

# Ansatz zur nutzerzentrierten Requirement-Analyse und Evaluation: Ein Framework-Entwurf

Dirk Zimmermann  
T-Mobile Deutschland GmbH  
Usability Engineering Services  
Landgrabenweg 151  
53227 Bonn  
dirk.zimmermann@t-mobile.de

Lennart Grötzbach  
Siemens AG Medical Solutions  
Image & Knowledge Management  
Hartmannstr. 16  
91052 Erlangen  
lennart.groetzbach@siemens.com

Maximiliane Freymann  
Universität Paderborn C-LAB  
Interactive Dialog Systems  
Fürstenallee 11  
33102 Paderborn  
maximiliane.freymann@c-lab.de

## Abstract

Das Einbringen von Usability Engineering Ergebnissen in den Entwicklungsprozess stellt für viele Praktiker ein alltägliches Problem dar. Die Autoren beschreiben ein Framework aus verschiedenen nutzerorientierten Requirement-

Typen, das zur Abbildung von Analyse-Ergebnisse in einer für das Software-Engineering geeigneten Form sowie zur Auswahl angemessener Methoden der Evaluation im Entwicklungsprozess genutzt werden kann.

## Keywords

Usability Engineering, Requirement Engineering, Evaluation, Requirement-Typen, Methoden

## 1.0 Einleitung

In der Literatur gibt es bereits einige Ansätze, die sich mit der Integration von User-Centered-Design (UCD) Aktivitäten in die Software-Entwicklung (SE) beschäftigen. Die meisten UCD-Prozesse sind jedoch monolithisch und haben nur relativ wenig definierte Schnittstellen zu bestehenden SE-Prozessen. Im praktischen Einsatz muss jedoch versucht werden, die Ergebnisse von UCD-Aktivitäten bei der Software-Entwicklung zu berücksichtigen und Usability-Kriterien als relevante Qualitätskriterien zu etablieren. In aktuellen Software-Entwicklungsmodellen werden Requirements dazu genutzt, die funktionalen und nicht-funktionalen Anforderungen an Systeme zu spezifizieren. Funktionale Anforderungen definieren hierbei die eigentlichen Features des Systems, wogegen nicht-funktionale Anforderungen auf bestimmte Qualitäten hinauszielen. In der ISO/IEC 9126 [5] werden sechs Typen nicht-funktionaler Anforderungen beschrieben, eine davon ist Usability.

Es gibt im Bereich des Usability Engineering vergleichbare Bestrebungen, entsprechende Anforderungs-Arten zu definieren. Ein Beispiel hierfür ist die CISU-R [1], in der Richtlinien vorliegen, um Usability Anforderungen in ausrei-

chender Detaillierung zu definieren, die einen effektiven Beitrag zum Design und zur Entwicklung machen und ebenso empirisch überprüfbar sind. In Bezug auf die Evaluation ist CISU-R verwandt mit dem CIF [1], in dem ein Format zur Ergebnisbeschreibung von Usability-Evaluationen vorgestellt wird.

Usability als Produkteigenschaft wird in der Standardisierung ebenso wie in CISU-R und CIF definiert als Ausmaß, in dem ein Produkt von definierten Nutzern in einem definierten Nutzungskontext zur Erreichung definierter Ziele mit Effektivität, Effizienz und Zufriedenheit genutzt werden kann [9241/11]. Während der System-Entwicklung sind allerdings zusätzlich weitere Kriterien relevant, so zum Beispiel die sieben Dialogprinzipien der Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Erwartungskonformität, Steuerbarkeit, Fehlertoleranz, Lernbarkeit und Individualisierbarkeit [9241/10] zur Gestaltung der Nutzer-System-Interaktion. Im Bereich der Informationsgestaltung sind vor allem Kriterien wie Lesbarkeit, Verständlichkeit, Konsistenz, Unterscheidbarkeit, Erkennbarkeit, Prägnanz und Klarheit [9241/12] wichtig. Eine Definition von Usability-Anforderungen/Kriterien sollte die verschiedenen Dimensionen dieser Gestaltungsprinzipien berücksichtigen und eine effiziente Verwendung im Entwicklungsprozess ermöglichen. Dieses Paper stellt im ersten Teil ein konzeptionelles Framework vor, das ein Gerüst verschiedener Anforderungstypen darstellt, welche sowohl die verschiedenen Facetten des Usability Engineering repräsentieren, als auch den Software-Engineering-Anforderungen genügen. Üblicherweise werden im Software Engineering die zuvor spezifizierten Anforderungen im Rahmen der Evaluation überprüft; zum Beispiel inwieweit die Lösung und deren Zwischenergebnisse (funktional) korrekt und einsetzbar sind. Ebenso werden zuvor spezifizierte messbare Ziele überprüft, welche die Software erfüllen muss. Evaluationsaktivitäten können an verschiedenen Zeitpunkten innerhalb des Entwicklungsprozesses durchgeführt werden - meist werden dabei zuvor spezifizierte Anforderungen getestet. Wie von Woletz [8] beschrieben, fokussieren Evaluations-Aktivitäten dabei in verschiedenen Phasen auf verschiedenen Aspekten der Usability. So werden in frühen Phasen initiale Zwischenergebnisse genutzt, um eine bessere Anforderungsspezifikation zu erreichen oder Schwachstellen im grundlegenden Kon-

zeptionelles Framework vor, das ein Gerüst verschiedener Anforderungstypen darstellt, welche sowohl die verschiedenen Facetten des Usability Engineering repräsentieren, als auch den Software-Engineering-Anforderungen genügen. Üblicherweise werden im Software Engineering die zuvor spezifizierten Anforderungen im Rahmen der Evaluation überprüft; zum Beispiel inwieweit die Lösung und deren Zwischenergebnisse (funktional) korrekt und einsetzbar sind. Ebenso werden zuvor spezifizierte messbare Ziele überprüft, welche die Software erfüllen muss. Evaluationsaktivitäten können an verschiedenen Zeitpunkten innerhalb des Entwicklungsprozesses durchgeführt werden - meist werden dabei zuvor spezifizierte Anforderungen getestet. Wie von Woletz [8] beschrieben, fokussieren Evaluations-Aktivitäten dabei in verschiedenen Phasen auf verschiedenen Aspekten der Usability. So werden in frühen Phasen initiale Zwischenergebnisse genutzt, um eine bessere Anforderungsspezifikation zu erreichen oder Schwachstellen im grundlegenden Kon-

Konzept zu identifizieren, wogegen in späteren Phasen ein summarischer Test des Systems gegen zuvor spezifizierte Usability-Ziele erfolgt. Aufgrund der unterschiedlichen Zielsetzung und der Unterschiedlichkeit der evaluierten Konzepte/Prototypen/Systeme sind jeweils auch verschiedene Methoden und Ansätze zur Evaluation sinnvoll. Im zweiten Teil des Papers werden die Autoren daher herleiten, in wie weit sich die verschiedenen Anforderungstypen idealerweise auch mit spezifischen Methoden überprüfen lassen und somit Hilfestellung geben, zu jeder Zeit im Entwicklungsprozess effektive und effiziente Vorgehensweisen zur Evaluation zu wählen.

#### 1.1 Framework-Ansatz

Die Autoren nutzen das Prozessmodell, der DIN EN ISO 13407 [3] als Grundlage für die Definition eines Frameworks von nutzerzentrierten Anforderungen [9]. Das Framework bezieht sich auf die Phasen des Modells (*Context of Use*, *User Requirement*, *Produce Design Solution* und *Evaluate Design*) und beschreibt wann es innerhalb des Prozesses sinnvoll ist, nutzerzentrierte Anforderungen zu generieren, zu verwenden und zu evaluieren, und welche Eigenschaften diese Anforderungen besitzen sollen.

Entsprechend der ersten drei Phasen des DIN EN ISO 13407 Modells werden im Frameworks drei Arten von Anforderungen generiert: *Usability Requirements*, *Workflow Requirements* und *User Interface (UI) Requirements*. Diese können während der vierten Phase des Prozessmodells, der Evaluationsphase, überprüft werden.

Die erste Anforderungsart, die *Usability Requirements*, werden während der *Context of Use*-Phase erhoben. Sie spezifizieren übergreifende, die Usability des späteren Systems betreffende, Anforderungen und beschreiben messbare

Kriterien, die sich auf die Effektivität, Effizienz und Zufriedenheit des späteren Systems beziehen.

Während der *Context of Use* Phase werden die späteren Nutzer des Systems mit ihren Aufgaben und Zielen analysiert, die antizipierte Systemnutzung und der Nutzungskontext erhoben. Usability Requirements werden aus den Ergebnissen der Kontextanalyse ermittelt und aus weiteren, grundlegenden Informationen, wie zum Beispiel aus Verbesserungsvorschlägen, aus Analysen der Konkurrenzsysteme, aus dem generellem Wissen von Mensch-Maschine-Interaktions-Prinzipien, oder ähnlichem. Dokumente, die solche Informationen enthalten, können zum Beispiel Coopers Persona Beschreibungen [...] sein, die Eigenschaften der Nutzergruppen archetypisch zusammenfassen und einen ersten Überblick über typische Nutzeraufgaben geben, oder die Contextual Design Models von Beyer und Holtzblatt [...] (Flow of Work, Sequence of Work, Work Artifact, Work Culture and Physical Model), in denen die Ergebnisse einer Kontextanalyse grafisch dargestellt werden können. Obwohl ein Großteil dieser Ergebnisse deskriptiv sind, da sie bei der Auswahl und Erzeugung der Nutzungs- und Systemmodellen unterstützen und sind keine "harten" Anforderungen sind, können aus ihnen doch die Nutzererwartungen bezogen auf die Effektivität und die Effizienz des Systems abgeleitet werden und Faktoren, die zu einer zufrieden stellenden Nutzung führen, ermittelt werden.

Ein Beispiel für ein *Usability Requirement* ist zum Beispiel die Forderung, dass die Hauptziele der Nutzer sich in höchstens der Hälfte der bisherigen Zeit (ohne System Einsatz) erreichen lassen und dass dabei 10% mehr Ergebnisse produziert werden.

Die zweite Anforderungsart sind die so genannten *Workflow Requirements*. Sie beschreiben, wie das zukünftige System die Arbeitsabläufe der Nutzer unterstützen soll. Sie werden während der *User Requirements* Phase des ISO-Prozesses definiert, in der die Analyseergebnisse der vorhergehenden Phase betrachtet und iterativ in (verbesserte) Arbeitsabläufe des zu entwickelnden Systems überführt werden. *Workflow Requirements* beschreiben, wie die einzelnen Interaktionsschritte der Nutzer mit dem System aussehen und dokumentieren auf diese Weise die Workflows, die für zur Erfüllung der Nutzeraufgaben nötig sind. Sie detaillieren die Schritte der Nutzerinteraktion mit dem System und die dazugehörige Systemunterstützung. Mit Hilfe dieser Requirements wird festgelegt, wie die optimale Unterstützung der Nutzer vom System zu erreichen und umzusetzen ist. Zur Dokumentation der Interaktionsschritte können zum Beispiel Use Cases oder vergleichbare Dokumente (z. B. Szenariobeschreibungen) genutzt werden: "A scenario is a concrete description of a specific flow of interaction, but one that is chosen to be typical or representative. [...] A use case is a generic scenario, describing one kind of interaction with a particular user interface." [10]. In dem Framework werden daher die Szenarien, welche die benötigten Workflows des zu erstellenden Systems beschreiben, als die Menge der *Workflow Requirements* angesehen.

*UI Requirements* werden während der *Produce Design Solution* Phase generiert. Sie definieren und spezifizieren die Logik, die Inhalte und die Details des User Interfaces aus Nutzersicht – das heißt, sie beschreiben die Elemente des User Interfaces und die Systembereiche. Mit ihrer Hilfe lassen sich das Verhalten, das Look and Feel, die Art und Eigenschaften der Interaktion und alle möglichen Zustände, Voreinstellungen und

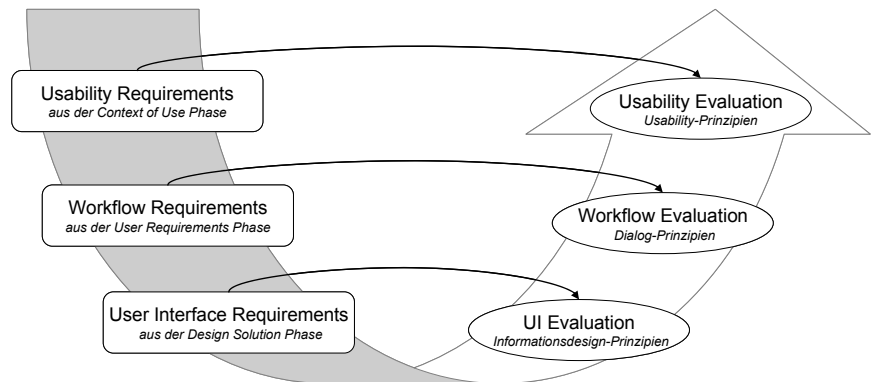
Systemreaktionen definieren. Verschiedene Arten von Anforderungen, die sich auf die oben genannten Bereiche beziehen, lassen sich unterscheiden:

- Anforderungen, welche die Informationsarchitektur und den Informationsfluss beschreiben. Mit ihrer Hilfe wird die dem User Interface zugrunde liegende logische Struktur definiert.
- Anforderungen, mit denen beschrieben wird, wie Systemkomponenten oder einzelne -elemente (z. B. ein Screen oder ein Widget) aussehen sollen.
- Anforderungen, mit denen Zusammenhänge zwischen einzelnen Elementen beschrieben werden - beispielsweise wie Drag&Drop innerhalb des Systems funktioniert.
- Anforderungen, mit deren Hilfe die Nutzer-System-Interaktion beschrieben wird.

Anforderungen, mit denen die verschiedenen Systembenachrichtigungen (Warnungen, Fehlermeldungen, etc.) beschrieben werden. Um sicherzustellen, dass das entwickelte System mit den Anforderungen der *Context of Use* und der *Specify Requirements* Phase übereinstimmt, werden die *UI Requirements* gegen die *Usability* und die *Workflow Requirements* evaluiert. Auf diese Weise wird sichergestellt, dass sowohl das mentale Model, welches der Nutzer von seiner Arbeit hat, als auch seine Erwartungen, bezogen auf die Nutzungsfreundlichkeit, sich in der Lösung wieder finden. Außerdem sollte die Lösung konform zu allgemeingültigen Styleguides, wie z. B. der DIN EN ISO 9241 Teil 12 sein.

### 1.2 Auswahl von Evaluationsmethoden basierend auf Usability-Kriterien

Im UE existieren verschiedene Kriterien, um die Usability einer Lösung zu beschreiben. Die am meisten verbreitete findet sich in der DIN EN ISO 9241-11, in der Usability als "das Ausmaß, in dem ein Produkt durch bestimmte Nutzer in



einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen". Im heutigen Gebrauch existieren verschiedene Usability-Evaluationsmethoden, welche diese drei verschiedenen Usability Kriterien Effektivität, Effizienz und Zufriedenheit überprüfen. Um die Usability eines Systems zu garantieren müssen alle drei Kriterien evaluiert werden – beispielsweise mit Hilfe eines inkrementellen Ansatzes, bei dem zuerst die Effizienz des Systems mit den entsprechenden Methoden überprüft wird, dann die Effektivität (oder anders herum) und abschließend die Zufriedenheit der Nutzer, um alle drei Usability Kriterien abzudecken.

### 1.3 Auswahl von Evaluationsmethoden basierend auf dem Prozessfortschritt

Der Fortschritt eines Entwicklungsprozesses wird sowohl in der Abfolge der Prozessphasen als auf der Verfeinerung der Zwischenergebnisse im Prozess reflektiert. Jede Phase beinhaltet mehrere Aktivitäten, die sich in der Verbesserung der Qualität einer Lösung oder der Quantität widerspiegeln. Ein idealer Entwicklungsprozess (aus Sicht von sowohl SE als auch UE) muss die Vollständigkeit, Korrektheit und die Gebrauchstauglichkeit (die durch die relevanten Kriterien dargestellt wird) garantieren.

Abbildung 1: Nutzerzentrierte Requirementarten und Evaluationsaktivitäten

Im SE gibt es eine Reihe verschiedener Methoden, welche die Vollständigkeit und die Korrektheit einer Lösung überprüfen. Auf Prozessebene wird das durch Standards wie z. B. der ISO / IEC 12207, die ein Prozessrahmenwerk für Entwicklungslebenszyklen definiert [6], sichergestellt. Im UE beschreibt gibt es die oben bereits erwähnte DIN EN ISO 13407, in der ein idealisiertes, standardisiertes Prozessmodell beschrieben wird.

Durch Prozessmodelle wird klar definiert, wann und wo Evaluationsaktivitäten stattfinden, d. h. wo die Arbeitsergebnisse früherer Phasen überprüft werden. Sie definieren jedoch nicht, welche Evaluationsaktivitäten durchgeführt werden sollen, und wie diese angewendet werden – es wird nur eine kontinuierliche Evaluation während allen Iterationen des Prozessmodells gefordert. Sowohl für SE als auch UE, die Anforderungen der kleinste gemeinsame Nenner für die Entwicklung einer Lösung, da diese Ansprüche an das System darstellen, die während des Prozesses getestet werden können. Weiterhin definiert die DIN EN ISO 13407, dass Evaluation gegen Anforderungen stattfinden muss, was mit der Idee einhergeht, dass in UE Anforderungen die Basis für Entwicklungsaktivitäten sind.

*Usability Requirements* können schon während ihrer Erhebung mit Hilfe von Interviews und Reviews verifiziert und z. B. durch Usability Tests und Fragebogen evaluiert werden. Die Anforderungen können von der Testabteilung zusammen mit anderen nicht-funktionalen Anforderungen getestet werden. Dabei stehen besonders die Usability-Prinzipien aus der DIN EN ISO 9241-11 im Vordergrund.

*Workflow Requirements* können mit Nutzern untersucht und verifiziert werden, um zu überprüfen, inwieweit die aktuelle Beschreibung des Workflows korrekt ist, und ob der optimierte Workflow eine tatsächliche Verbesserung ihrer Aufgaben darstellt. Später im Entwicklungsprozess können die Requirements dazu genutzt werden, einen Prototypen oder ein System zu evaluieren. Dabei wird untersucht, ob der Workflow unterstützt wird, d. h. ob das Interaktionsmodell mit den Workflow Requirements übereinstimmt, und ob dabei die generellen Dialogprinzipien aus der DIN EN ISO 9241-10 beachtet werden.

Mit Hilfe z. B. von Fragebogen, Cognitive Walkthroughs und Usability Tests kann untersucht werden, inwiefern das User Interface die Anforderungen aus Nutzersicht repräsentiert, also die UI Requirements widerspiegelt. Die DIN EN 9241-12 definiert dafür Information Design Guidelines. Entwickelte Designlösungen können mit Nutzern überprüft werden, um herauszufinden, ob ihre Workflows unterstützt werden (die in den zuvor definierten Workflow Requirements dokumentiert sind) und auf der anderen Seite gegen messbare Kriterien, von denen die Usability Requirements einen Teil darstellen. Da also die drei Requirementarten aufeinander aufbauen, zielen Evaluationsmethoden, mit denen Workflow und UI Requirements überprüft werden, immer auf die grundlegenden Usability Kriterien Effektivität,

Effizienz und Zufriedenheit ab, jedoch bilden diese drei Kriterien nur eine Bewertungsdimension.

ungen), kann die Usability der Lösung sichergestellt werden.

Durch die eingebettete Evaluationspha-

Tabelle 1: Evaluationsmethoden nach Usability Kriterien und Anforderungsarten gruppiert

	Usability Requirements	Workflow Requirements	Require-ments	UI Requirements
<b>Effektivität</b>	Use Case Validierung Beobachtung	Cognitive Walkthrough Heuristische Evaluation Review mit SME UTAPAM Keystroke GOMS		Cognitive Walkthrough Heuristische Evaluation Experten Review UTAPAM Usability Tests Performance Tests
<b>Effizienz</b>	Use Case Validierung Focus Gruppen Interview	Focus Gruppen Think-Aloud Interview		Interview Fragebogen SUMI AttrakDiff
<b>Zufriedenheit</b>	Fragebogen Interview Must-Have Requirements			

Durch die Einführung von *Workflow- und UI Requirements* wurde das etablierte Konzept von nutzerzentrierten Anforderungen erweitert. Die beiden neuen Arten erlauben eine genauere Darstellung von UCD Anforderungen, die in den späteren Phasen eines Entwicklungsprozesses entstehen. Das Framework erlaubt eine Hierarchie von Anforderungen, die zum einen schrittweise verfeinert werden und zum anderen rückverfolgbar sind, angefangen mit grundlegenden Anforderungen, die den allgemeinen Gebrauch des Systems adressieren, gefolgt von speziellen Anforderungen, die an die Aufgabenbearbeitung gestellt werden, bis hin zu Anforderungen, welche die spezifischen Implementierung des Systems betreffen.

Das Framework bietet weiterhin eine Anleitung, abgestimmt auf die Ergebnisse, die innerhalb der Phasen der DIN EN ISO 13407 entstehen, und damit auf den Fortschritt eines Projekts abzielen, der in den drei verschiedenen Anforderungsarten widerspiegelt wird. Durch die Verbindung der Requirement-Arten zu den entsprechenden Ergebnissen (Anforde-

se, wie sie in der DIN EN ISO 13407 definiert wird, und durch die Definition von testbaren Ergebnissen, wird es einfacher, die passende Evaluationsmethode für jeden Zeitpunkt innerhalb des Entwicklungszyklus' zu identifizieren. Tabelle 1 zeigt eine Teilmenge von Evaluationsmethoden, die nach den Usability-Kriterien, die sie adressieren (für Usability Requirements) und den Ergebnissen des Entwicklungszyklus (beschrieben als nutzerzentrierte Requirements), auf die sie angewendet werden können, gruppiert sind.

Durch diese Unterscheidung wird die Auswahl vereinfacht, und somit eine kontinuierliche Evaluation der Usability während des Prozesses attraktiver.

#### 1.4 Nutzen

Durch den Einsatz des Frameworks können SE-Organisationen eine Hierarchie von nutzerzentrierten Anforderungen einsetzen, die zuerst übergreifende Anforderungen für den Einsatz der Software definieren, anschließend spezifische Anforderungen für die Aufgabendurchführung festlegen und schließlich detaillierte Anforderungen für die Umsetzung und Realisierung spezifizieren.

Zudem erleichtert die Unabhängigkeit des Frameworks von spezifischen UCD- und SE-Prozessmodellen sowohl die Skalierbarkeit des Vorgehens als auch die Anpassbarkeit an existierende Rahmenbedingungen und individuelle Gegebenheiten.

Die Auswahl geeigneter Evaluationsmethoden unterstützt, dass im Rahmen der allgemeinen Qualitätssicherung die jeweiligen Anforderungen überprüfbar sind und auch geprüft werden.

Auf diese Weise können einerseits nutzerzentrierte Anforderungen (und damit UCD-Aktivitäten) gezielt in bestehende SE-Prozesse eingebunden und andererseits so Methoden zur Überprüfung der Anforderungen bereitgestellt werden, die eine Evaluation der Gebrauchstauglichkeit der zu entwickelnden Software ermöglichen.

## 2.0 Literaturverzeichnis

[1] CISU-R. Common Industry Specification for Usability-Requirements, Draft Version 0.86, National Institute of Standards and Technology, 18.03.2006.

[2] DIN EN ISO 9241, parts 10-12. Ergonomic requirements for office work with visual display terminals (VDTs). International Organization for Standardization, 1998.

[3] DIN EN ISO 13407. Human-centered design processes for interactive systems, International Organization for Standardization, 1999.

[4] DIN IEC TR 18529. Software engineering - Product quality. International Organization for Standardization/International Electrotechnical Commission, 2001.

[5] ISO/IEC 9126. Software Engineering - Product Quality, International Organization for Standardization, 2001.

[6] ISO/IEC 12207: 1995. Information technology -- Software life cycle processes.

[7] ISO/IEC 25062:2006. Software product Quality Requirements and Evaluation (SQuaRE) - Common Industry Format (CIF) for usability test reports, 2006.

[8] Woletz, N.: Evaluation eines User-Centred Design-Prozessassessments - Empirische Untersuchung der Qualität und Gebrauchstauglichkeit im praktischen Einsatz. Dissertation. Universität Paderborn, Deutschland, 2006.

[9] Zimmermann, D., Grötzbach, L.: A Requirement Engineering Approach to User Centered Design. HCI International 2007 – 12th International Conference on Human-Computer Interaction, Beijing, China, 2007.

[10] Constantine, L: What do users want? Engineering Usability into Software, Windows Tech Journal , December 1995.

