



## KISS: Keep it simple, stupid!

Diese vor allem in der IT gebräuchliche Weisheit scheint bei Beobachtungen von IT-Projekten oder eingesetzter Software in der Praxis nicht bekannt zu sein oder nicht ernst genommen zu werden. Die einfache und Erfolg versprechende Regel ist auf alle Planungs- und Lösungsprozesse komplexer Aufgabenstellungen anwendbar.

Betrachten wir kleine und große IT-Lösungen, so müssen wir zum Schluss kommen, dass deren ursprünglicher Zweck, das Vereinfachen von Problemlösungen, Verbessern von Abläufen oder die Steigerung der Arbeitsplatzproduktivität nur selten erreicht wird. Komplexe, überfrachtete und in Bezug auf deren Planung, Einführung, Betrieb und Bedienung schwer beherrschbare Software stellt eher die Regel als die Ausnahme dar. Leistungsfähigkeit und Einfachheit schließen sich, so scheint es, aus. Effiziente Unterstützung ist das Ziel von Softwarelösungen, sei es am Arbeitsplatz oder bei der Optimierung betrieblicher Abläufe. Unhandliche Lösungen sind für die Anwender ärgerlich, verursachen darüber hinaus hohe Kosten und stellen potenzielle Sicherheitslücken dar. Eine weitere scheinbar unmögliche Kombination ist die Entwicklung von sicherer und einfacher Software. Paradebeispiel der einen Kategorie sind Applikationen von Microsoft, trivial in der Bedienung, komplex und überladen der Unterbau und weltmeisterlich in Quantität und Qualität von Sicherheitsschwachstellen. Am anderen Ende der Skala wird gern Open-Source-Software gesehen, sicher, effizient, aber schwer zu beherrschen. Dass diese Schwarz-Weiß-Malerei weder für die eine noch die andere Seite vollständig zutrifft, liegt auf der Hand, doch die Erfahrung zeigt die tendenzielle Bestätigung dieser Einordnung. Obwohl Open-Source-Software bezüglich Usability stark zugelegt hat, gibt es immer noch Applikationen, deren Bedienung einiges an Auseinandersetzung damit voraussetzt.

Ist es unmöglich, leistungsfähige, sichere Software einfach bedienbar zu machen? Es gibt doch immer wieder Beispiele, die zeigen, dass sich Leistungsfähigkeit und Usability verbinden lassen. Die sehr erfolgreichen Applikationen aus dem Mozilla-Umfeld zeigen, dass es möglich ist, schlanke und sichere State-of-the-Art-Software anzubieten, die einfachst zu bedienen ist. Der Webbrowser Mozilla-Firfox und der E-Mail-Client Mozilla Thunderbird werden immer häufiger eingesetzt und erreichen ernst zu nehmende Marktanteile gegenüber Outlook und Internet-Explorer. Doch der Erfolg hat einen langen, teils beschwerlichen Weg hinter sich und geht ursprünglich auf Netscape zurück. Nachdem Netscape seine Internet-Suite der Open-Source-Community übergeben hatte, wurde mit der Weiterentwicklung des Mozilla-Paketes begonnen. Die Entwickler erkannten aber den Bedarf – im Gegensatz zur integrierten Gesamtlösung Mozilla – an schlanken, schnellen

und einfachen Applikationen. Der Startschuss war ein Stand-Alone-Browser, der nach einigen Namensänderungen nun Firefox heißt, später wurde der E-Mail-Client Thunderbird entwickelt und zurzeit befindet sich ein Kalender-Tool im Beta-Stadium. Der Erfolg ergibt sich durch die Ausrichtung am Bedarf der Anwender und an der Beherrschbarkeit des Entwicklungsprozesses, der im Vergleich zu fetten Gesamtpaketen managebar bleibt. Die Anzahl der gefundenen Sicherheitslücken zeigt die positive Auswirkungen der Designentscheidung und die Anwender danken dies mit hohen Downloadraten. Erstmals seit einigen Jahren wird an der Monopolstellung des Internet-Explorers geknabbert.

Aber nicht nur bei relativ überschaubarer Software wie Webbrowser und Mail-Client ist das KISS-Prinzip erfolgreich anzuwenden. Auch bei komplexen Unternehmensanwendungen sind Komplexitätsreduktion bei Planung und Implementierung durch einfache Schnittstellen und der Orientierung am Notwendigen und nicht am Machbaren möglich. Dieses Prinzip widerspricht jedoch den Interessen etablierter Softwarehersteller, deren Produktmarketing an vielfältiger Funktionalität, verpackt in meist perfekter Hülle, aufbaut. Mit einfachen Lösungen lassen sich schwerer Kunden gewinnen. Der Markt ist noch nicht reif für effiziente Software, auch weil die Anwender und Unternehmen nicht das nötige Know-how haben, um Anforderungen an Hersteller so zu formulieren, dass die einfachsten Lösungen das Ergebnis sind. Die Katze beißt sich in den Schwanz. Schließen wir mit Albert Einstein: „Mache die Dinge so einfach wie möglich – aber nicht einfacher.“

Dipl.-Ing. Dr. techn.  
Stefan Grünwald

