes Dokument mit dem falschen An-

sprechpartner aus der Fax-Adreßverwaltung so an die verkehrte Nummer abzusenden, daß Windows durch einen Schnittstellenfehler abstürzt, Dokument und Datenbank löscht und Du hinterher feststellst, daß Du ein Dokument übertragen hast, daß die Firma unter keinen Umständen verlassen durfte.

Die Filemaker-Fiesheit (Grundsatz):

Datenbank-Oberflächen sind dazu da, Dich nicht durchzulassen.

Die Filemaker-Fiesheit (Konkretisierung):

1. Deine Datenbankoberfläche wird diejenige sein, die am umständlichsten zu bedienen ist.
2. Vergleichbare Programme, die Du bei anderen siehst, könntest Du sofort und im Schlaf bedienen.

Die Forest&Trees-Feststellung:

1. Die wahre Funktion eines »Management Informationssystem unter Windows« ist, daß Dein Chef erkennt, daß die Anschaffung des entsprechenden Programms entgegen Deiner Behauptung unsinnige Geldverschwendung war.
2. Er wird Dich dafür verantwortlich machen.

Die Formmaster-Formel (auch »allgemeines Datenbett des Prokrustes« genannt:

Du wirst in Deiner Formulardatenbanksoftware immer nur solche Formulare zur Verfügung haben, die zu Deinen Daten nicht passen und nur unter großem Aufwand und unter Aufgabe jeglichen inhaltlichen Bezug inner-

halb der Datensätze in das Formular gepreßt werden können.

Die Freelance-Handbuch-Frechheit (Leitsatz):

Wenn eine potentielle Fehlerquelle in einem Handbuch zur Sicherheit zweimal erklärt ist, dann unterscheiden sich beide Erklärungen grundlegend voneinander.

Die Freelance-Handbuch-Frechheit (erste Ableitung):

Du kannst auf keinen Fall erkennen, welche von den beiden Erklärungen die richtige ist.

Die Freelance-Handbuch-Frechheit (zweite Ableitung):

Du kannst grundsätzlich davon ausgehen, daß beide falsch sind.

Die Freelance-Handbuch-Frechheit (dritte Ableitung):

Aber du darfst dich nicht darauf verlassen.

Die Freelance-Handbuch-Frechheit (vierte Ableitung):

Wenn Du dich nicht darauf verläßt und das erklärte Problem selbständig zu bewältigen versuchst, dann lag der Fehler woanders.

Das Globalink-Gesetz:

Dein Übersetzungsprogramm unter Windows wird immer bestrebt sein,

- die falschen Originaldateien zu bearbeiten
- die Ergebnisse in das falsche Fenster auszugeben
- mit falschen Ergebnissen die richtige Originaldatei zu überschreiben.

In jedem anderen Fall wird Dein Übersetzungsprogramm aus der Liste der möglichen Wortpaare die jeweils so geringfügig falsche Übersetzung wählen, daß Dir der Fehler nicht auffällt, aber beim Empfänger die maximal mögliche Heiterkeit beziehungsweise das größtmögliche Mißverständnis auslöst.

Das Harvard-Graphics-Gebot:

1. Egal, was für Daten Du aufbereiten mußt, Dein Programm wird die einzige dafür sinnvolle grafische Darstellungsart nicht beherrschen.
2. Ist die entsprechende Darstellungsart verfügbar, hast Du die gesamte Tabelle dafür falsch aufgebaut.
3. Du wirst 300 Einzelwerte nur als Balkengrafik und eine zeitliche Entwicklung ausschließlich als Tortendiagramm darstellen können.

Die GWS-Gaukeleien:

1. Wenn Du eine Grafik von einem Programm in ein zweites übertragen willst, werden wahlweise Quell- oder Zielprogramm keinen passenden Im- oder Exportfilter besitzen.
2. Eines der beiden Programme wird Daten auch nicht über die Zwischenablage austauschen können.
3. Um die Grafik zu konvertieren, wirst Du stets drei Grafikkonvertierungsprogramme benötigen: Eines, das das Ausgangsformat lesen kann, eines, das das Zielformat schreiben kann und ein drittes, das die Daten für die beiden aufbereiten kann.

4. Du wirst stets nur zwei von den drei Programmen besitzen.

Die Hexedit-Heimsuchung:

Die einzige Stelle, auf die Dein Sektoreditierprogramm ohne abzustürzen zugreifen kann, sind die Sektoren, auf der die permanente Swapdatei von Windows liegt. In diesem Fall wird Windows just in dem Moment auf die geänderten Sektoren der Swapdatei zugreifen und abstürzen, sobald die Chance möglichst hoch ist, daß Du unwiederbringlich Daten verlierst.

Die Hyparchiv-Hypothek:

Jedes Backup- und Archivierungsprogramm unter Windows ist bestrebt, die jeweils älteste Version Deiner Daten- und die jeweils jüngste Version Deiner Programmdateien zu archivieren, um einerseits möglichst viel Deiner Arbeit zu vernichten und andererseits die Gefahr eines Virenbefalls so groß wie möglich zu machen.

Der IconAuthor-Idealismus:

Auf dem Arbeitsbildschirm eines Autorensystems, das Programmteile als Icons bearbeitet, ist immer ein Icon fehlerhaft beschriftet. Dieses Icon sollte gelesen werden als »Wer hier anklickt, sollte alle Hoffnung fahren lassen«.

Arbeitsanweisungen zum IconAuthor-Idealismus:

1. Versuche nie, dieses Icon zu finden. Es wird Dir nicht gelingen.

2. Wenn es Dir doch gelingt und Du den entsprechenden Programmteil verbesserst, wird ein anderes Icon die Fehler übernehmen und optimieren.

Globale Verallgemeinerung des IconAuthor-Idealismus:

Generell sollte jedes Windows-Icon gelesen werden als »Wer hier anklickt, sollte alle Hoffnung fahren lassen«.

Die Illustrator-Instabilität:

Der Zusammenhalt innerhalb kombinierter Grafikobjekte bei einem Windows-Grafikprogramm ist nur so lange stabil, wie Du es in Ruhe läßt. Sobald Du versuchst, mit ihm etwas zu tun – es beispielsweise zu verschieben, umzufärben oder anzusehen – werden die einzelnen Objektteile sich gegeneinander so verschieben, daß es Dir unmöglich ist, sie jemals wieder passend zueinander anzuordnen.

Ausnahme von der Illustrator-Instabilität:

Nicht zusammengehörende Teile wirst Du nie wieder voneinander trennen können.

Die Image-In-Idiotie:

Das, was Du bei Deinem OCR-Programm im Texterkennungsfenster siehst, wird sich immer von dem unterscheiden, was das Programm tatsächlich tut.

Konsequente Fortführung der Image-In-Idiotie:

Erkennungsfehler werden sich stets hinter den Rollbalken verstecken, falsche Textzeilen werden stets außerhalb des Texterkennungsfensters positioniert.

Die Image-Studio-Illusion:

Dein Windows-Bildverarbeitungsprogramm wird nicht in der Lage sein, aus einer verschwommenen, unscharfen Vorlage etwas anderes zu machen als etwas Verschwommenes und Unscharfes.

Erste Fortsetzung der Image-Studio-Illusion:

Dein Windows-Bildverarbeitungsprogramm wird auch nicht in der Lage sein, aus einer vorzüglichen Vorlage etwas anderes zu machen als etwas Verschwommenes und Unscharfes

Zweite Fortsetzung der Image-Studio-Illusion:

Dein Windows-Bildverarbeitungsprogramm wird auch nicht in der Lage sein, aus irgendetwas anderem etwas anderes zu machen als etwas Verschwommenes und Unscharfes.

Der Ingres-Impetus:

Ein Programm, mit dem man Windows-Programme unter einer grafischen Unix-Oberfläche ablaufen lassen kann, wird die Nachteile beider vereinigen und die Vorteile beider unterbinden.

Die Instant-Orgchart-Ignoranz:

1. Die einzige Managementposition, die Deine Organigramm-Software stets unterschlagen wird, ist die Deines Chefs.
2. Er wird es merken.
3. Er wird Dich dafür verantwortlich machen.

Der Jetform-Jokus:

Formularmanagement unter Windows ist der gelungene Versuch, bürokratische Fehler mit Windowsfehlern zu kombinieren, um bürokratische Windowsfehler zu bekommen, die Deine Arbeit behindern und die Bürokratie optimieren.

Die Knowledge-Pro-Konkretisierung:

Interaktive Multimedia-Anwendungen unter DOS bedeuten, daß das Programm von alleine abstürzt. Interaktive Multimedia-Anwendungen unter Windows hingegen stürzen erst nach einer beliebigen Benutzeraktion ab.

Der Lantastik-Lapsus:

Wenn Du in Deinem Netzwerk auf unterschiedlichen Arbeitsplätzen gleichzeitig DOS- und Windowsversionen Deiner Netzwerksoftware betreibst, wird Windows spätestens bei einem Chat-Versuch zwischen einer DOS- und einer Windows-Version abstürzen.

Erweiterung des Lantastik-Lapsus:

Packst Du Deine Netzwerksoftware in den Auto-Ordner, bootet der Rechner beim Windows-Start.

Die Legacy-Lotterie:

1. Wenn du ein Wort löschen willst, verschwindet garantiert die ganze Zeile.
2. Wenn Du eine Zeile löschen willst, verschwindet ein ganzer Absatz.

3. Wenn Du einen Absatz löschen willst, verschwindet der ganze Text.
5. Wenn Du den ganzen Text löschen willst, passiert gar nichts.

Schlußfolgerung aus der Legacy-Lotterie:

Keiner dieser Vorgänge ist rückholbar.

Die Lotus-123-Lüge:

Jedes Tabellenkalkulationsprogramm, das ein richtiges und angenehmes Ergebnis hervorbringt, lügt.

Die Mathcad-Methode:

Die Formel, die Dein Formelprogramm akzeptiert, paßt nicht zum Problem.

Die Mathematica-Mühe:

Wenn Du Dein Windows-Bildbearbeitungsprogramm endlich marktreif hast, wird die einzige Funktion, die bei Anwendern und Journalisten gerade en vogue ist, fehlen.

Der Meta-Design-Moloch:

1. Du wirst nie ein Organigramm komplett ins Arbeitsfenster bekommen.
2. Du wirst es auch nicht komplett auf eine Druckseite bekommen.
3. Du wirst es auch nicht auf mehrere Druckseiten bekommen.
4. Du wirst es überhaupt nicht bekommen.

Verdeutlichung des Meta-Design-Molochs:

Der entscheidende Fehler ist stets außerhalb des Bildschirms.

Die Mikro-Auftrag-Maximen:

1. Wenn Du aufgrund knapper Lagerbestände auf Dein Lagerhaltungsprogramm angewiesen bist, wird es vollautomatisch und großzügig Bestände nach oben runden.
2. Wenn Du aufgrund knapper Kapitaldecke auf Dein Lagerhaltungsprogramm angewiesen bist, wird es vollautomatisch und großzügig Bestände nach unten runden.
3. Wenn Du aufgrund voller Auftragsbücher auf Dein Lagerhaltungsprogramm angewiesen bist, wird es vollautomatisch und großzügig Bestände zusammengehörender Teile nach unten und nach oben runden

Die MPBChiffre-Misere:

1. Wenn eine Information vertraulich ist, bleibt Sie aus Versehen als Datei auf der Diskette, die an die Konkurrenz geht.
2. Die Einladungstexte zur hausinternen Gartenparty hingegen werden stets verschlüsselt.
3. Bei wirklich wichtigen Dateien, von denen keine Sicherheitskopie vorhanden ist, wird das Paßwort vergessen.

4. Ist eine Sicherheitskopie vorhanden, so ist auch sie verschlüsselt.
5. In jedem anderen Fall wurde die Datei mit einer Programmversion verschlüsselt, die nur auf einer älteren Windowsversion läuft.

Die MS-Project-Mühsal:

Der Unterschied zwischen einem normalen Projekt und einem mit einem Windows-Projektmanagement verwalteten Projekt ist, daß ein normales Projekt immer länger als geplant dauert. Ein mit einem Windows-Projektmanagement verwaltetes Projekt dauert viel länger.

Persönliche Erweiterung der MS-Project-Mühsal:

Das mit Deinem Windows-Projektmanagement verwaltete Projekt dauert sehr viel länger.

Der Myfriend-Mystizismus:

Der Unterschied zwischen dem Dateimanager von Windows und dem Dateimanager-Programm, das Du separat gekauft hast ist der, daß Du beide nicht vernünftig bedienen kannst.

Die Nortons-Desktop-Neurosen:

1. Jegliche Windows-Software wird stets Deine CONFIG.SYS-, AUTOEXEC.BAT-, WIN.INI- und SYSTEM.INI-Datei so durcheinanderwirbeln, bis Du sie nicht wiedererkenntst.
2. Du wirst dabei ebenso exakte wie unexakte wie unverständliche Installationshinweise erhalten (»Ihre Pfadlänge übersteigt etwa 9832 Byte«).

Die Object-View-Observation:

1. Daten verändern sich zum Schlechten, wenn Du Sie beobachtest.
2. Daten verändern sich auch zum Schlechten, wenn Du Sie nicht beobachtest.
3. Datenverschlechterungen in mehreren Fenstern veranstalten Wettrennen.

Das Objectvision-Obligat:

1. Je mehr Entwicklungsarbeit Du in ein hauseigenes Windowsprogramm steckst, umso mehr Leute arbeiten in Deiner Firma ausschließlich mit DOS und umso weniger Leute können es benutzen.
2. Je weniger Entwicklungsarbeit Du in ein Programm stecken konntest, umso mehr Leute werden es verwenden und Dich für die darin zahlreich entdeckten Fehler verantwortlich machen.

Erste Folgerung aus dem Objectvision-Obligat:

Je mehr Fehler in Deinen Programmen stecken, umso schlechter ist der Support beim Hersteller des Entwicklungspakets.

Zweite Folgerung aus dem Objectvision-Obligat:

Nur das Programm, das niemand braucht, läuft fehlerfrei.

Die OCP-Mail-Obszönität:

Die einzige Zeit des Jahres, für die sich die Anschaffung Deines E-Mail-Programms gelohnt hat, ist die Zeit Deines Urlaubs.

Begründung der OCP-Mail-Obszönität:

Du wirst nämlich nach Deiner Rückkehr mehrere hundert Nachrichten in Deinem E-Mail-Postfach vorfinden.

Verfeinerungen der OCP-Mail-Obszönität:

1. Wenn Du sie alle ungelesen löschst, wirst Du eine unwiederbringliche und wichtige Information mitlöschten.
2. Wenn Du sie alle liest, ist nichts Wichtiges dabei.

Das Omnipage-Odium:

Wenn ein Texterkennungsprogramm eine Erkennungsrate von 99,95 Prozent hat, bedeutet das nur, dass Deine Texte stets zu den 0,05 Prozent gehören, die Du zu 100 Prozent neu tippen mußt.

Das Omnis-VII-Ondit:

Applikationen, die Du mit einem Datenbankentwicklungssystem entwickelt hast, das laut Programmbeschreibung kompatiblen Quellcode für mehrere Plattformen produziert, sind nur dann nicht kompatibel, wenn Du tatsächlich für mehrere Plattformen entwickelst.

Das One-for-all-Omen:

1. Je mehr Funktionen auf der Rückseite der Verpackung angepriesen werden, umso weniger wird das Programm beherrschen.
2. Wenn draufsteht »für alle Anwendungen« ist es für niemanden brauchbar.

3. Alle Funktionen, wegen denen Du das Programmpaket gekauft hast, müssen als Spezialmodule extra bezahlt werden.

Die Outline-Obstruktionen:

1. Softfonts haben ihren Namen von ihrem ständigen erfolgreichen Versuch, auf Bildschirm und Drucker auseinanderzufließen und ihre Form und Größe zu ändern.
2. Dein Softfontprogramm wird sie stets dabei so unterstützen, das der schlechtestdenkbare Output dabei herauskommt.

Die Pagemaker-Postulate:

1. Wenn Du einen Text in Deinem DTP-Programm positionierst, paßt die letzte Zeile nicht mehr auf die Seite.
2. Paßt Du eine Grafik in einen Text ein, paßt die letzte Zeile nicht mehr auf die Seite.
3. Benutzt Du Initialen, paßt die letzte Zeile nicht mehr auf die Seite.
4. Formatierst Du den Text um die kleinstmögliche Einheit kleiner, oder nimmst Du eine der Aktionen zurück, ist der Text zu kurz.
5. Jede weitere Aktion wird den Überlauf auf die nächste Seite um mindestens eine Zeile verlängern.

Die »Paradox für Windows«-Peinigungen:

1. Egal, wie die Verhältnisse in Deiner Datenbank aussehen oder sich verändern – alles dauert immer

- länger, als nach Deiner pessimistischsten Prognose.
2. Egal wie groß Du eine Maske anlegst, sie hat immer zuwenig Felder.
 3. Wenn dennoch genug Felder definiert sind, sind diese zu kurz.
 4. Wenn Du Dich darüber ärgerst, daß Du in Deinem Datenverwaltungsprogramm 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 überflüssig gewordenen Daten wegwerfen.
 5. Ein Dateiverwaltungsprogramm zerstört nur den Datensatz, den Du dringend benötigst.
 6. Wenn kein Backup existiert, werden alle Datensätze zerstört.
 7. Existiert ein Backup, ist der Datensatz auch dort verschwunden.
 8. Wenn Du eine Datenbank mit »n« Datensätzen anlegen willst, kann das einzige verfügbare Dateiverwaltungsprogramm bestenfalls »n-1« Datensätze verwalten.
 9. Das wird Dir von dem Programm allerdings erst dann mitgeteilt, wenn Du »n-2« Datensätze bereits eingegeben hast.
 10. 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 vorsortierte Datensätze in einer annehmbaren Zeit heraus.

11. Egal wieviele Adressen Du in einer Datenbank gespeichert hast: Diejenige, die Du suchst, ist noch nicht eingegeben, sondern steht auf einem Schmierzettel.

Tonis Erweiterungen der elften »Paradox für Windows«-Peinigung:

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.

Die PC-Freiberufler-Prämissen:

1. Wenn Du Verwaltung und Abrechnung weiterhin mit einem DOS-Programm machst, um Zeit zu sparen, wirst Du doppelt so viel Zeit für Verwaltung und Abrechnung brauchen wie Dein härtester Konkurrent, der sie mit einem Windows-Programm erledigt.
2. Wenn Du Deine gesamte Verwaltung und Abrechnung auf Windows umstellst, um Zeit zu sparen, wirst Du doppelt so viel Zeit für Verwaltung und Ab-

rechnung brauchen wie Dein härtester Konkurrent, der sie mit einem DOS-Programm erledigt.

3. Wenn Du Deine gesamte Verwaltung und Abrechnung wieder auf DOS umstellst, wirst Du doppelt so viel Zeit für Verwaltung und Abrechnung brauchen wie vorher.

Das PC-Tools-Phänomen:

Das einzige Hilfsprogramm in Deiner teuren Windows-Tool-Sammlung, das Du brauchst, läuft nur unter DOS.

Die PerForm-Perfidie:

Du wirst nie mit Deiner Formularsoftware ein Formular anlegen können, das

- Deinem Chef gefällt
- mit dem bürokratischen Ablauf im Haus harmoniert
- auf der Hardware im Haus zu verarbeiten ist
- mit Windows-Software zu verarbeiten ist
- sinnvoll ist.

Das Photo-Styler-Paradoxon:

1. Egal wie die Form aussieht, die Du mit einem Raster oder einem Muster füllen willst: Das Programm findet immer Mittel und Wege, das gesamte Zeichenfenster einzuschwärzen.
2. Dieser Vorgang ist nicht rückholbar.
3. Hast Du alle Linien und Übergänge doppelt und dreifach darauf untersucht, daß ein Füllmuster nicht

aus der zu füllenden Form entwischen kann, wirst Du mit dem Füllwerkzeug daneben klicken.

Das Picture-Publisher-Pejorativ:

1. Die einzigen Datenformate, die Dein Bildbearbeitungsprogramm beherrscht, kannst Du mit Deinem Animationsprogramm nicht lesen.
2. Du wirst Sie auch mit Deinem Desktop-Publishing-, Deinem Grafik- und Deinem Textverarbeitungsprogramm nicht lesen können.
3. Ein Konvertierungstool wird zwar die Datenformate Deines Bildbearbeitungsprogramms neben drei Dutzend anderer lesen und schreiben können, nicht jedoch die Formate, die Dein Animations-, Dein Desktop-Publishing-, Dein Grafik- und Dein Textverarbeitungsprogramm beherrschen.
4. Kopieren über die Zwischenablage ist hoffnungslos.

Die Pinboard-Pittoreske:

Wenn Du Deine Pinboard-Software nicht benutzt, steht die Ausschreibung für den Traumjob Deines Lebens dran.

Wenn Du sie benutzt, ist das elektronische Pinboard leer.

Die Plot&View-Plumpheit:

Der einzige Weg, einen Plotter und einen Drucker aus Windows heraus anzusteuern ist, Deine Kurven zu drucken und die Texte zu plotten.

Die Powerpoint-Prädestination:

Vergiß alles, was der Verkäufer Dir über VDI-Treiber und laserdruckfähige Overheadfolien erzählt hat.

Praxisnahe Begründung der Powerpoint-Prädestination

1. Deine Tabellenkalkulation kann die Daten nicht so aufbereiten, daß es Dein Businessgrafikpaket versteht.
2. Dein Businessgrafikpaket wird keinen Windows-Treiber für Dein Ausgabegerät besitzen.
3. Ein Diabelichter ist die teuerste Möglichkeit, festzustellen, daß in der Grafik noch ein Fehler ist. Ansonsten dient er dazu, statt der Grafik die Steuerungsdatei in ASCII auf ein Dia auszugeben.
4. Wenn alles funktioniert, gibt es im Präsentationsraum keinen Strom.
5. Wenn Du Dich vorher versicherst, daß es im Präsentationsraum Strom gibt, fällt die Veranstaltung aus oder Du vergißt das alles entscheidende Kabel.
6. In jedem anderen Fall wird sich Windows während der Präsentation mit einem UAE verabschieden.
7. Klappt alles, interessiert sich niemand für das von Dir präsentierte Ergebnis.

Das Professional-Write-Problem:

Ein Ausdruck ist nie vollkommen.

Beweis des Professional-Write-Problems:

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

Die Paintbrush-Prädomination:

1. Die Funktionen, die Du zur Bearbeitung einer großen Bitmap brauchst, wird das von Dir verwendete Zeichenprogramm in der Zoom-Darstellung nicht zur Verfügung stellen.
2. Die Funktionen, die Du zur exakten Bearbeitung einer Bitmap brauchst, wird das von Dir verwendete Zeichenprogramm in der Vergrößerungs-Darstellung nicht zur Verfügung stellen.

Die QuarkXPress-Qualen:

1. Klickst Du auf einer layouteten Seite auf einen Kasten, um ihn zu verschieben, wirst Du den ihn umgebenden Text bewegen.
2. Klickst Du auf den Text, um ihn zu verschieben, wirst du lediglich den Kasten bewegen.
3. Willst Du beides verschieben, bewegen sich die Spaltenhilfslinien.
4. Keiner dieser Vorgänge ist rückholbar.
5. Ist einer der Vorgänge rückholbar, so wird sich irgendetwas um einen Punkt verschieben und das gesamte Layout ruinieren.

Die Recognita-Regel:

Ein Texterkennungsprogramm wird nur solange fehlerfrei arbeiten, wie Du den Lernmodus eingeschaltet hast oder auf den Bildschirm schaust. Sobald es eigenständig arbeitet, wird es an die Stellen Erkennungsfehler einbauen, die

- a) möglichst schwierig zu finden sind und die
- b) möglichst schwierig zu beseitigen sind.

Die Ressource-Kit-Randerscheinung:

Wenn Du Dich endlich entschlossen hast, für Deine Mitarbeiter das Windows-Ressource-Kit anzuschaffen, damit sie mit Windows effektiver arbeiten können, kommen Sie wegen der beiliegenden Spiele überhaupt nicht mehr zum Arbeiten.

Der Softtype-Schicksalsschlag:

Teuer gekaufte Fontbibliotheken werden wahlweise

- von Deinem Programm
- von Deinem Drucker oder
- von Deinem Monitor

nicht korrekt verarbeitet, in der Regel weder vom einen noch vom anderen, aber auf unterschiedliche Arten und Weisen falsch.

Die SQLWindows-Schädigung:

1. Vergiß alles, was Du jemals über einheitliche Abfragen bei SQL-Datenbanken gehört hast.
2. Nicht einmal Deine SQL-Datenbank ist zu sich selber einheitlich.

3. Du wirst alles finden, nur nicht das, was Du suchst.
4. Alles, was Du gefunden hast, wird Windows auf dem Weg zwischen Datenbank und Ausgabefenster wegwerfen.

Die Superbase-Schlußfolgerungen:

1. Wenn Du in Deiner Bilddatenbank nach einem Bild suchst, wirst Du Dich zwar noch an das Bild, jedoch nie an das Stichwort erinnern, mit dem Du es wiederfinden kannst.
2. Die Bilder im Bildübersichtsfenster sind stets so klein, daß Du auf das falsche Bild klickst – mehrmals.

Das Technobox-Drafter-Trauerspiel:

Egal, welches von zwei nebeneinanderstehenden Grafikobjekten Du anklicken willst – Du erwischst immer das andere.

Die Time-Manager-Täuschung:

1. Ein Zeitverwaltungsprogramm unter Windows hilft Dir, täglich eine Stunde zu sparen.
2. Zur Bedienung des Programms brauchst Du täglich zwei Stunden.
3. Du hast also pro Tag drei Stunden weniger Zeit.
4. Der letzte Satz ist unabhängig davon, ob Du Dein Zeitverwaltungsprogramm tatsächlich einsetzt oder nicht.

Umsetzung der Time-Manager-Täuschung:

Bis Du Windows gestartet hast, ist der wichtigste Termin am Tag bereits vorbei.

Die Titan-Tyrannie:

Die einzigen Suchworte, die Deine Windows-Volltextdatenbank zuverlässig indiziert und findet, sind die, die Du von der Indizierung ausgenommen hast, wie »der«, »eine« oder »und«.

Die Toolbook-Tragik:

Dein Autorensystem wird nicht in der Lage sein, externe Text- oder Bilddateien zu verwalten.

Folgerungen aus der Toolbook-Tragik:

Du wirst Fehler in Texten und Bildern erst bemerken, wenn Du Sie importiert hast.

Die TypeAlign-Technophobie:

1. Die französische Übersetzung von WYSIWYG ist »Honni soit qui mal y pense« oder kurz »HSQMYP«.
2. Die englische Übersetzung von WYSIWYG ist »What You see is what you might get« oder kurz »WYSIWYMG«.
3. Sobald bei Dir irgendwelche Hoffnungen auftauchen, Du würdest Dich mit der Arbeit einem Ende nähern, lautet die englische Übersetzung von »WYSIWYG« »What You see is what you never get« oder kurz »WYSIWYNG«.
4. Das deutsche Kürzel von »WYSIWYG« ist »GNAGNAGNA«.

Die Uninstall-Unabänderlichkeit:

Ein Windows-Programm, das Du einmal installiert hast, wird seine Daten so über die Windows-Verzeichnisse aller Deiner Festplatten verteilen, daß Du sie bei der Deinstallation nie wieder alle loswirst.

Erste Konkretisierung der Uninstall-Unabänderlichkeit:

Daran ändert auch ein Uninstall-Programm nichts.

Zweite Konkretisierung der Uninstall-Unabänderlichkeit:

Die Dateien, die Du beim Deinstallationsprozess vergißt, werden diejenigen sein, die den meisten Speicherplatz verbrauchen.

Dritte Konkretisierung der Uninstall-Unabänderlichkeit:

Die Dateien, von denen Du beim Deinstallieren nicht sicher bist, ob Du sie löschen kannst, weil Du nicht weißt, zu welchem Programm sie gehören, verbrauchen den meisten Speicherplatz und richten den größten Schaden an, wenn Du sie doch löschst.

Die Ventura-Verschärfung (auch »allgemeines Puzzle-Gesetz« genannt):

Dein Desktop-Publishing-Programm wird die einzelnen Kapitel Deines Dokuments dann und nur dann in der falschen Reihenfolge aneinanderhängen sowie Inhaltsverzeichnis, Index und Seitenzahlen durcheinanderbringen, wenn Du keine Chance mehr hast, den Fehler zu bemerken, geschweige denn, ihn zu beheben.

Das VisiNet-Vexierspiel:

Jede Windows-Netzwerkinstallation besitzt drei Aspekte. Den offiziellen, das ist der, wie es im Handbuch beschrieben ist, den administrativen, das ist der, wie es der Netzwerkadministrator verwaltet und den realen. Das ist der, wie es von den Angestellten installiert wurde und betrieben wird.

Das Vivaldi-Verärgerung:

Jedes windowsfähige Btx-Programm wird die Unzulänglichkeiten von Bildschirmtext so mit den Windowsfehlern kombinieren, daß Du im Btx-Fenster statt CEPT- nur ASCII-Zeichen siehst und außerhalb jedes ASCII-Zeichen nach CEPT konvertiert ist.

Die WibuKey-Weissagung:

Wenn Du mal vergißt zu speichern, stürzt Dein Programm ab, wenn Du versuchst zu drucken.

Der WinBasic-Witz:

Du kannst ohne großen Aufwand ein Basic-Programm schreiben, das unter Windows lauffähig ist.

Verallgemeinerung des WinBasic-Witzes:

Du kannst ohne großen Aufwand in irgendeiner Programmiersprache ein Programm schreiben, das unter Windows lauffähig ist.

Das WinBase-Windei:

Die einzigen Adressen, die Dein Datenbankprogramm findet, sind diejenigen, die falsch sind.

Die WinCad-Watsche:

Gegeben sei das von Dir gekaufte CAD-Programm und das Anwendungsgebiet, für das Du das Programm einsetzen willst. Ausgehend von den Faktoren »Funktionen«, »Ausgabegerät« und »Symbolbibliothek« lassen sich folgende nichtrevidierbaren Aussagen ableiten:

1. Verfügt Dein CAD-Programm über die von Dir dringend benötigten Funktionen, dann unterstützt der Windows-Treiber für das Ausgabegerät sie nicht.
2. Unterstützt der Treiber die Funktionen, dann ist für das Anwendungsgebiet, für das Du das Programm einsetzen willst, die entsprechende Symbolbibliothek nicht verfügbar.
3. Ist die benötigte Symbolbibliothek verfügbar, dann fehlen Dir dringend benötigte Funktionen.

Der WinChecker-Wink:

Die häufigste Fehlerursache in Deinen Texten wird Dein Rechtschreibkorrekturprogramm nicht finden. Wenn Du beispielsweise öfters Groß- und Kleinschreibung verwechselt, wird es nicht in der Lage sein, diese Fehler zu bemerken.

Der WindowBase-Wust:

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, stehen das, was man vorne hineinsteckt (also die Datensätze), das was hinten herauskommt (also die sortierten Daten-

sätze, Listen oder ähnliches) und die Zeit dazwischen in keinem Verhältnis.

Das WinBack-Wimmern:

1. Ein Backup braucht immer eine Diskette mehr, als Du vorrätig hast.
2. Dein Backup-Programm wird in dem Moment versagen, wo Du es benötigst.
3. Das Backup-Programm wird bei seinem Versagen mit der zerstörten Dateiversion die einzige noch vorhandene Sicherheitskopie überschreiben.
4. Wenn Du das Backup zurückspielen willst, wirst Du feststellen, daß Dein Backupprogramm die Sicherheitskopien nicht mehr lesen kann.
5. Lesefehler treten ausschließlich bei der Datei auf, die Du noch unbedingt brauchst und von der Du keine Kopie hast.

Werners weiterführendes WinBack-Wimmern:

1. Erst wenn Du bei Deinem Formatierprogramm auf »Ja« in der Alertbox »Sind Sie sicher, daß Sie diese Diskette formatieren wollen« geklickt hast, fällt Dir ein, daß Du Dir ganz und gar nicht sicher bist.
2. Wenn Du danach die Diskette kontrollierst, bist Du Dir sicher: Du hast soeben Deine wichtigste Datei gelöscht.

Die Winfax-Wiederholung:

Wenn du ein Fax losschickst, das aus mehreren Seiten besteht, von der Du aber eigentlich nur eine übermitteln

willst, wird dies die einzige sein, die nicht mit übertragen wird.

Der WinFibu-Winkelzug:

Der erste, der merkt, daß Dein Finanzbuchhaltungsprogramm Fehler hat, wird die Steuerfahndung sein.

Die WinRix-Wut:

1. Dein Grafikprogramm verfügt nur über scheußliche Schriften.
2. Ein Postscript-Druckertreiber ist nur zur Zierde da.
3. Daß Dein Malprogramm über einen Postscript-Treiber verfügt, heißt noch lange nicht, daß es auch EPS-Dateien lesen kann oder Schriften ohne Pixel-treppen ausdruckt.

Erste Erweiterung der WinRix-Wut:

Jedes Grafikprogramm wird sein neues Bildformat als Standard ausgeben und kann auch nur dieses lesen und schreiben.

Zweite Erweiterung der WinRix-Wut:

Kaufst Du Dir ein neues Grafikprogramm, wird dieses alles können – nur nicht das Bildformat des alten lesen.

Die TIFF-Steigerung der WinRix-Wut

Wenn Dein Grafikprogramm dennoch ein gängiges Bildformat lesen kann, dann nur in seiner ungebräuchlichsten Form. Kann es beispielsweise TIF-Dateien lesen, dann kann es nur das ungebräuchliche unkomprimierte TIF-Format lesen.

Der Word-Wahn:

Ein Textverarbeitungsprogramm und eine grafische Benutzeroberfläche vom selben Hersteller vertragen sich miteinander.

Der vergebliche WordPerfect-Wunsch:

Kann Windows nicht einmal dann abstürzen, wenn ich meinen Text gerade gespeichert habe?

X-Touch-Xenologie:

Ein Microsoft-Maus-kompatibler Touchscreen-Treiber wird nur dann kompatibel sein, wenn Du Monitor und Touchscreen über den Schreibtisch rollst.

Das Y-Gesetz:

In einer Aufzählung von »A« bis »Z« fällt Dir zu mindestens einem Buchstaben nichts ein.

Die ZIP-Zynismen:

1. Wenn Du aus einer öffentlichen Mailbox ein Windows-Hilfsprogramm downlädst, ist es gepackt und Du hast den entsprechenden Entpacker nicht.
2. Der Entpacker ist in der Mailbox nicht abrufbar.
3. Ist er doch in der Mailbox abrufbar, ist er mindestens zehnmal länger, als das Windows-Tool, für das Du ihn brauchst.

Konsequente Fortsetzung der ZIP-Zynismen:

1. Anschließend läuft das Windows-Hilfsprogramm nicht.
2. Läuft es doch, dann ist keine Anleitung dabei.

3. Ist doch eine README-Datei dabei, dann löst sie Dein Problem mit dem Tool nicht.
4. Hast Du das Problem mit dem Tool gelöst, dann wird das Tool Dein Windows-Problem nicht beheben.

Der ZSoft-Typepaket-Zorn:

Der Schriftschnitt, weswegen Du Dir ein Schriftenpaket für Dein Windows-Programm gekauft hast

- ist in dem Paket nicht enthalten
- ist nur in der falschen Größe enthalten
- wird von Deinem Windows-Programm nicht unterstützt
- wird von Deinem Drucker nicht unterstützt.



Dein Dateiverwaltungsprogramm wird stets in der Lage sein, Feldlängen willkürlich zu verkürzen und dabei die nun überflüssig gewordenenen Daten wegzuwerfen.

F. Die Optimierung der in Programmzeilen gegossenen in Silizium geätzten Heimtücke: Windows und die Hardware

Während Software bekanntlich der geglückte Versuch ist, die Fehler der Hardware vorauszuahnen, vorhandene Irrtümer zu optimieren, zu speichern und mit immer höherer Geschwindigkeit zu produzieren ist Hardware der geglückte Versuch, die Fehler der Software vorauszuahnen, vorhandene Irrtümer zu optimieren, zu speichern und mit immer höherer Geschwindigkeit zu produzieren. Hier gilt

Swansons Prinzip des prelusiven Programmierens:

Jede gegebene Hardware wird lernen, die Fehler der Software zu emulieren und zwar binnen einer Stunde nach seiner Beseitigung.

Während unter DOS verschiedene Softwarepakete mit derselben Hardware auf immer unterschiedlichere Arten und Weisen Fehler und Pannen produzierten, so ist es dank der genormten Windows-Treiber nun möglich, daß verschiedene Softwarepakete mit derselben Hardware auf immer unterschiedlichere Arten und Weisen Fehler und Pannen produzieren.

Für Hard- und Software gemeinsam gilt das

Doppelgesetz von komplexen Systemen:

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

Erste Windows-Erweiterung des Doppelgesetzes von komplexen Systemen:

Komplexe Systeme produzieren zusammen mit einer neuen grafischen Benutzeroberfläche komplexe Fehler.

Zweite Windows-Erweiterung des Doppelgesetzes von der komplexen Hardware:

Einfache Systeme produzieren zusammen mit einer neuen grafischen Benutzeroberfläche komplexe Fehler.

Dritte Windows-Erweiterung des Doppelgesetzes von komplexen Systemen:

Jedes System produziert zusammen mit einer neuen grafischen Benutzeroberfläche neue und alte Fehler.

Vierte Windows-Erweiterung des Doppelgesetzes von komplexen Systemen:

Jedes System produziert zusammen mit irgendeiner grafischen Benutzeroberfläche neue und alte Fehler.

Schlußfolgerungen:

1. Grafisch unterstützte Systeme neigen dazu, ihre eigene Funktion zu behindern.
2. Grafisch unterstützte Systeme neigen dazu, ihren Benutzer zu behindern.

3. Grafisch unterstützte Systeme neigen zum Wachstum und werden dadurch anmaßend.

Hardware setzt sich zusammen aus 5 Prozent Fehlfunktionen sowie aus 95 Prozent in Silizium geätzter Heimtücke. Windows optimiert dieses Verhältnis einerseits soweit, daß Hardware, die unter Windows eingesetzt wird, aus 100 Prozent Fehlfunktionen sowie aus 100 Prozent in Silizium geätzter Heimtücke besteht. Andererseits besteht Windows selbst (sowie zusätzlich jedes Anwendungsprogramm) aus 100 Prozent Fehlfunktionen sowie aus 100 Prozent in Programmzeilen gegossener Heimtücke, was damit für jeden Anwender zu einer insgesamt mindestens sechshundertprozentigen Sicherheit führt, daß irgendetwas schief läuft. Diese Leistungsfähigkeit eines Windows-Systems definieren wir nach einer einfachen Formel:

Berechnung der Leistungsfähigkeit eines Windows-Systems:

$$L = I(H) + I(W) + I(P) + G + A$$

Also berechnet sich die Leistungsfähigkeit eines Windows-Systems aus der Summe der Intelligenz (sprich: der Anzahl der fest eingebauten Fehler) von Hardware, von Windows sowie dem eingesetzten Windowsprogramm. Dazu kommt die Geschwindigkeit G, mit der das Gesamtsystem eine größtmögliche Zahl von Katastrophe produziert sowie der Antwortzeit A. Das ist die Zeit, die das Windows-System braucht, um sich von

den Eingaben des Anwenders zu erholen und danach unter Datenverlust abzustürzen.

Diese Leistungsfähigkeit nutzt ein Windows-System aus der Sicht des Anwenders dazu, 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.

Für Programmierer und Hersteller hingegen nutzt ein Windows-System diese Leistungsfähigkeit dazu, 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.

Seit Jahren ist die Computerindustrie mit wachsendem Erfolg dabei, die Leistungsfähigkeit und Fehleranfälligkeit ihrer Computersysteme zu erhöhen. Wie bekannt, gilt allgemein

Das Assembly-Axiom:

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

Verschärfungen des Assembly-Axioms:

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.

4. Letzteres gilt insbesondere für Alert- und Diskettenboxen.

Mit der Erhöhung der Fehleranfälligkeit der Computersysteme (wobei die Einführung grafischer Benutzeroberflächen eine nicht zu unterschätzende Rolle gespielt hat) wird zugleich eine immer größere Zahl von Supportingenieuren, Reparaturbetrieben, Installationsfachleuten und Troubleshooting-Experten in Lohn und Brot gesetzt. Diesem wachsenden Markt steht auf der anderen Seite die immer geringer werdende Zahl von Firmen gegenüber, die noch ohne Computersysteme 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 ihrer Computersysteme positive Bilanzen zu erreichen.

Windows baut nun das durch den Computer optimierte Gesetz, daß alles, was schiefgehen kann, auch schiefgeht, zur ungeahnten Vollkommenheit aus. Da diese Regel sowohl für das Windows-System an sich als auch für die Beziehung zwischen Computer und Windows einerseits sowie zwischen Windows-System und dem Rest der Welt gilt, sind die Chancen groß, daß es schon in absehbarer Zeit außer Computerherstellern und Reparaturbetrieben keine anderen Firmen mehr gibt, bis auch diese aufgrund obligatorischer Computerfehler Konkurs anmelden und eine Gesellschaft übrigbleibt, in der außer Computerfehlern und Windows-Updates nichts mehr existiert.

Die für den Windows-Anwender relevante Hardware zerfällt bekanntlich in Drucker, Monitor, Eingabegeräte, sowie in dem Moment, wenn Windows ausnahmsweise beschlossen hat, nicht abzustürzen. Deswegen hat sich der Autor im folgenden seine Untersuchung murphyologischer Korrelationen zwischen Windows und der Hardware auch auf diese Peripheriegeräte beschränkt, obwohl verallgemeinernd gilt:

Allgemeines Windows-Hardwaregesetz:

Für das am dringendsten benötigte Peripheriegerät wird kein funktionsfähiger Windows-Treiber verfügbar sein.

Dem (in jeder Bedeutung des Wortes) letzten Peripheriegerät, dem Mensch, widmet er sich erst im nächsten Kapitel. Beginnen wir vielmehr mit dem Drucker, dem letzten Notnagel digitaler Heimtücke:

Die Ultima Ratio des Ausdrucks:

1. Wenn alles funktioniert hat, wird der Windows-Druckertreiber dem Drucker den Befehl zum Versagen übermitteln.
2. Wenn der Drucker doch druckt, 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.

Allgemeine Drucker-Definition:

Es gibt unter Windows vier Arten von Druckern:

- Lokale Arbeitsplatzdrucker, die Deine Druckjobs wegwerfen
- Netzwerkdrucker, die die Druckjobs der anderen drucken und Deine Jobs wegwerfen
- Netzwerkdrucker, die die Druckjobs der anderen wegwerfen und Deine Jobs wegwerfen
- Drucker, deren Treiber Deine Druckjobs wegwerfen.

Erste Konkretisierung der allgemeinen Drucker-

Definition:

Dein Windows-Druckertreiber wird dafür sorgen, daß alles ausgedruckt wird. Außer dem Wichtigen.

Zweite Konkretisierung der allgemeinen Drucker-

Definition:

Dein Tabellenkalkulationsprogramm wird zum Beispiel Zellenrahmen, Raster und Dateinamen, nicht jedoch irgendwelche Zahlen drucken.

Dritte Konkretisierung der allgemeinen Drucker-

Definition:

Dies passiert dann und nur dann, wenn Du den Ausdruck dringend benötigst und keine Zeit hast, Drucker oder Druckertreiber zu wechseln.

Windows-Erkenntnis von der Druckvorschau:

Der Ausdruck paßt auch dann nicht auf eine Seite, wenn Dein Windows-Programm Dir in der Preview nur eine Seite angezeigt hat.

Erweiterung der Windows-Erkenntnis von der Druckvorschau:

Der Ausdruck paßt auch in keinem anderen Fall auf eine Seite.

Widerlegung der Windows-Erkenntnis von der Druckvorschau:

Wenn Du einen Text auf zwei Seiten drucken willst, wird der Ausdruck auf eine Seite gedruckt.

Cornelias Diplom-Gesetz:

1. Der Drucker, auf dem du deine Diplomarbeit drucken willst, steht Dir einen Tag vor dem Abgabetermin nicht zur Verfügung.
2. Wenn es dir gelingt, einen zweiten Drucker aufzutreiben, fehlt Dir der Windows-Treiber dafür.
3. Hast Du ihn doch, dann besitzt der zur Verfügung stehende Drucker die Schriften nicht, mit denen Du Deinen Text formatiert hast.
4. Kennt er wider Erwarten die Schriften, dann geht mitten im Ausdruck wahlweise
 - der Toner aus
 - das Farbband aus
 - die Druckwalze kaputt
 - der Druckkopf kaputt
 - die Entwicklereinheit kaputt.

Das TIF-Gesetz:

1. Ein nicht behebbarer Fehler im Drucker tritt nur dann auf, wenn Du eine sehr große Datei ausdruckst, bei der Windows lange braucht, um sie in den Spooler zu schicken.
2. Wenn der Drucker die Datei anstandslos verarbeitet, hat er Papierstau, druckt auf das falsche Papier oder den falschen Schacht.
3. Dieser Fehler ist beliebig reproduzierbar, es sei denn, dein Chef fragt Dich, warum Du zum Ausdrucken so lange brauchst.

»PRTSCR«-Erweiterungen des TIF-Gesetzes:

1. Du kannst keine Bildschirm-Hardcopy drucken.
2. Wenn Du eine Bildschirm-Hardcopy doch drucken kannst, sind an der entscheidenden Stelle schwarze Streifen.
3. Fehlen die schwarzen Streifen, so sind statt dessen weiße Streifen zu sehen.
4. Gibt es weder weiße noch schwarze Streifen, so hast Du die Windows-Farben so gewählt, daß im Ausdruck nur schwarze Schrift auf schwarzem Untergrund zu sehen ist.
5. In jedem anderen Fall hast Du statt dem Fensterinhalt den ganzen Bildschirm in die Zwischenablage kopiert.
6. Oder umgekehrt.

Der mathematische Druckerpapier-Beweis:

Bei einer auszudruckenden Datei, die eine Länge von »n« Seiten aufweist, ist die zur Verfügung stehende Restmenge Druckerpapier »n-1« Blatt.

Die Zwei-Schacht-Erweiterung des mathematischen Druckerpapier-Beweises:

1. Bei einem Drucker mit zwei Schächten wird Dein Windows-Druckertreiber vollautomatisch erkennen, daß der gewünschte Schacht leer ist und auf den anderen Schacht umstellen.
2. Du wirst in diesem Fall die letzten im Haus befindlichen Bögen von Laserformularvordrucken oder teurem Briefpapier verhunzen.

Das Reimer-Syndrom (auch »gemeiner Etikettenschwindel« genannt):

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

Die Epson-Gesetze von den Leistungen eines Matrixdruckers:

1. »Near Letter Quality« bedeutet, daß ein Drucker unter Windows etwas zu Papier bringt, was so ähnlich wie Buchstaben aussieht.
2. »Letter Quality« ist die Schriftart, bei der man nur beim Hinsehen erkennt, daß sie aus 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.

Die Hewlett-Packard-Gesetze von den Leistungen eines Laserdruckers:

1. »300 dpi« bedeutet, daß jede Schrift, die kleiner ist als zehn Punkt durch den groben Toner kaum lesbar ist und jede Schrift in anderer Größe durch unsauberen Toner verschmiert ist.
2. »600 dpi« bedeutet, daß jede Schrift, die kleiner ist als neun Punkt durch den groben Toner kaum lesbar ist und jede Schrift in anderer Größe durch unsauberen Toner verschmiert ist.
3. »800 dpi« bedeutet, daß Du für die Leistung eines 600-dpi-Druckers zuviel Geld ausgegeben hast.
4. »PostScript-Kompatibel« bedeutet, daß Dein Drucker keine PostScript-Befehle beherrscht.
5. »Original PostScript« bedeutet, daß Dein Drucker nur die wichtigen PostScript-Befehle nicht beherrscht, dafür aber keine TrueType-Schriften druckt.
7. »Original TrueType« bedeutet, daß Dein Drucker weder TrueType- noch PostScript-Schriften druckt.

Die Tally-Erweiterung:

Ein PostScript-Interpreter kann gar nicht so fehlerhaft sein, als daß der Hersteller Deines Druckers ihn nicht eingebaut hätte.

Petras Druckertreiber-Analyse für Windows:

1. Dein Drucker ist bei den Windows-Treibern nicht dabei.
2. Dein Drucker ist auch bei den erweiterten Microsoft-Windows-Treibersammlungen nicht dabei.
3. Der Hersteller Deines Druckers hat die Produktion des Modells, das Du besitzt, längst eingestellt und wird auch keinen Windows-Treiber mehr entwickeln.
4. In der optimalen Emulation beherrscht Dein Drucker ausschließlich den altgriechischen Zeichensatz und druckt nur im tibetanischen Dreiecksformat von links unten nach rechts oben.
5. Ist dennoch ein Windows-Treiber für Deinen Drucker verfügbar, dann ist er kaputt.
6. Ist er vorhanden und nicht kaputt, dann arbeitet er nicht mit Deinem Anwendungsprogramm zusammen.
7. Alternativ arbeitet er nur mit der Windows-Version zusammen, für die Deine Anwendungsprogramme nicht konzipiert sind. (Er arbeitet also beispielsweise nur unter Windows 3.0 fehlerfrei, wenn es Dir endlich gelungen ist, Deine Anwendungsprogramme problemlos unter Windows 3.1 zu installieren).

Das allgemeine PostScript-Postulat:

1. Die benötigte Schrift fehlt.
2. Ist die Schrift vorhanden, fehlt der gewünschte Schriftschnitt.
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.
5. Wenn er doch druckt, druckt er die Fehlermeldung.

Das spezifische PostScript-Postulat:

1. Was der PostScript-Standard ist, definieren Drucker und der jeweilige Windows-Druckertreiber völlig unterschiedlich.
2. Was der PostScript-Standard ist, definieren auch zwei unterschiedliche Windows-Druckertreiber völlig unterschiedlich.
3. Die einzige Schrift, auf die sich Drucker und der jeweilige Windows-Druckertreiber verständigen können, wird Courier 10 pt sein.

Ausnahme des spezifischen Postscript-Postulats:

Satz 3 gilt nur, wenn Du diese Schrift nicht brauchen kannst. Willst Du Courier 10 pt einsetzen, werden Drucker und Programm sich auf die Konvertierung sämtlicher ASCII-Zeichen in die »Symbol«-Schrift einigen.

Ausnahme von der Ausnahme des spezifischen

Postscript-Postulats (auch »Regel vom ¶« genannt):

»ASCII« interpretieren DOS- und Windows-Programme aus naheliegenden Fiesheitsgründen selbstverständlich unterschiedlich, so daß Du alle für's Schreiben nötigen Umlaute von Hand konvertieren mußt.

Charles' Grundregel:

Ein Drucker wird nie so wenig Daten bekommen, als daß er nicht mindestens ein Blatt Papier mit Schrottzeichen versauen könnte.

Charles' erweiterte Regel:

Egal wieviel Schrott ein Drucker druckt – er wird dabei mindestens eine Zeile über das letzte komplett vollgeschriebene Blatt hinausdrucken.

Charles' Einzugserkenntnis:

1. Kein Traktor kann Einzelblätter, kein Einzelblatteinzug Endlospapier vernünftig einziehen.
2. Darüberhinaus ist ein Traktor ausschließlich dazu da, Endlospapier schief einzuziehen.
3. Ein Einzelblatteinzug hingegen ist ausschließlich dazu da, Einzelblätter schief einzuziehen.

Murphys Widerspruch zu Charles' Einzugserkenntnis:

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

Der ASCII/Centronics-Grundsatz von den Druckerstandards:

Das einzige, was zwischen verschiedenen Druckern standardisiert ist, sind Steckdose und Netzkabel.

Widerlegung des ASCII/Centronics-Grundsatzes:

Dein Drucker wird alternativ ausschließlich mit einem englischen Netzstecker ausgestattet sein oder nur mit 110 Volt arbeiten.

Ergänzung für Windows-Druckertreiber:

Sind zwei Drucker standardisiert, so wird sie spätestens der Windows-Druckertreiber unterschiedlich behandeln.

Folgerung:

Er wird bei jedem Drucker ein anderes falsches Ergebnis ausdrucken.

Gesetze vom phantasievollen Bildschirmadapter:

1. Ellipsen werden im Windows-Fenster als treppenförmige Eier dargestellt und ausgedruckt.
2. Treppenförmige Eier bleiben im Windows-Fenster treppenförmige Eier.
3. Eine Linie beginnt immer ein Pixel daneben.
4. Kreise sind keine Kreise.
5. Ein 10-Punkt-Raster mit 32 Grad Neigung wird spätestens auf dem Drucker zu einem schmierigen Etwas.
6. Auch jedes andere Raster wird spätestens auf dem Drucker zu einem schmieriges Etwas.

7. Auch jedes Füllmuster wird spätestens auf dem Drucker zu einem schmieriges Etwas.
8. Eine als schmieriges Etwas angelegte Fläche wird spätestens auf dem Drucker ein geometrisch exaktes, sofort als Computergrafik erkenntliches, unbrauchbares Etwas.

Ausnahmen:

1. Sollen zwei Linien ein Pixel Abstand voneinander haben, dann werden sie sich überlappen.
2. Zwei Linien, die sich überlappen sollen, werden exakt einen Pixel Abstand haben.
3. Kreise sind dann (und nur dann) Kreise, wenn Du sie als Ellipsen zeichnest. Dann wird sie der Drucker korrekt als Ellipsen ausgeben.
4. Eine Verwischfunktion erzeugt immer gleichmäßige, exakt parallele Linien.

Der Windows-Auflösungsdreisatz:

1. Eine Auflösung von 320 x 200 Bildpunkten in 256 Farben bedeutet, daß ein Kreis auf dem Bildschirm aussieht wie Stonehenge von oben nach einer Sprayer-Invasion – nur verschmiert.
2. Eine Auflösung von 640 x 480 Bildpunkten in 16 Farben bedeutet, daß ein Kreis auf dem Bildschirm aussieht wie Stonehenge von oben in einer Fehlfarben-Fotografie – nur unscharf.
3. Eine Auflösung von 1024 x 768 Bildpunkten bedeutet, daß ein Kreis auf dem Bildschirm gar nicht

aussieht. Dein Grafikprogramm wird diese Auflösung nicht unterstützen.

Postulat vom Einsatz von Tastatur, Maus und Trackball unter Windows:

1. Der Tastatureinsatz unter Windows dient dazu, mit unlogischen Tastenkombinationen aus unübersichtlichen Menüs schwerverständliche Befehle auszuwählen, die besser per Maus angewählt werden.
2. Der Mauseinsatz unter Windows dient dazu, mit unlogischen Mausbewegungen aus unübersichtlichen Menüs schwerverständliche Befehle auszuwählen, die besser per Tastatur angewählt werden
3. Hingegen dient der Trackballeinsatz unter Windows dazu, einen unbeweglichen Mauszeiger auf dem Bildschirm zu beobachten, weil Dein Trackball-Treiber entgegen Herstellerbehauptung nicht mit den von Windows unterstützten Mäusen kompatibel ist.

Das Baby-auf-dem-Schoß-Axiom (auch »Alt - F4 - Tab - Return - Axiom« genannt):

Ein Kind, das mit seinen Händen an die Tastatur gelangt, erwischt bei der ersten Berührung die einzige Tastenkombination, bei der etwas zu zerstören ist. Gibt es mehr als eine Möglichkeit, sucht es sich die Unheilvollere aus.

Eingeschränktes Baby-auf-dem-Schoß-Axiom:

Hindert man es daran, eine verhängnisvolle Tastenkombination zu erwischen, führt das zumindest zu

Mausgesteuertes Baby-auf-dem-Schoß-Axiom:

Wer denkt, daß ein Kind mit einer Maus geringeren Schaden anrichten kann, irrt.

Der Trackball/Maus-Unterschied:

1. Du hast immer soviel Platz für eine Maus auf Deinem Schreibtisch, als daß Du die Unannehmlichkeiten eines Trackball-Einsatzes in Kauf nehmen würdest.
2. Du hast immer weniger Platz auf Deinem Schreibtisch, als Deine Maus braucht.

Die grundlegende Systemsteuerungs-Erkenntnis:

Egal, wie Du in der Windows-Systemsteuerung die Maus-und Tastatur-Parameter einstellst, sie sind zum Arbeiten stets unbrauchbar.

Konkretisierungen der grundlegenden Systemsteuerungs-Erkenntnis:

1. Deine Mauszeigergeschwindigkeit ist zu hoch oder zu niedrig.
2. Stimmt Deine Mauszeigergeschwindigkeit, ist Deine Doppelklickgeschwindigkeit zu hoch oder zu niedrig.
3. Ist Deine Maus richtig eingestellt, ist Deine Tastaturwiederholrate zu niedrig oder zu hochhhhhhhhhhhhh.

Icon-Anwendung der grundlegenden Systemsteuerungs-Erkenntnis:

1. Wenn Du zu schnell auf das Icon klickst, tut sich nichts, weil Du zu schnell geklickt hast.

2. Wenn Du zu langsam auf das Icon klickst, tut sich nichts, weil Du zu langsam geklickt hast.
3. Wenn Du nur einmal auf das Icon klickst, tut sich nichts, weil Du nur einmal geklickt hast.
4. Wenn Du in der richtigen Geschwindigkeit zweimal auf das Icon klickst, tut sich nichts, weil Du nur einmal hättest klicken dürfen.
5. Ansonsten tut sich nichts, weil Windows abgestürzt ist.

Das Wurstfingerphänomen:

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

Gesetz vom »#«

1. Du wirst nie Return alleine drücken können.#
2. Wenn Du das »#« einmal brauchst, findest Du es auf Deiner Tastatur nicht.

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:

Es fällt an Deiner Maus immer die rechte Taste aus.

Ergänzung zu Gregors Seufzer:

Wenn Du in der Systemsteuerung die Maustasten austauscht, wird die linke Taste ausfallen und die rechte wieder gehen.

Drei-Tasten-Maus-Verallgemeinerung von Gregors Seufzer:

An einer Drei-Tasten-Maus wird niemals die mittlere Taste ausfallen.

Das Schnittstellenphänomen:

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



Hinter allem, was Du auf einem Windows-Bildschirm siehst, wartet eine Computerpanne auf den für Dich ungünstigsten Zeitpunkt, um unbarmherzig zuzuschlagen.

G. Die zwei Seiten der Medaille: Programmierer, Anwender und Windows als feixender Rand, auf dem sie stehenbleibt

In der murphyologischen Forschung wird generell eine systemimmanente Rivalität zwischen Anwendern und Entwicklern postuliert, zum Beispiel in

Katz's Gesetz:

Anwender und Entwickler werden erst dann vernünftig miteinander umgehen, wenn alle anderen Methoden versagt haben.

Verschärfung von Katz's Gesetz:

Anwender und Entwickler werden auch dann nicht vernünftig miteinander umgehen, wenn alle anderen Methoden versagt haben.

Seit der Einführung grafischer Benutzeroberflächen stellt sich jedoch der Unterschied zwischen Programmierer und Anwender völlig anders dar. Während der Windows-Programmierer daran scheitert, unter einem inkonsistenten Betriebssystem mit einer mangelhaften Benutzeroberfläche, fehlenden Treibern und mit logischen Irrtümern behafteten Programmen, Autorensystemen, Entwicklungssystemen und Programmiersprachen zu arbeiten und dafür viel zuwenig Geld erhält, scheitert der Anwender bei demselben Versuch,

zahlt jedoch viel zu viel dafür. So gelten für alle Beteiligten:

Brasingtons Beobachtungen:

1. Fortschrittliche Software ist es nicht.
2. Benutzerfreundliche Software ist es nicht.
3. Kontextsensitive und interaktive Hilfe ist weder noch.

Allgemeingültige Formulierung für Brasingtons Beobachtungen:

Fortschrittliche, benutzerfreundliche Software mit kontextsensitiver, interaktiver Hilfe ist ein Insiderwitz.

Bekanntlich sind sich Softwarehäuser und Anwender seit Jahren darüber einig, daß ihrer beider Leben ohne die überbezahlten Programmierer und deren Einwände über die Machbarkeit bestimmter Programmanforderungen sehr viel leichter wäre, was gemeinhin ausgedrückt wird im

Mathematischen Gesetz über die Programmierer/Durchblick-Relation:

Die Summe des eingesetzten Programmierer-Know-Hows ist negativ proportional zu der Zahl der von den Anwendern verständlichen Funktionen.

Darüberhinaus sind sich Softwarehäuser und Programmierer seit Jahren darüber einig, daß ihrer beider Leben ohne unfähige Anwender und deren unverständliche Wünsche über die Bedienbarkeit von Programmen

sehr viel leichter wäre, was nun wiederum verallgemeinert ausgedrückt wird als

Das Narrengesetz:

Es ist hoffnungslos, ein Programm narrensicher zu machen, denn die Narren sind genial.

Nun haben Softwarehäuser in diesem Widerstreit zwei Grundprämissen:

Erste Grundprämisse von Windows-Softwarehäusern:

Es ist moralisch verwerflich, naiven Anwendern zu erlauben, ihr Geld zu behalten.

Zweite Grundprämisse von Windows-Softwarehäusern:

Es ist moralisch verwerflich, Programmierern zu erlauben, für ihr Hobby Geld zu kassieren.

Deswegen sind sich Anwender und Programmierer seit Jahren darüber einig, daß ihrer beider Leben ohne geldgierige Softwarehäuser und deren nicht nachvollziehbaren Anforderungen an Wirtschaftlichkeit und Effektivität sehr viel leichter wäre, was nun wiederum verallgemeinert ausgedrückt wird als

Das allgemeine Effektivitätsgesetz der Software-Entwicklung:

Die ersten neunzig Prozent des Programms dauern neunzig Prozent der Entwicklungszeit. Die anderen zehn Prozent brauchen die anderen neunzig Prozent.

Windows-Ergänzung zum allgemeinen

Effektivitätsgesetz der Software-Entwicklung:

Unter Windows dauert beides doppelt so lang und kostet viermal soviel.

Da sich alle Beteiligten nun, wie wir jetzt gesehen haben, darüber einig sind, daß sie entbehrlich sind, ist die konsequente Folgerung aus dem Windows-Erfolg die konsequente Einführung computerunterstützter Softwareentwicklung, Vermarktung und Anwendung mittels »CASE« (»Computer Aided Desaster«), weil letztendlich nur ein Computer Windows-Programme so schreiben kann, daß andere Computer sie auch in der richtigen Art mißverstehen können.

Die MC-Entdeckung:

Die Klage eines Anwenders, ein Windows-Programm sei schlecht, wird der Entwickler immer mit der Aussage kontern, daß dieser nicht verallgemeinern solle. Was sind schließlich schon 19 Bugs im Verhältnis zu den 50000 anderen Programmzeilen.

Michis Windows-Grunderkenntnis:

Ändere nie ein unter Windows lauffähiges Programm.

Erweiterung von Michis Windows-Grunderkenntnis:

Ändere auch kein anderes Programm.

Axiom der Fehleroptimierung:

Wenn ein Windows-Programm bei Dir fehlerfrei läuft, mach Dir keine Sorgen. Es wird vorbeigehen.

Folgerungen aus dem Axiom der Fehleroptimierung:

1. Wenn es nicht mehr schlimmer werden kann, wird es schlimmer.
2. Wenn Du glaubst, der Fehler sei behoben und das Programm läuft wieder, hast Du etwas übersehen.
3. Jedes fehlerfreie Windows-Programm produziert eine allgemeine Schutzverletzung.
4. Wenn eine Kette von Ereignissen schiefgehen kann, wird es in der schlimmstmöglichen Reihenfolge geschehen.
5. Wenn es schlimm gewesen ist, wird es nochmal passieren.
6. Wenn das Schlimmstmögliche eingetreten ist, ist jemandem, den Du kennst, genau das erst vor kurzem passiert – nur alles viel schlimmer.

Axiom von der Problemvermehrung:

In jedem großen Problem steckt ein kleines, das gerne raus will.

Schainkers Umkehrung:

In jedem kleinen Problem steckt ein großes, das gerne raus will.

Joachims Stoßseufzer:

Auch wo überhaupt kein Problem ist, steckt ein großes, das gerne raus will.

Gesetz von der statistischen Logik der Windows-Entwickler:

80 Prozent der Benutzer setzen laut Marktuntersuchungen nur 20 Prozent der Funktionen ein.

Erste logische Ableitung:

20 Prozent der Benutzer benötigen die 80 Prozent der Funktionen, die sein Programm nicht besitzt.

Zweite logische Ableitung:

Du gehörst mit hundertprozentiger Sicherheit zu diesen 20 Prozent.

Verdeutlichung:

Ein Windows-Programmierer wäre der letzte, der sein Programm auch anwendet.

Lehrsatz vom Nutzen eines Windows-Programms:

Das, was du mit Deinem Programm machen willst

- steht nicht im Handbuch
- wird erst im Update des Handbuchs erklärt
- wird erst in der nächsten Version des Programms implementiert.

Die vier allgemeinen Grundlagen der Windows-Programmierung:

1. Jedes Windows-Programm, das fehlerfrei läuft, ist veraltet.
2. Jedes nützliche Windows-Programm wird geändert.
3. Jedes unsinnige Feature wird sofort dokumentiert.

4. Jeder Fehler wird sofort als neue Funktion eingebaut.

Verallgemeinerung:

Wenn Du irgendetwas verstanden hast, ist es veraltet.

Das CBT-Gesetz:

1. Die, die können, tun.
2. Die, die nicht können, unterrichten.
3. Die, die nicht unterrichten können, verkaufen Schulungen.
4. Die die keine Schulungen verkaufen können, schreiben CBT-Programme.
5. Die die keine CBT-Programme schreiben können, entwickeln CBT-Autorensysteme.

Weiner's Gesetz über API's und DLL's:

Es gibt keine Antworten. Nur Cross-Referenzen.

Raimond Reicherts Windows-Definition:

Für jedes Softwarehaus ist ein Gramm schöne Oberfläche wichtiger als ein Kilo Performance.

Raimond Reicherts Ableitung für

Fensterprogrammierer:

Der darzustellende Text paßt nie in das Fenster, das der Bildschirm zur Verfügung stellt.

Brasingtons erstes Gesetz:

Die Wahrscheinlichkeit, daß ein gegebenes Programm den Anforderungen entspricht ist negativ proportional zu der Selbsteinschätzung des Programmierers.

Brasingtons zweites Gesetz:

Wenn ein Programmierer sagt »kein Problem«, hast Du eines.

Brasingtons drittes Gesetz:

Wenn ein Programmierer sich zu einem Abgabetermin verpflichtet, stell sicher, daß er auch Monat und Jahr nennt.

Brasingtons viertes Gesetz:

Kein Windows-Programm wird je komplett debugged werden.

Bernhards erste Erweiterung von Brasingtons viertem Gesetz:

Wenn Du einen Fehler in Deinem Windows-Programm beseitigst, kommt unweigerlich ein weiterer Fehler hervor, der vorher durch den ersten aufgehoben wurde.

Bernhards zweite Erweiterung von Brasingtons viertem Gesetz:

Dieser neue Fehler hat ungleich schlimmere Folgen.

Scotts Gesetz:

Wenn ein Fehler entdeckt und beseitigt wurde, wird festgestellt, daß es keiner war.

Mexners Speicheraxiom:

Programmcode neigt dazu, den kompletten zur Verfügung stehenden Speicher auszufüllen und zu überschreiten.

Die Windows-Ergänzungen von Mexners

Speicheraxiom:

1. Wenn Du sämtliche Kommentarzeilen löschst und umständliche Programmroutinen neu und kürzer programmierst, wird das Programm hinterher länger sein, mehr Speicherplatz benötigen, zu groß für den Compiler sein und darüberhinaus nicht mehr funktionieren.
2. In jedem Programm neigen Fehler dazu, am entgegengesetzten Ende Deiner Fehlersuche aufzutreten.
3. Wenn ein Listing Fehler aufweist, sieht es fehlerfrei aus.
4. Wenn ein Fehler entdeckt und korrigiert wurde, stellt sich heraus, daß es schon zu spät ist.
5. War es nicht zu spät, war die Korrektur falsch und der ursprüngliche Text richtig.

Erste Folgerung:

Nachdem die Korrektur falsch war, wird es unmöglich sein, den Anfangszustand wieder herzustellen.

Zweite Folgerung:

Von zwei möglichen schlechten Ereignissen wird nur das tatsächlich eintreten, bei dem der Fehler auf Dich zurückzuführen ist.

Brasingtons fünftes Gesetz:

Entwicklungsprojekte laufen schnell, bis sie zu neunzig Prozent fertig sind. Dann bleiben sie für immer zu neunzig Prozent fertiggestellt.

Brasingtons Programmiererweisheit:

Bei einem Entwicklungsprojekt, das nachträglich verändert werden kann, wird die Änderungsrate die Rate der Fortentwicklung stets überschreiten.

Brasingtons sechstes Gesetz:

Ein schlecht geplantes Entwicklungsprojekt braucht durchschnittlich dreimal solange wie geplant. Ein gut geplantes wird nur viermal solange brauchen.

Brasingtons siebtes Gesetz:

Das Backup, das Du gerade gemacht hast, wirst Du nie brauchen.

Brasingtons achtes Gesetz:

Das einzige Backup, daß Du je brauchst ist entweder

- das, für das Du keine Zeit hattest, oder
- das, das Du zwar gemacht hast, aber nun nicht mehr lesen kannst.

Brasingtons neuntes Gesetz:

Es besteht keine Gefahr, daß die Röntgenstrahlen im Flughafen die Daten auf Deiner Diskette oder Deinem Band löschen. Aber jeder Spielzeugmagnet im Umkreis von 50 Metern wird es tun.

Das »Jetzt erzählen sie es uns«-Gesetz:

Erst wenn ein Programm fertiggestellt ist, die Handbücher geschrieben und die Marketingaktionen mit der Werbeagentur abgesprochen sind, werden die Entwickler die Informationen bekommen, die ein Redesign absolut unverzichtbar machen.

Cole's Begründung:

Die Summe der Intelligenz auf diesem Planeten ist konstant, die Bevölkerung wächst.

Gesetz des Vergessens:

Die Wahrscheinlichkeit, etwas zu vergessen ist direkt proportional zu ... ist direkt proportional zu ... zu ...

Das Machrone-Statement:

Wenn Du es entwickeln kannst, ist es überholt.

Die Cirrus-Konkretisierung zum Machrone-Statement:

Erst wenn du mit großen Anstrengungen ein Windows-Programm fertig entwickelt hast und Deine letzten Kredite in den Entwurf von Anzeigen und Werbematerialien gesteckt hast, wirst Du feststellen, daß Microsoft ein gleichartiges Programm auf den Markt werfen wird.

Cheops Erkenntnis:

Kein Projekt wurde jemals pünktlich und innerhalb eines gesetzten Kostenplans beendet.

Die Programmierer-Ausrede auf Cheops Erkenntnis:

Der frühe Vogel fängt den Wurm. Der frühe Wurm – wird gefressen.

Daniels Gesetz:

Die einzigen Dinge, die Du pünktlich beginnst, sind die, bei denen Du zu spät dran bist.

Goldwyns Gesetz über Entwicklungsaufträge:

Ein mündlicher Vertrag ist das Papier nicht wert, auf dem er geschrieben steht.

Das Windows-Netzwerkgesetz:

Zwei Drittel der Erdoberfläche ist mit Wasser bedeckt. Die anderen zwei Drittel sind bedeckt mit Netzwerk-administratoren der Firmenleitung.

Lewis' Gesetz vom Direktvertrieb:

Egal wie lange Du nach einer billigen Einkaufsquelle suchst. Wenn Du das Windows-Programm schließlich gekauft hast, wirst Du einen Händler finden, bei dem Du es hättest billiger bekommen können.

Ehrlich gemeinter Programmierer-Segen:

Gesegnet sei der End-Benutzer, der nichts erwartet. Er wird nicht enttäuscht werden.

Nicht-Reziprokes Gesetz der Erwartung:

Schlechte Erwartungen bedingen schlechte Erwartungen. Gute Erwartungen bedingen schlechte Ergebnisse.

Fazit für Windows-Anwender:

Glaube nicht an Wunder, verlaß dich auf sie.

Das »Darf ich Ihnen mal unverbindlich mein Windows-Demo zeigen«-Gesetz:

Je höher die hochgestellten Manager sind, bei denen es Dir gelungen ist, sie zur Besichtigung Deines Demos zu

bewegen, umso niedriger sind Deine Chancen, auch eine erfolgreiche Demonstration durchzuführen.

Daniels Gesetze über Testberichte:

1. Dein Programm wird von Computerzeitschriften so lange nicht getestet, bis Konkurrenzprodukte auf den Markt gebracht werden, die besser sind.
2. Trifft Satz 1 nicht zu, wird der Testredakteur behaupten, daß die herausragenden Features Deines Programms keiner braucht.
3. Treffen Satz 1 und Satz 2 nicht zu, hat der Testredakteur die herausragenden Features Deines Programms nicht bemerkt.

Daniels Ableitungen:

1. Dein Programm schneidet immer am schlechtesten ab.
2. Das beste Testergebnis bekommt immer Dein härtester Konkurrent.
3. Die Wertung ist umso katastrophaler, je wichtiger die testende Zeitschrift für die anvisierte Zielgruppe ist.

Theoretisches Gesetz der Programmiersprachen-Kompatibilität (neuerdings nur noch »'für-Windows'-Axiom« genannt):

- 1) Prämisse: Selbst wenn es gelänge, alle Programmiersprachen der Welt durch eine einzige, einheitliche Programmiersprache zu ersetzen – es wird auch dann immer genug Hersteller geben, die diese

einzigste, einheitliche Programmiersprache in einer eigenen Spezialentwicklung auf den Markt bringen.

- 2) Folgerung: Diese Spezialentwicklungen werden zu nichts kompatibel sein außer zu sich selbst.
- 3) Einschränkung: Die Inkompatibilität erstreckt sich aber selbstverständlich auch auf verschiedene Versionsnummern derselben Spezialentwicklung.

Grundsatz der Windowsanwendung:

Wenn nichts mehr funktioniert, lies endlich die Gebrauchsanweisung.

Widerspruch zum Grundsatz der Windowsanwendung:

Ein Handbuch wird nicht gelesen.

Ausnahmen:

1. Schlechte Handbücher werden von Testredakteuren gelesen.
2. Es werden nur die Abschnitte im Handbuch gelesen, die einen Anwender dazu veranlassen, das Falsche zu tun.
3. Jedes Handbuch ist bei Drucklegung veraltet.

Axiom von der Handbuch/Programm-Relation :

Wenn Du etwas so genau erklärst, daß es nicht mißverstanden werden kann, wird es irgendwer doch tun.

Vogels Zusammenhang zwischen Testbericht und Handbuch:

1. Machst Du ein gutes Handbuch, ist es dem Testredakteur nicht ausführlich genug.

2. Machst Du ein gutes und ausführliches Handbuch, werden Handbücher beim Test nicht bewertet.
3. Machst Du ein schlechtes Handbuch, ist es das ausschlaggebende Kriterium für die Testberichte in allen Computerzeitschriften.

Allgemeine Anwendererkenntnis für Handbücher von Windows-Programmen:

1. Jedem Windows-Programm liegen mehrere Handbücher bei.
2. Sie werden sich stets an den entscheidenden Stellen widersprechen.

Gesetze vom Softwarekauf:

1. Angaben des Herstellers über die Leistung sollten mit dem Faktor 0,5 multipliziert werden.
2. Ansprüche der Anwender über die Leistung werden mit dem Faktor 0,25 multipliziert.
3. Mitgelieferte Handbücher und Systemdisketten werden sofort in der Poststelle abgelegt und bleiben dort unauffindbar.

Bernhards genereller Trost:

Löscht man Windows oder läßt das Computern gar ganz sein, um etwas anderes, nützliches zuwege zu bringen, dann wird Murphys Gesetz sofort und unmittelbar bei ebenjener anderen Tätigkeit gültig.



Egal was Du tust: Du wirst Deine Daten verlieren.

H. Zitate

Aus dem morphologischen Dunstkreis fehlerhafter Hardware, fehlerhafter Software und Windows tauchen immer wieder Zeitgenossen auf, die mit klugen Worten über die Vergänglichkeit des Menschen und die Unvergänglichkeit von Computerfehlern philosophieren. Einige belegte Zitate sind im folgenden aufgeführt.

Über Windows und die Hardware:

»Personal Computer mit 80286-Prozessor, 640 KB konventioneller Speicher sowie 256 KB Erweiterungsspeicher«

(Von Microsoft gestellte Systemanforderungen auf der Windows-3.1-Packung)

Über Windows und die Software:

»Wer Speicher nicht allokiert oder an absolute Adressen legt oder nicht allokierten Speicher oder an absoluten Adressen liegenden Speicher sich verschafft und in Programmen verwendet, wird mit Speicherentzug nicht unter 2 MByte bestraft.«

(Spruch an der Wand in der Hotline von Borland)

Über Microsofts Programmierpolitik:

»Information ist Macht. Richtige Information zur richtigen Zeit veröffentlicht, kontrolliert die Gegenwart. Richtige Information zur richtigen Zeit zurückgehalten kontrolliert die Zukunft.«

(Charles J.C. Lyall)

Über Windows-Programme (1):

»Die preiswertesten, schnellsten und effektivsten Computerprogramme sind die, die es nicht gibt.«

(Gorden Bell)

Über Windows-Programme (2):

»Windows-Programme sind wie Entropie. Sie sind schwierig zu fassen, wiegen nichts und werden immer größer.«

(Norman R. Augustine)

Über MPEG, ISO, OSF, ISO, MPC und Computerhändler:

»Vier Dinge hemmen den menschlichen Fortschritt: Ignoranz, Dummheit, Normungskomitees und PC-Verkäufer.«

(Charles J.C. Lyall)

Über den Windows-Benchmark von PC-Magazine:

»Es gibt drei Arten von Lügen: Einfache Lügen, gemeine Lügen und Benchmarktests.«

(John Dvorak, PC-Magazine 12/91)

Achim Rohdes Erweiterung:

»Benchmarks lügen nicht. Aber Lügner verwenden Benchmarks.«

Über Hardware im Allgemeinen:

»Unbelebte Objekte können in drei Kategorien eingeteilt werden:

– solche, die nicht funktionieren

- solche, die kaputtgehen und
- solche, die verlorengehen.«

(Russel Baker)

Über Computer im Besonderen:

»Das Unsympathische an den Computern ist, daß sie nur ja oder nein sagen können, aber nicht vielleicht.«

(Brigitte Bardot)

Über Computer im Speziellen:

»Ich glaube an Computer, aber nur, wenn sie verliebt sind.«

(Groucho Marx)

Über das Leben:

»Das ganze Leben ist sieben zu fünf dagegen.«

(Damon Runyon)

Über Kohl:

»Hier gilt das Prinzip »GIGO« (garbage in – garbage out): wenn Unsinn (»Kohl«) hineingesteckt wird, kommt Unsinn (»Kohl«) heraus.«

(»Der Umgang mit Computern«, herausgegeben von der Landeszentrale für Politische Bildungsarbeit, 3. Auflage 1985, Seite 192)

Über Technologie:

»Technologie don't transfer.«

(Conrad Stenton)

Über künstliche Intelligenz:

»Es kann sein, daß es einen Markt für Künstliche Intelligenz gibt. Auf jeden Fall gibt es einen größeren Markt für Künstliche Dummheit.«

(Charles J.C. Lyall)

Über EDV-Berater:

»EDV-Berater sind wie hochpreisige Huren: sie kennen mehr Praktiken als ihre Kunden, sie können es besser als ihre Kunden, aber die meiste Zeit machen sie es, wie es die Kunden wollen und versuchen es so zu machen, wie es die Kunden lieben.«

(Charles J.C. Lyall)



*Jede Windows-Anwendung weiß, was gut für
Dich ist und was Du wirklich brauchst.*

I. Erkenntnisse

Grundlegende Erkenntnisse (»Grafische Benutzeroberflächen nützen Dir«, »Die Erde ist eine Scheibe«) setzen sich im Leben von Computerbenutzern nur langsam durch. Deswegen hat der Autor im folgenden allgemeingültige Erkenntnisse, die sich im harten Windows-Einsatz durchgesetzt haben, zusammengestellt. Tröstlich ist bei der Lektüre dieses Kapitels einzig die

Allgemeine Erkenntnis:

Alle allgemeinen Erkenntnisse sind verkehrt.

Die Windows-Erkenntnis:

Windows inspiriert den Anwender zu großen Meisterwerken und hindert ihn zugleich, sie auszuführen.

Erste Konkretisierung der Windows-Erkenntnis:

Der Mensch denkt und Windows macht, was es will.

Zweite Konkretisierung der Windows-Erkenntnis:

Laß unter keinen Umständen ein Windows-Programm wissen, daß Du in Eile bist.

Die Microsoft-Ausrede:

»Sie haben schon mit schnelleren und besseren Oberflächen als Windows gearbeitet? Die waren aber nicht von uns.«

Das Gespräch beim Händler:

»Ich möchte Windows kaufen.« »Sind sie verrückt?«
»Gehört das zu den Lizenzbedingungen?«

Die doppelte Multitasking-Erkenntnis:

1. Alleingelassen werden sich die Dinge vom Schlechten zum sehr Schlechten verändern.
2. Jeder Task braucht die doppelte Zeit als Du vermutest. Dieses Gesetz ist zwingend rekursiv.

Die Teppich-Erkenntnis:

Die Chance, daß ein Brot auf die Butterseite fällt, ist direkt proportional zum Wert des Teppichs.

Die Doppelerkenntnis vom Windows-Graphics-Accelerator:

1. Verwende keine sinnlose Gewalt. Hol einen größeren Hammer.
2. Manchmal hilft Einstecken.

Die Disketten-Regel:

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

Hawkin's Fortschrittserkenntnis:

Fortschritt ist nicht die Ersetzung einer falschen Theorie durch eine richtige, sondern die Ersetzung einer falschen Theorie durch eine, die mehr subtil falsch ist.

Die allgemeine Backup-Erkenntnis

Nähert sich das Ende der Welt – mach' schnell ein Backup.

Die Windows-Backup-Erkenntnis:

Ich hab nicht meinen Verstand verloren, ich hab ihn irgendwo auf einem Band archiviert.

Die Installations-Erkenntnis:

Mißtraue jedem, der Dir erzählt, etwas wäre leicht zu installieren. Er ist entweder ein Irrer, ein Ignorant oder ein Computerhändler.

Die allgemeine Datenverarbeitungs-Erkenntnis:

Es ist leichter, Daten in ein Programm hineinzubekommen, als sie wieder herauszukriegen.

Die spezifische Datenverarbeitungs-Erkenntnis:

In einer beliebigen Ansammlung von Daten ist der Wert, der nach menschlichem Ermessen mit größter Sicherheit richtig ist, derjenige, in dem der größte Fehler steckt.

Die Turbo-Erkenntnis:

Sobald sich an Windows irgendwas beschleunigt, beschleunigt sich auch die Absturzrate.

Die Erkenntnis von den fehlenden**Glücksvoraussetzungen:**

Nur wenn du dreimal auf Holz klopfen willst, stellst du fest, daß die Welt aus Plastik und Aluminium besteht.

Die Freak-Erkenntnis:

Hilf jemandem bei Problemen, und er wird sich an dich erinnern – wenn er wieder Probleme hat.

Die Fenster-Erkenntnis:

1. Ein Objekt, daß sich auf dem Bildschirm bewegt, bewegt sich in die falsche Richtung.
2. Ein stillstehendes Objekt ist am falschen Platz.

3. Das am dringendsten benötigte Objekt wird am unzugänglichsten sein.

Horngrens VR-Erkenntnis:

Die reale Welt ist ein spezieller Fall.

Ginsbergs Erkenntnis von der Windows-Thermodynamik:

1. Du kannst nicht gewinnen.
2. Du kannst nicht mal unterbrechen.
3. Du kannst nicht quitten.

Die Erkenntnis von der Vergeblichkeit grafischer Benutzeroberflächen:

Wenn Gott wirklich gewollt hätte, daß Computer leicht zu bedienen seien, hätte er den Menschen ein Interface gegeben.

Die Erkenntnis von den kinderleicht zu bedienenden Programmen:

Nur Erwachsene haben Probleme mit kindersicheren Flaschenverschlüssen.

Die Druckertreiber-Erkenntnis:

Der Wert eines Programms ist proportional zum Gewicht seines Outputs.

Die Gleichungs-Erkenntnis:

In jedem Problem mit »n« Gleichungen werden »n+1« Unbekannte sein.

Die Pannen-Erkenntnis:

Pannen treten in Wellen auf.

Die NT-Erkenntnis:

Jede Form neuer Technologie ist untrennbar mit Formen von Magie und Zauberei verknüpft.

Golds Iconerkenntnis:

Ein Programm, das funktioniert, sieht häßlich aus.

Lyalls fundamentale Erkenntnis:

Das wichtigste Bein bei einem dreibeinigen Stuhl ist das Bein, das fehlt.

Conners geheime Erkenntnis:

Was vertraulich ist, bleibt im Drucker liegen.

Die Datenbank-Erkenntnis:

Eine virtuelle Datenbank ist ein Teil Deiner Phantasie.

Throops universelle Erkenntnis von grafischen Benutzeroberflächen:

Auch das Universum ist nicht benutzerfreundlich.

Erkenntnis von der Unentrinnbarkeit klebriger Flüssigkeit

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

Folgerungen:

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-Lese-Öffnung fließen.
5. Stehen mehrere Disketten zur Auswahl, wird die Flüssigkeit in die Diskette fließen, auf der mehr Dateien sind, deren Inhalt nicht wiederbeschaffbar 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.

Die Erkenntnis über den Unterschied zwischen Microsoft und einem Autohersteller:

»Kein Autohersteller dürfte es sich erlauben, den Kunden durch Bleigewichte an der Karrosserie zum Kauf eines PS-stärkeren Modells zu zwingen, damit er wieder das gewohnte Tempo 200 fahren kann.«

(Wolfgang Redtenbacher, ISO-Programmiersprachennormungsausschuß)

Die 3,5-Zoll-Müll-Erkentnis

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

Die Diskettenbox-Erkenntnis

1. Du bekommst eine Diskette leichter in die Schachtel hinein als wieder hinaus.
2. Die Plastiktrenner in der Diskettenschachtel dienen dazu, Dir den Blick auf die gesuchte Diskette zu versperren.
3. Ansonsten klappen Sie die gesuchte Diskette 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 einmal 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.
8. 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.
9. Du hast immer zuwenig Diskettenhüllen, zuwenig Diskettenboxen oder zuwenig Disketten.

Die Etiketten-Erkenntnis:

1. Schreibschutzetiketten kleben nicht richtig.

2. Schreibschutzetiketten wandern an die unzugängliche Stelle des Diskettenlaufwerks.
3. Schreibschutzetiketten gehen nicht wieder ab.
4. Schreibschutzetiketten fehlen dann, wenn Du aus Versehen eine wichtige Diskette formatierst oder Dir einen Virus eingefangen hast.
5. Schreibschutzetiketten kleben dann auf der Datendiskette, wenn Du speichern willst und das Anwendungsprogramm diesen Fehler nicht abfangen kann.
6. In diesem Fall wird das Programm unter Datenverlust abstürzen.

Tussemans Fehler-Erkenntnis:

Nichts ist so erfolgreich, wie ein Fehler, dessen Zeit gekommen ist.

Erkenntnis der Inkompetenz:

Inkompetenz kennt keine Grenzen von Raum und Zeit.

Waltons Menü-Erkenntnis:

Bei zwei gegebenen Entscheidungsalternativen wirst Du Dich für die falsche entscheiden – zweimal.

Die Additions-Erkenntnis:

Auf der Welt gibt es drei Arten von Menschen: solche, die zählen können, und solche, die nicht zählen können.



*Dein Touchscreen-Treiber ist nur dann
microsoftmauskompatibel, wenn Du Monitor
und Touchscreen über den Schreibtisch rollst.*

J. Definitionen

Die OLE-Definition:

Programme und Daten in ein anderes Programm einzubetten hat den gleichen Effekt, wie Hundertmarkscheine in die Kloschüssel einzubetten und abzuziehen.

Die DDE-Definition:

Dynamischer Datenaustausch zwischen zwei Programmen bedeutet, daß die korrekten Daten der beiden Programme durch die fehlerhaften Daten des jeweils anderen vollautomatisch ersetzt werden.

Verallgemeinerung der DDE-Definition:

Ein dynamischer Datenaustausch ist es nicht.

Verallgemeinernde Definition von grafischen Benutzerschnittstellen:

Eine grafische Benutzerschnittstelle bedeutet, daß sich der Benutzer geschnitten hat, wenn er meint, er könne damit vernünftig arbeiten.

Geschwindigkeits-Definition von Pen-Windows:

Da ist ein »n« zuwenig.

Die Host-Definition:

Ein Großrechner ist das größte Stück PC-Peripherie, das auf dem Markt ist.

Definition des Winchesterlaufwerks:

Hardware, die dazu da ist, Dateien zu zerschießen.

Definition der modernen Wissenschaft:

1. Wenn es grün ist oder sich bewegt, ist es Biologie
2. Wenn es stinkt, ist es Chemie.
3. Wenn es nicht funktioniert, ist es Informatik.

Die Betriebswirtschafts-Definition:

Betriebswirtschaft ist die Lehre vom Geld und wie es die Gesetze von Mathematik und gesundem Menschenverstand mißachtet.

Die Informatik-Definition:

Informatik ist die Lehre vom Computer und wie sie die Gesetze von Betriebswirtschaft, Mathematik und gesundem Menschenverstand mißachtet.

Horwoods erste Definition der Daten:

Gute Daten sind die, die Du bereits in Händen hältst.

Horwoods zweite Definition der Daten:

Schlechte Daten verdrängen gute Daten.

Horwoods dritte Definition der Daten:

Die Daten, die du für die gegenwärtige Krise hast, wurden für die letzte gesammelt.

Horwoods vierte Definition der Daten:

Die verfügbaren Daten erscheinen umso glaubwürdiger, je länger es gedauert hat, sie zu beschaffen und je weiter die Datenquelle vom Betrachter entfernt ist.

Horwoods fünfte Definition der Daten:

Daten können zwar zwischen verschiedenen Büros bewegt werden, sie können aber weder neu geschaffen noch gelöscht werden.

Horwoods sechste Definition der Daten:

Wenn du die richtigen Daten hast, hast du das falsche Problem. Oder umgekehrt.

Horwoods siebte Definition der Daten:

In komplexen Systemen gibt es keine Relation zwischen den verfügbaren Daten und den daraus folgendenden Entscheidungen.

Die Adlib-Definition:

Kanadier wären so glücklich mit der Kombination aus amerikanischem Know-How, französischer Kultur und englischer Verwaltung. Und was haben sie? Englisches Know-How, französische Verwaltung und amerikanische Kultur.

Definition der Benutzerfreundlichkeit

Benutzerfreundlichkeit ist das entgegenkommende, höfliche und duldsame Verhalten des Anwenders gegenüber dem patzigen, rätselhaften und unflexiblen Verhalten von Windows und Windows-Programmen.

Die Ressource-Kit-Definition:

Der gelungene Versuch, Firmen mehrere hundert Mark dafür abzunehmen, daß sie die Windows-Hotline nicht mehr anrufen.

Definition des praxisgerechten Befehlsumfangs:

Das von Dir gekaufte Programm kann nur die Hälfte.

Die Logik-Definition:

Logik ist die systematische Methode, mit den richtigen Überlegungen zu den falschen Schlußfolgerungen zu kommen.

Die Kaffee-Definition:

Kaffee macht jeden Programmierer weiser und läßt ihn auch durch halbgeschlossene Augen sehen.

Die Oktal-Definition:

Oktal ist wie ein Zehn-Finger-System, wenn Dir zwei Finger fehlen.

Definition der zu großen Fehlerrate:

Wenn die Fehlerrate so groß geworden ist, daß sie bemerkt werden kann, ist sie zu groß.

Augustines Definition des optimalen**Normierungsgremiums:**

Das optimale Normierungsgremium hat null Mitglieder.

Definition der acht Computer-Katastrophen:

Der Anwender, der Hersteller, das Modell, der PC-Verkäufer, das Betriebssystem, die Programmiersprache, das Anwendungsprogramm und die Benutzeroberfläche.

Definition der Verhaltensweise von Computer-Anwendern:

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

Definition von idiotensicheren Systemen:

Systeme, die idiotensicher sind, werden auch nur von Idioten bedient.

Definition der Verhaltensweise von Computern in Netzwerken:

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

Definitionen von Erweiterungen:

1. Begriffe wie »Enhanced«, »Advanced« oder »Extended« bedeuten, daß es dem Hersteller gelungen ist, einen Teil der Fehler zu beheben, die die Vorgängerversion unbrauchbar gemacht haben.
2. Solch ein Produkt wird erst dann auf den Markt gebracht, wenn Du die Vorgängerversion gekauft hast.
3. Jeder Hersteller wird in seinen verbesserten Produkten noch genügend Fehler verstecken, damit es noch für mindestens eine wiederum »verbesserte« Version reicht.

Definition von Optionen:

1. Wenn ein Hersteller sagt, sein Produkt habe diese und jene Option, so heißt das nur, daß der Hersteller bewußt ein wichtiges Modul weggelassen hat, damit man sie extra kaufen muß.
2. Die Erweiterung wird mehr kosten als das eigentliche Produkt.

3. Die Erweiterung wird funktionieren, allerdings nicht bei Dir.
4. Wenn Du es irgendwann einmal verkaufen willst, wirst Du die Erweiterung nicht los.

Definition des Unterschieds zwischen Gebrauchtwagen- und Computerhändler:

Ein Gebrauchtwagenhändler weiß, daß er lügt.

Allgemeine Definition der Programmierer:

Die einzige Sprache, die alle Programmierer beherrschen, ist Gleichgültigkeit.

Allgemeine Definition der Anwender:

Die einzige Sprache, die alle Anwender beherrschen, ist Ignoranz.

Definition eines Windows-Programmierers:

Windows-Programmierer sind Menschen, die zu nachtschlafender Zeit mit völlig untauglichen Entwicklungspaketen für nicht zusammenpassende Konglomerate fehlerverseuchter Hardware auf einer fehlerhaften Benutzeroberfläche über einem unausgereiften Betriebssystem versuchen, im Auftrag von unfähigen Bedienern deren einander widersprechenden Anforderungen in Programme umzusetzen, die am Schluß niemand verwendet.

Definition eines Windows-Anwenders:

Windows-Anwender sind Computerperipheriegeräte, die versuchen, mit nicht zusammenpassenden Konglomeraten fehlerverseuchter Hardware auf einer

fehlerhaften Benutzeroberfläche über einem unausgereiften Betriebssystem mit einem unverständlichen Programm von unfähigen Windows-Programmierern ein Problem zu lösen, das es ohne Computer gar nicht geben würde.

Definition von Computerverlagen:

Computerverlage produzieren Windowsbücher, um darin das zu erklären, was Du in Windowszeitschriften nicht verstanden hast. Windowszeitschriften werden aus dem umgekehrten Grund produziert.

Konsequente Folgerungen:

1. Du verstehst weder das eine, noch das andere.
2. Der Testbericht über das Dich interessierende Programm erscheint eine Woche, nachdem Du es gekauft hast.
3. Das von Dir gekaufte Programm hat immer die schlechtmöglichste Bewertung.
4. Das Programm, das Du fast gekauft hättest, hat die bestmöglichste Bewertung.

Definition der Computer-Softwarefirmen:

Softwarefirmen sind wie Wein: Der Flaschenhals ist immer oben.

Definition eines Computermarktforschers:

Ein Computermarktforscher ist jemand, der über die Entwicklung genauso verwirrt ist wie Du, aber der festen Überzeugung ist, er wäre es aus mehr tiefschürfenden Gründen.

Definition einer Computermarktstudie:

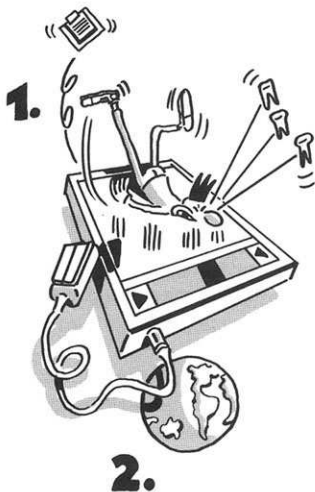
Jede Marktstudie über die Computerindustrie ist kompatibel zum Missouri: Eine Meile breit, nur ein Zentimeter tief, und es bedarf eines irren Aufwandes, eventuell darin enthaltene Goldkörnchen herauszuwaschen, weil in der Regel keine drin sind.

Definition der Computerjournalisten:

Computerjournalisten sind wie Romanautoren. Beide schreiben Märchen. Aber nur die Romanautoren geben es zu.

Definition der Computerverkäufer:

Es ist einfach, zu behaupten, ein Computerverkäufer lügt. Schaut auf seine Lippen: Wenn er sie bewegt, lügt er.



*Es gibt nur zwei unverrückbare
Erkenntnisse im Leben:*

- 1. Grafische Benutzeroberflächen machen das
Bedienen von Computern einfacher.*
- 2. Die Erde ist eine Scheibe.*

K. Index

Gesetz vom Index in Handbüchern für Windows-Programme:

Egal, was Du im Index des Handbuchs Deines Windows-Programms suchst: Das Stichwort wird nicht enthalten sein.

Erweiterung des Gesetzes vom Index in Handbüchern für Windows-Programme:

Du wirst das Stichwort auch nicht im Inhaltsverzeichnis, in der Gliederung oder im Glossar finden.

Schlußerkenntnis

Masochistische Grundtendenzen sind zwar keine Grundbedingung zum Einsatz von Windows. Aber sie helfen enorm.

Lieber Computer-Freund!

Computer tragen bekanntermaßen wie kaum etwas anderes dazu bei, Murphys Gesetz – daß nämlich alles, was schiefgehen kann, auch schiefgeht – zu optimieren. In seinem neuen Buch tritt Erfolgsautor Joachim Graf den Beweis an, daß wie jede andere Katastrophe auch Computerpannen auf ihre unsägliche Spitze getrieben werden können.

Grafische Benutzeroberflächen wie Windows führen zu einer neuen Qualität im Umgang mit Computern. Diese neue Qualität findet man insbesondere in Murphys Gesetzen, die die in Silizium und Programmzeilen gegossene Heimtücke der EDV-Produkte beschreiben.

Das Buch ist Pflichtlektüre für jeden Computerbesitzer, der insgeheim von Anfang an wußte, was Windows bedeuten kann:

»Desaster at your fingertips«.

ISBN 3-87791-364-4



Markt & Technik

DM 9,90

sFr 9,90

öS 77,-