

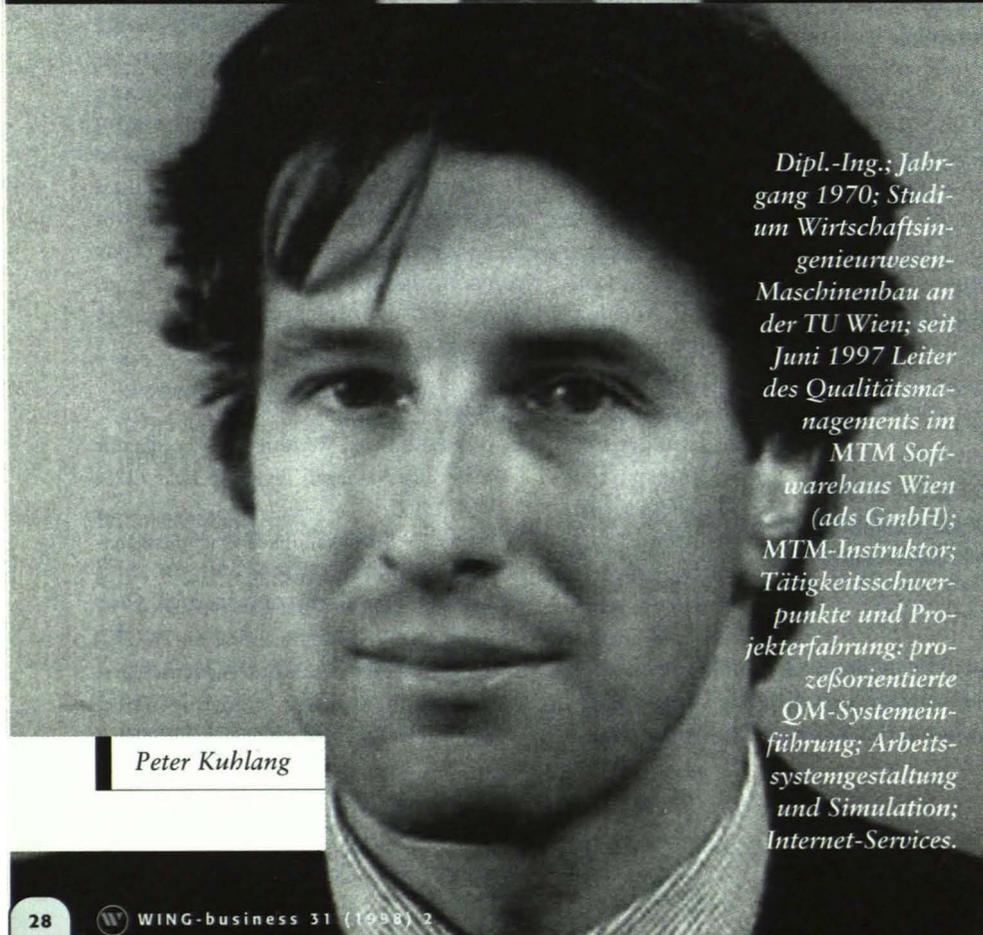
Softwarequalitätsmanagement



Kurt Matyas

Dipl.-Ing. Dr. techn.; Jahrgang 1963; Studium des Maschinenbaus (Studienzweig Betriebswissenschaften) an der TU Wien; seit 1989 Universitätsassistent am Institut für Betriebswissenschaften der TU Wien; Promotion 1992; Lehr- und Forschungsschwerpunkte: Betriebliche Instandhaltung, Qualitätsmanagement, Fabriklayoutplanung, Logistik, Produktionsplanung und -steuerung.

Viele Unternehmensberater, Techniker wie Kaufleute, private Anwender und sogar Systemadministratoren klagen über die „Qualität“ einer Software. Der Grund dafür ist, daß Abstürze, Programmfehler und eine mühsame Programmhandhabung den Arbeitsprozeß erschweren. Das kostet Nerven und vor allem viel Zeit.



Peter Kublang

Dipl.-Ing.; Jahrgang 1970; Studium Wirtschaftsingenieurwesen-Maschinenbau an der TU Wien; seit Juni 1997 Leiter des Qualitätsmanagements im MTM Softwarehaus Wien (ads GmbH); MTM-Instruktor; Tätigkeitsschwerpunkte und Projekterfahrung: prozessorientierte QM-Systemeinführung; Arbeitssystemgestaltung und Simulation; Internet-Services.

Qualität einer Software

Im Bereich des Industrial Engineering ergab sich aufgrund der Wichtigkeit dieser Problematik eine Kooperation zwischen der Technischen Universität Wien, Abteilung für Betriebstechnik und der ads GmbH, einem Tochterunternehmen der Deutschen MTM Vereinigung e. V.

Das Projekt beschäftigte sich mit der Umsetzung von Werkzeugen des Qualitätsmanagements in der Softwareentwicklung. Das Ziel war die Implementierung von präventiven Fehlerverhütungsmaßnahmen. Wesentliche Untersuchungsobjekte stellten die in vielen Industriebereichen sehr erfolgreich angewandte FMEA (Fehlermöglichkeits- und Einflußanalyse) sowie den Aufbau einer Fehlererkennungssystematik dar.

Fehler vermeiden statt beheben

Das MTM-Verfahren

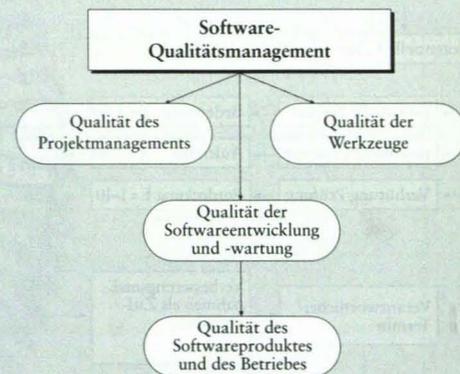
Dieses Verfahren und darauf aufbauende Themenbereiche wie Arbeitsplanung, Taktung von Montagelinien sowie ergonomische Arbeitsgestaltung und Arbeitssystemgestaltung stellen die Basis für die Softwareentwicklung durch adS dar. Das MTM-Verfahren selbst gliedert Bewegungsabläufe in ihre Grundbewegungen. Jeder dieser Grundbewegungen ist ein Normzeitwert zugeordnet, der in seiner Höhe durch die Ausprägungen der Einflußgrößen (vor-)bestimmt ist. Durch Anwendung dieses Verfahrens können Arbeitsabläufe vor Arbeitsbeginn detailliert festgelegt werden. Dem MTM-Prinzip der Ablaufplanung entsprechend, wird bereits im Planungsstadium die beste Methode bestimmt. Anstelle der Kostensenkung durch nachträgliche Arbeitsgestaltung, tritt die Kostenvermeidung durch vorherige Ablaufplanung.

Beim Aufbau von Planzeiten entfallen im Vergleich zu anderen Zeitermittlungsmethoden Einflußgrößenrechnungen und Leistungsgradbeurteilungen. Somit erhält man eine fundierte Basis zur leistungsabhängigen Entlohnung.

QM-Tools: Fehlerklassifizierung und FMEA

Ausgangspunkt für das Projekt war ein neuentwickeltes, bereits im Industrie-einsatz befindliches Industrial Engineering-Softwarepaket. Basierend auf Erkenntnissen und Erfahrungen der derzeit im Einsatz befindlichen Softwareprodukte beeinflussen die untersuchten Qualitäts-tools den Entwicklungs-, Umsetzungs- und Implementierungsprozeß neuer Produkte. Die Definition der „Qualität einer Software“ ist ein komplexer, durch

Abb. 1: Beziehungen des Software-Qualitätsmanagements [2]



objektive und subjektive Faktoren beeinflusster Vorgang, der sowohl den Entwicklungsprozeß als auch die Software selber beinhaltet.

Die Produktqualität drückt sich beispielsweise durch Zuverlässigkeit, Vollständigkeit und kurze Antwortzeiten aber auch durch die Güte der Dokumentation aus. Die Qualität der Entwicklung wird z. B. durch die Güte der Vorgaben und der verwendeten Werkzeuge sowie die Einhaltung von Richtlinien mitbestimmt.

Die anfängliche Definition der Fehler steht im Zentrum jeglicher Art von Qualitätsmanagement-Methoden. Ein Fehler ist, per definitionem, die Abweichung eines Sachverhalts von vorliegenden Normvorstellungen und somit immer ein relativer Begriff. Die Unterscheidung in Software- und Hardwarefehler erscheint auf den ersten Blick klar und unmißverständlich. Eine Klassifizierung der in diesem Grenzgebiet auftretenden Fehler (vgl.: Schutzverletzungen) wird aufgrund der verschwommener werdenden Nahtstellen jedoch immer schwieriger.

Für Softwaresysteme definiert der Benutzer die Bezugsnorm. Ein Unter-

nehmen, das Software entwickelt, muß für die Fehlerdefinition und -Klassifizierung eine durchgängige Regelung einführen, um subjektive Interpretationen und Mißverständnisse auszuschließen.

Fehlerklassifizierung

Das Sammeln qualitätsrelevanter Daten wird in allen Phasen des Produktlebenszyklus durchgeführt und beginnt bei der Zieldefinition in der Anforderungsphase, beinhaltet den Entwurf, die Implementierung und die Integration sowie den Betrieb und die Wartung.

Die entwickelte **Fehlererkennungssystematik** ist eine Methodik, die mittels Datensammlung, -auswertung und -analyse kausale Zusammenhänge zwischen Fehlern und Fehlerursachen aufdeckt. Man erhält Hinweise auf Möglichkeiten zur Vermeidung, frühzeitigen Entdeckung und Behebung von Fehlern. Diese Systematik stellt, bei richtiger Anwendung, ein Methodengerüst zur Verfügung, das hilft, die Produktqualität von Beginn an positiv zu beeinflussen und Fehlerfortpflanzungen vermindern.

Fehlerkataloge bieten die Möglichkeit, Verbesserungspotentiale aufzuzeigen. Fehlermeldungen, Testberichte und Hotlineaufzeichnungen werden gesammelt,

Die entwickelte Fehlererkennungssystematik ist eine Methodik, die mittels Datensammlung, -auswertung und -analyse kausale Zusammenhänge zwischen Fehlern und Fehlerursachen aufdeckt.