

Fehlervermeidung

Es gibt keine Medizin gegen den Tod, und gegen Fehler fand noch niemand eine Regel.
Sigmund Freud (Widmung, die er auf sein Porträt schrieb)

Anwender von Textverarbeitungssystemen, Spreadsheets, Einrichtungen zur Datenbankabfrage, Flugleitsystemen und anderen interaktiven Systemen machen weit häufiger Fehler, als man annehmen sollte. Card & Co. (1980) berichteten, dass erfahrene professionelle Anwender von Textverarbeitung und Betriebssystemen in 31 % der Aufgaben, die ihnen gestellt wurden, Fehler machten oder ineffektive Strategien nutzten. Brown und Gould (1987) fanden heraus, dass sogar erfahrene Autoren in fast der Hälfte ihrer Spreadsheets Fehler machten. Andere Studien enthüllten, wie problematisch Anwenderfehler sind und welches Ausmaß der dadurch entstehende Produktivitätsverlust annimmt.

Eine Möglichkeit, den durch Fehler entstehenden Produktivitätsverlust zu reduzieren, ist die Verbesserung der durch das Computersystem angezeigten Fehlermeldungen. Shneiderman (1982) berichtete von fünf Experimenten, in denen Änderungen bei den Fehlermeldungen zu verbesserten Erfolgen bei der Fehlerbewältigung, geringeren Fehlerraten und verbesserter individueller Befriedigung führte.

Überlegene Fehlermeldungen waren präziser, im Tonfall positiv und konstruktiv (der Anwender erhielt Hinweise, was zu tun sei, anstatt einfach nur auf das Problem aufmerksam gemacht zu werden).

Die Designer wurden angehalten, statt vage oder feindselige Meldungen wie SYNTAX ERROR oder ILLEGAL DATA zu benutzen, informative Botschaften wie FEHLENDE RECHTE KLAMMER oder AUSWAHLMÖGLICHKEITEN LIEGEN IM BEREICH 1 BIS 6 zu verwenden.

Verbesserte Fehlermeldungen sind jedoch nur Stückwerk. Ein effektiverer Ansatz ist die generelle Vermeidung von Fehlermöglichkeiten. Dieses Ziel ist für viele Systeme einfacher zu erreichen, als es den Anschein haben mag.

Der erste Schritt ist das Verständnis von der Natur der Fehler. Eine Perspektive ist, dass Menschen Fehler oder »Ausrutscher« (Norman, 1983) unterlaufen, und Designer können ihnen helfen, diese zu vermeiden, indem sie Anzeigen und Menüs funktional anordnen, Befehle oder Menüauswahl klar unterscheidbar designen und es den Anwendern schwer machen, unumkehrbare Aktionen auszuführen. Norman bietet andere Richtlinien an wie Nicht-Möglich-Modi, Feedback über den Status des Systems geben und das Design nach der Konsistenz der Befehle ausrichten. Normans Analyse stellt praktische Beispiele und eine nützliche Theorie vor.

Drei Techniken können Fehler reduzieren, indem vollständige und richtige Aktionen sichergestellt werden: **korrekt passende Paare, vollständige Sequenzen und korrekte Befehle.**

Korrekt passende Paare

Ein häufiges Problem sind fehlende korrekt passende Paare. Es hat viele Ausprägungen und einige einfache Präventionsstrategien. Ein Beispiel ist der Fehler, nach einer linken Klammer die schließende rechte Klammer zu vergessen.

Wenn ein bibliografisches Suchsystem Boolesche Ausdrücke wie COMPUTER UND (PSYCHOLOGIE ODER SOZIOLOGIE) erlaubt, und der Anwender vergisst die rechte Klammer, gäbe das System eine Botschaft wie SYNTAX ERROR aus oder hilfreicher noch eine aussagekräftigere Botschaft wie RECHTE KLAMMER FEHLT.

Entsprechend sind andere paarweise Markierungen nötig, um Fettdruck, Kursivschreibung oder unterstrichenen Text in einer Textverarbeitung oder beim Webprogrammieren ein- und auszuschalten. Wenn die Textdatei `<b>Dies ist Fettdruck</b>` enthält, erscheinen die drei Worte zwischen den Markierungen fett gedruckt. Wenn das nachfolgende `</b>` fehlt, könnte der nachfolgende Text unbeabsichtigt fett gedruckt werden.

In jedem dieser Fälle ist eine paarweise Markierung für die vollständige und korrekte Operation nötig.

Unterlassung der abschließenden Markierung kann durch die Verwendung eines Editors, vorzugsweise bildschirmorientiert, verhindert werden, der die öffnende und die schließende Komponente eines Paares auf dem Bildschirm in eine Handlung legt. Beispielsweise erstellt das Eintippen einer linken Klammer gleichzeitig eine linke und rechte Klammer und setzt den Cursor dazwischen, um die Eingabe von Inhalt zu erlauben. Ein Versuch, eine der beiden Klammern zu

löschen, führt zur Löschung der anderen Klammer (und möglicherweise auch des Inhaltes dazwischen). Somit kann der Text nie in syntaktisch inkorrektur Form erscheinen. Einige mögen diesen rigiden Ansatz zu restriktiv finden. Für sie mag eine mildere Form des Schutzes angemessen erscheinen. Beispielsweise könnte beim Eingeben einer linken Klammer in der linken unteren Ecke des Bildschirms eine Botschaft mit dem Hinweis erscheinen, dass die rechte Klammer fehlt, bis dieses Zeichen eingetippt wurde.

Korrekte Befehle

Industrielle Designer erkennen, dass erfolgreiche Produkte sicher sein und sie den Anwender davor bewahren müssen, gefährlichen falschen Gebrauch vom Produkt zu machen. Flugzeugmotoren können nicht eher in den Rückwärtsgang geschaltet werden, bis die Landevorrichtungen vollständigen Bodenkontakt haben, und bei Autos kann man den Rückwärtsgang nicht einlegen, wenn man schneller als 5 Meilen pro Stunde fährt. Viele einfache Kameras verhindern Doppelbelichtungen (auch wenn der Fotograf ein Bild zweimal belichten möchte), und Geräte haben Sicherheitsvorkehrungen, damit man nicht daran herumbasteln kann, während der Strom eingeschaltet ist (auch wenn Experten gelegentlich Untersuchungen durchführen müssen).

Die gleichen Prinzipien können bei interaktiven Systemen angewendet werden. Schauen wir uns diese typischen Fehler bei der Benutzung von Befehlssprachen an:

Es werden nicht verfügbare Befehle eingegeben, nicht existente Dateien angefordert oder nicht akzeptable Datenwerte eingegeben. Diese Fehler werden oft durch ärgerliche typografische Fehler verursacht, wie z.B. einer unrichtigen Befehlsabkürzung, das Betätigen einer Reihe von Tasten statt der gewünschten einzelnen Taste, Verschreiben beim Dateinamen oder kleinere Fehler wie Auslassen, Einfügen oder Vertauschen von Zeichen.

Fehlermeldungen reichen vom lästigen ? oder WHAT? über das vage UNRECOGNIZED COMMAND oder SYNTAX ERROR bis zum tadelnden BAD FILE NAME oder ILLEGAL COMMAND.

Das kurze »?« ist geeignet für Experten, die einen trivialen Fehler gemacht haben und ihn erkennen können, wenn sie die Befehlszeile auf dem Bildschirm sehen. Aber wenn ein Experte es gewagt hat, einen neuen Befehl auszuprobieren, und seine Durchführung falsch verstanden hat, ist die kurze Botschaft nicht hilfreich. Sie müssen Ihre Planung unterbrechen, um sich der Korrektur des Problems zuzuwenden - und Ihrer Frustration, nicht das zu bekommen, was Sie wollten.

Einige Systeme bieten eine automatische Vervollständigung der Befehle an, die den Benutzern erlaubt, nur einige Buchstaben eines sinnvollen Befehls einzutippen. Sie könnten den Computer durch das Betätigen der Leertaste auffordern, den Befehl zu vervollständigen, oder der Computer kann das übernehmen, wenn die Eingabe ausreicht, um diesen Befehl von anderen zu unterscheiden.

Automatische Befehlskomplettierung kann Tastatureingaben ersparen und wird von vielen Anwendern gewünscht, aber es kann auch unterbrechend wirken, weil der Anwender überlegen muss, wie viele Zeichen für jeden Befehl eingegeben werden müssen, und sicherzustellen hat, dass der Computer die beabsichtigte Vervollständigung durchgeführt hat.