

Alltagstauglichkeit im Kontext der Migration von Anwendungssystemen

Oliver Bohl, Andreas Höfer, Shakib Manouchehri, Udo Winand

Universität Kassel
Fachgebiet Wirtschaftsinformatik
Nora-Platiel-Strasse 4
34127 Kassel

{bohl, hoefer, manouchehri, winand}@inf.wirtschaft.uni-kassel.de

Abstract: Aktuelle Entwicklungen in den Bereichen der Informations- und Kommunikationstechnologien (IKT) und von IT-Systemen sowie -Architekturen führen zum Entstehen neuartiger oder zur Anreicherung bestehender Anwendungssysteme (AWS). Getrieben werden die (Fort-)Entwicklungen dabei oftmals aus einer technisch-orientierten Perspektive heraus, beispielsweise durch die Umstellung auf service-orientierte Architekturen oder durch die Nutzung ubiquitär verfügbarer Infrastrukturen. Häufig bedingen diese Neuerungen eine Umstellung bisheriger Schnittstellen zwischen dem IKT-gestützten System und seinem Nutzer.

1 Einleitung

Neue oder fortentwickelte Nutzerschnittstellen und die mit ihnen verbundenen Möglichkeiten zur Etablierung innovativer, häufig technisch und funktional sehr komplexer AWS werden bei dem Übergang von alten in neue AWS zumeist mit dem Versprechen der verbesserten Anpassung an ihre (potenziellen) Nutzer, speziell in Bezug auf ihre Gewohnheiten, ihre Erwartungen sowie ihren Kontext [Ta03], entwickelt. Dabei wird in der Regel eine verstärkte Korrelation mit dem alltäglichen privaten und beruflichen Habitus und damit eine Reduzierung der nutzerseitig wahrgenommenen Nutzungskomplexität angestrebt. Die AWS sollen den zunehmend komplexen Erfordernissen entsprechen und gleichfalls konform zu den bisherigen Verhaltensweisen des beruflichen aber auch des privaten Alltags ausgestaltet werden. Das zentrale Ziel einer solchen Orientierung an etablierten Prozessen des Arbeits- und Privatlebens ist die Vereinfachung der Nutzbarkeit der Anwendung aus Nutzerperspektive. In diesem Kontext ist jedoch ein Spagat zwischen dem idealisierten Abdecken notwendiger Aktivitäten und Prozesse, ggf. auch von solchen, welche bislang nicht durch AWS unterstützt werden (quasi eine revolutionäre Gestaltung von AWS) und den unter Umständen durchaus als suboptimal angesehenen, aber bereits etablierten IKT-unterstützten Aktivitäten (eine eher evolutionäre Gestaltung von AWS) zu vollziehen.

Dies verdeutlicht einen wesentlichen Aspekt der alltagstauglichen Gestaltung von AWS: die Bewahrung bestehenden, teils auch durch suboptimale Nutzungs- oder Verhaltensweisen determinierten Know-hows. Das alltägliche Verhalten bei der Bewältigung von Aktivitäten und dem Mitwirken an Prozessen verschiedener Natur ist neben optimierten Handlungs- und Verhaltensweisen in der Regel auch durch (teils nur vermeintliche) Unzulänglichkeiten bei der Bedienung und Nutzung bestehender AWS geprägt.

Im Rahmen der Entwicklung und Gestaltung von AWS werden diese suboptimalen Zustände häufig jedoch nicht etwa expliziten Analysen unterworfen, sondern vielmehr durch idealisierte Rollen ersetzt und eine ggf. auch kontraproduktive Konsistenz der zu etablierenden AWS mit diesen optimierten Zuständen eingefordert. Dabei werden dem Entwurf von AWS vielfach und anstelle einer konsequenten Ausrichtung an alltägliche, teils sicherlich nicht optimal gestaltete Nutzungsweisen, mechanisch wirkende Anforderungen an ihre Konsistenz zu Grunde gelegt. Grudin (1989) verdeutlicht drei wesentliche Kriterien in Bezug auf die Benutzeroberfläche: „User interface consistency is used in three interrelated senses: the internal consistency of a design with itself; the external consistency of a design with other interface designs familiar to a user; and an external analogic or metaphoric correspondence of a design to features in the world beyond the computer domain.“ [Gr89] Gleichsam erläutert er die hierdurch entstehende Problematik: “A close examination suggests that consistency is an unreliable guide and [...] designers would often do better to focus on users’ work environments.” Er schlägt als Alternative einen Perspektivenwechsel bei der Gestaltung von Benutzeroberflächen vor. Der grundlegende Konflikt zwischen der Suche nach den sichtbaren Eigenschaften der Nutzeroberfläche sowie der nutzerorientierten Gestaltung und Tätigkeitsanalyse kann andernfalls leicht übersehen werden. Grundin folgert schlussendlich, dass ein besseres Verständnis für die Nutzer und ihre täglichen Aufgaben etabliert werden soll [Gr89].

Seit einiger Zeit finden zunehmend entsprechende nutzer- und alltagsorientierte Betrachtungen Einzug in die Gestaltungsparadigmen von AWS. Exemplarisch kann an dieser Stelle Tarasevich angeführt werden, welcher ausführt: „Developers need to understand people and how they interact with their surroundings.“ [Ta03, S. 60] Dies betrifft speziell den Einbezug von im beruflichen und privaten Alltag erworbenem Wissen und bestehender Nutzungskompetenzen und geschieht vor der Prämisse der alltagstauglichen Gestaltung von AWS. Diese alltagskonforme Gestaltung sowie die Anpassung an die bisherigen Verhaltensweisen werfen speziell bei der Migration von AWS Fragen auf, die im Folgenden behandelt werden.

2 Alltagstauglich, Alltagswissen und Nutzungskompetenz

Alltagstauglich wird in Duden (2007) als „sich [gut] für die alltägliche Verwendung eignend“ definiert. Zunehmend ist der Alltag in weiten Teilen der Welt geprägt durch den Umgang mit Technologien. Dabei ist, neben der Feststellung, dass „unser Denkvermögen (...) für eine Umwelt entwickelt (wurde), die wesentlich schlichter und überschaubarer strukturiert war als die, in der wir heute leben“ [Dö97], auch festzuhalten, dass unser Bestand an Alltagswissen eben nicht unveränderbar ist. Die Drohung an den eigenen PC, ihm notfalls den Strom abzdrehen, also das verinnerlichte Wissen, dass ohne Strom der PC nicht mehr laufen wird, ist sicher keine Erkenntnis, die dem modernen Menschen in den Genen schlummert [Wi00].

Und so formt das Lebensumfeld auch mit der Zeit das Alltagswissen. Der Mensch erwirbt beständig neues Wissen, optimiert dabei seine Annahmen über Wirkungsmechanismen. Und so ist Alltagswissen kein fester Satz von Routinen, sondern formt sich im Verlauf der Zeit, differenziert sich auch, je nach den eigenen Erfahrungen. Für einigermaßen computerliteraten Nutzer stellt sich manches, was anderen als Geheimwissen vorkommen kann, als vollkommen alltagstauglich dar: Der Klick auf das Kreuz oben rechts schließt das Fenster, F1 öffnet die Hilfe oder die Menüleiste ist im oberen Bereich angeordnet.

Dies alles ist Routinewissen, welches sich erst mit der Zeit erlernen lässt, von da an aber für diejenigen Leute, welche dieses Wissen erworben haben, vollkommen alltäglich sein mag. Bereits geringfügige Änderungen, also beispielsweise eine veränderte Position, können zu Verwirrungen und Benutzungsproblemen führen. Sachen, die für den einen eine Geheimwissenschaft darstellen, erschließen sich auch bei neuen Anwendungssystemen anderen sofort aufgrund dieser erworbenen Annahmen über die Arbeits- und Funktionsweisen von AWS dieser Art. Dieses Wissen kann sich aus unterschiedlichen Quellen speisen (aus formellen und informellen Lernarrangements) und verschiedenartige Ausprägungen annehmen. Die Abgrenzung zwischen den nachstehend vorgestellten Wissens- und Know-how-Kategorien ist bewusst unscharf gehalten, sie dienen entsprechend nur als Anhaltspunkte:

- **Faktenwissen:** Der Anwender weiß genau, dass er für das beabsichtigte Ergebnis bestimmte Schritte durchführen muss. Das hat er mit der Zeit gelernt, vielleicht in formellen Lernszenarien bspw. in Schulungen oder durch eher informelle Lernarrangements wie durch das Learning-by-Doing.
- **Gewöhnung:** Durch den Umgang hat er sich an bestimmte Besonderheiten des Programms auch dann gewöhnt, wenn sie nicht unbedingt logisch oder suboptimal gestaltet sind und mit seinem Alltagswissen nur in Teilen oder nicht korrelieren.
- **Transferwissen:** Der Anwender transferiert Problemlösungsmechanismen, die er in anderen Feldern gelernt hat, vielleicht auch außerhalb des gesamten Umgangs mit AWS, und überträgt diese Problemlösungsmechanismen auf die Problemstellungen in dem AWS.

Im Kern gilt, dass der Anwender seine Nutzungskompetenz für das gegebene AWS sicherlich nicht nur aus der Bedienungsanleitung zieht, sondern dass er auch aus anderen Erfahrungsbereichen, also auch und insbesondere aus dem Alltag, einen Transfer dort gültiger Annahmen über Ursachen- und Wirkungszusammenhänge vornimmt. Für jeden Anwender ist diese Nutzungskompetenz unterschiedlich ausgeprägt und auch der Weg, auf dem er diesen, seinen Stand an Nutzungskompetenz erwarb, wird von Nutzer zu Nutzer unterschiedlich sein. Für den einen mag die Nutzungskompetenz für ein AWS in erster Linie durch das intensive Lernen im engeren, eher faktenorientierten Sinne entstanden sein, während andere aufgrund ihres alltäglichen Umgangs mit Computern im Allgemeinen oder ähnlich gelagerten AWS oder auch vollkommen anderen Themenfeldern, deren Erkenntnisse aber transferierbar sind, ihre Kompetenz erwarben.

3 Alltagstauglichkeit im Kontext der Gestaltung von AWS

Betrachtet man AWS, die zunehmend den betrieblichen und privaten Alltag prägen, gerät die Nutzung zum permanenten Intelligenztest, wenn nicht sogar zum Erschrecken, zur Frustration bis hin zur Verweigerung [Wi00]. Gleichfalls lässt sich eine Art von Gewöhnung der Nutzer an diese disfunktional gestalteten AWS konstatieren.

AWS als „Mensch-Maschine-Interaktionssysteme, die medien-, kommunikations- und informationstechnische Komponenten umfassen und Individuen, Gruppen und Institutionen bei der Bewältigung der täglichen Aufgabenprozesse unterstützen“ [Wi00], sind charakterisiert durch hohe Komplexität, Kontingenz und Situationsbestimmtheit. Ihre Planung und Realisierung basieren auf Vorgehensmodellen unterschiedlicher Art, die u.a. Ausdruck von situativer Problemstellung, von Entwurfsstrategien und –prinzipien sind. Damit steigt bei der Entwicklung von neuen AWS abermals die Notwendigkeit, möglichst störenden Einflüssen, Fremdheitsempfindungen, Misstrauen und Überforderungen vorzubeugen [Wi00]. Die Analogie zu den gewohnten, bekannten, bewährten Lebens- und Arbeitswelten sowie mit dem Alltagswissen korrespondierenden, entgegnet Aversionen, die gegen eine Nutzung entstehen können. Demnach wird die Wahrnehmung von Alltagstauglichkeit durch Personen und/oder Organisationen im Spiegel ihrer Erwartungen und geprägt durch ihr Alltagswissen realisiert. Die Adressaten sind vor allem die Komponenten von AWS und Diensten und die an sie gerichteten Alltagserwartungen. Alltagstauglichkeit kann weiterhin manipuliert werden, ist situationsgebunden (im Unternehmen anders als in der Privatsphäre), bildet ein funktionales Äquivalent zur Handhabung von Komplexität und Ungewißheit und leistet einen Beitrag zur Zielerreichung [Wi00].

Winand [2000] stellt Ausgangsthesen für den Einbezug von Alltagstauglichkeit als Paradigma bei der Gestaltung von AWS zusammen und schreibt, dass Alltagstauglichkeit für alle Facetten von AWS (wie für Hardware, Software, Infrastrukturen, Organisations- und Prozess-Strukturen, Entwurfsmethoden, Informationsinhalt etc.) thematisiert werden muss. Er definiert darauf aufbauend Anforderungen an alltagstaugliche AWS und lehnt sich dabei an Paradigmen des stufenweise Herantastens, des Ausprobierens sowie des Reflektierens von Bewährtem an. Demnach müssen alltagstaugliche AWS:

- Reaktionen bzw. Interaktionen hervorbringen, die aus dem Alltagsgeschehen heraus analogisierbar sind,
- bei Störungen, die erwartet werden können, beherrschbar bleiben,
- sowie sich in die generellen Bedürfnisstrukturen und Wunschvorstellungen einfügen.

Keil-Slawik [2000a] versteht unter Alltagstauglichkeit und Alltagspraxis die Entwicklung komplexer technischer Konfigurationen sowie die Evaluierung ihrer Herstellung und auch ihre Nutzung unter Alltagsbedingungen als Forschungsstrategie innerhalb des Fachgebietes Informatik und Gesellschaft. Er stellt, im Kontext multimedialen Lernens, vier Aspekte der Alltagstauglichkeit in den Vordergrund der evolutionären AWS-/Systemgestaltung: Ressourcenneutralität, Übertragbarkeit, Offenheit sowie Nachhaltigkeit [Ke00b]. Damit wird Alltagstauglichkeit selbst zum Innovationsziel [Ke00a].

Der Grad der Alltagstauglichkeit reflektiert unterdessen inwieweit unter gegebenen Randbedingungen die Komponenten des AWS und die Kombination dieser Komponenten den Alltagserwartungen/-wissen der Betroffenen nahe kommt, inwieweit sie auf Vertrautes stoßen, das es ihnen erleichtert, das AWS in ihre Arbeits- und Entscheidungsprozesse zu integrieren und damit die Funktionalität des AWS abzusichern und auszuschöpfen [Wi00]. An dieser Stelle stellt sich die Frage danach, wie die Nutzungskompetenz der Nutzer und somit die Alltagstauglichkeit von AWS bei dem Übergang von einem Zustand in den nächsten bewahrt oder sogar überführt, migriert werden kann. Ansätze von Migration sollen hierbei Ansätze zur Erreichung des Gesamtzieles der alltagstauglichen Gestaltung von AWS bieten.

4 Migration

Bedingt durch technologischen Wandel, durch veränderte Bedürfnisse oder auch durch strategische Ausrichtungen (Kunden-, Prozess-, Wissensorientierung, Integration und Vernetzung), erfordern Veränderungsprozesse in Wirtschaft und Gesellschaft ein spezielles Management von Wandlungsprozessen [St98]. Dabei sind erfolgreiche Wandlungsprozesse nur möglich, wenn gesellschaftliche und technische Prozesse sowie Systeme aufeinander abgestimmt werden.

Mit dieser Problematik des Übergangs zwischen zwei verschiedenen Prozessausprägungen beschäftigen sich, in jeweils anderen Schwerpunkten, diverse Forschungsgebiete wie das Migrationsmanagement, das Change Management, das Innovationsmanagement oder das Technologiemanagement. Dadurch wird der Begriff der Migration in ganz unterschiedlichen Kontexten verwendet und hat dadurch eine enorme Bedeutungsvielfalt erlangt. In der Informatik bezieht er sich beispielsweise auf die Ablösung oder das Upgrade von einem Anwendungs- oder Software-System auf ein potenziell besseres [Si92]. Konzeptionelle Überlegungen zum Thema des Migrationsmanagements als Teilgebiet der Anwendungssystem (AWS)-Entwicklung und Implementierung finden sich auch aus wirtschaftsinformatischer Perspektive. Migration wird hierbei als Managementaufgabe betrachtet, bei der neben technischen auch organisatorische Fragestellungen eine wesentliche Rolle spielen [BMW07].

Die Migration als eine Methode der Erneuerung, die die zugleich auch auf Bewahrung des Bewährten zielt. Durch Migrationen können sich (partielle) Ablösungen von existierenden Anwendungssystemen durch Neuentwicklungen ergeben, es können sich jedoch gleichfalls Erweiterungen ergeben, welche zu großen Teilen auf Elementen des vorherigen Systems basieren [Dö98]. Bei Migration handelt es sich um eine Technik des Wandels, eine Arbeitsmethode, welche darauf abzielt, in einem gleichsam evolutionären statt revolutionären Prozess Verbesserungen zu ermöglichen und dabei eingeführte mit neuartigen, ggf. innovativen Systemkomponenten und Technologien zu einer funktional verbesserten und ökonomisch ausbalancierten Gesamtlösung zu verknüpfen.

Der Grundgedanke der Migration, also des Transfers von Bewährtem in neue Zusammenhänge, beschränkt sich unterdessen nicht ausschließlich auf informationstechnische Systeme. In Anlehnung an „Legacy Systems“ könnte man davon sprechen, dass vielerorts an „Legacy Processes“ festgehalten wird. Prozesse, die eigentlich obsolet sind oder zumindest angepasst werden sollten oder sogar müssten, wobei jedoch Prozesse und Systeme derart eng aufeinander abgestimmt sind, dass sich weder das eine, noch das andere ohne erhebliche Kosten und Risiken verändern lässt. Diese nicht-technische Komponente ist bei Migrationsvorhaben keinesfalls zu vernachlässigen.

Migrationsstrategien dürfen also nicht nur auf Techniksysteme (Anwendungen, Architekturen, Konzepte, Entwürfe, Schnittstellen) bezogen sein, sondern müssen sich auch auf organisatorische Aufbaustrukturen, personale Qualifikationen (Humankapital), Prozesse und Funktionen erstrecken. Die Migration erfordert somit eine balancierte Gestaltung einer Vielzahl von Systemelementen. Eine Fokussierung auf einzelne Dimensionen ist ungeeignet, das Bündel positiver Migrationswirkungen zu realisieren. Migration kann dabei die Akzeptanz und Motivation der vom Systemwandel Betroffenen stärken. Migration kann durch Wiederverwendung von vorhandenen Qualifikationen, Know-how und Kompetenzen, von Strategien der Fehlervermeidung und von Prozesswissen die Chance bieten, Kosten zu senken und die Leistungsqualität zu steigern. Bestehende Geschäftsprozesse und -logiken können dabei zielgerichtet weiterverwendet werden [BMW07].

5 Auswirkungen von Migrationen auf die Nutzungskompetenz

Im Folgenden wird die Überführung eines Prozesses von einem Altzustand in einen Neuzustand betrachtet. Dabei wird allein auf den Teilbereich der durch AWS gestützten Prozessschritte in einen Neuzustand eingegangen. Ausgegangen wird dabei davon, dass im Rahmen der Überführung des Prozesses eine Veränderung der Mensch-Maschine-Schnittstelle, also beispielsweise eines grafischen Interface, notwendig ist. Um Erkenntnisse und Gestaltungsempfehlungen zu gewinnen, wird im Folgenden ein Modell der Nutzungskompetenz und ihrer Veränderung im Zuge eines AWS-Wechsels etabliert.

Demnach sei die Nutzungskompetenz eines Anwenders die Menge aller, theoretisch identifizierbaren, (Einzel-)Lösungskompetenzen, welche er im Umgang mit einem AWS erlangt hat. Diese lässt sich individuell beschreiben: So weiß ein gegebener Anwender beispielsweise bei einer Textverarbeitung, wie er einen Text öffnet, ihn editiert, ihn speichert und ihn ausdruckt. Formatierungen beherrscht er hingegen nicht und mit weitergehenden Funktionen wie beispielsweise Formatvorlagen oder Kopf- und Fußzeilen kennt er sich ebenfalls nicht aus.

Diese individuelle Nutzungskompetenz lässt sich nicht auf einen Wert verdichten und die Nutzungskompetenz zweier Nutzer lässt sich folglich nicht im Sinne einer „größer“ oder „kleiner“ Relation vergleichen, es sei denn die Nutzungskompetenz des einen Nutzers sei eine Teilmenge der Nutzungskompetenz des anderen Nutzers. Zum Beispiel mag Nutzer A ein Textverarbeitungsprogramm beinahe komplett beherrschen, nur mit der Serienbrieffunktion setzte er sich nie auseinander. Nutzer B beherrscht nun, neben Basisfunktionen auch den Umgang mit Serienbriefen, aber kennt sich ansonsten kaum aus. Keiner der beiden Nutzer hat damit per se und in Bezug auf das komplette Textverarbeitungsprogramm eine höhere Nutzungskompetenz als der andere. Allerdings mag es Anforderungen geben, die sich aus dem individuellen Bedarf der Nutzer ergeben, welcher wiederum zum Beispiel aus beruflichen Anforderungen resultiert. Diese Anforderungen wiederum stellen jeweils Mengen von Einzelkompetenzen dar, welche notwendig sind, um eine gegebene Problemstellung erfolgreich bearbeiten zu können. Anhand solcher Anforderungen lässt sich feststellen, ob ein Nutzer genug Nutzungskompetenz hat, um spezielle Problemstellungen bearbeiten zu können. In diesem Kontext mögen manche Einzelkompetenzen einem gegebenen Nutzer auch helfen, seine Produktivität zu erhöhen, indem er umständliche Wege vermeidet, zum Beispiel durch das Nutzen von Tastatur-Schnellkommandos oder VBasic-Makros.

In dem Zustand vor der Prozessüberführung, dem Altzustand, hat jeder Anwender ein individuelles Können bezüglich des Umgangs mit dem betrachteten AWS. Wird nun die technische Basis, auf welcher der Anwender bisher arbeitete, geändert, so hat das unweigerlich Auswirkungen auf seine Nutzungskompetenz. Dabei ist im ersten Moment, quasi im Moment des ersten Programmstarts, ein Verlust bzw. eine zukünftig nicht weiterhin realisierbare Nutzbarkeit von Nutzungskompetenzen feststellbar.

Das Wissen über die notwendigen Bedienoperationen zum Erreichen eines gewünschten Zieles geht in dem Maße verloren bzw. unterliegt einer Erosion, indem sich das neue AWS vom alten unterscheidet. Das gilt auch für solches Wissen, welches sich über Transferleistungen aus anderen, zum Teil alltäglichen, Bereichen ergibt und im Alzustand genutzt wurde. Im ersten Moment bedeutet jede Änderung der AWS-Mensch-Schnittstelle damit eine weiterhin nicht mögliche Verwendbarkeit bzw. gar einen Verlust von Nutzungskompetenz für den Anwender in jedem Bereich, in dem das neue AWS dem alten nicht exakt gleicht.

Beispielsweise bedeutet in der Regel ein übliches Versions-Upgrade einer Standardsoftware einige neue Funktionalitäten unter Beibehaltung aller alten. Die Benutzeroberfläche wird nur moderat modernisiert. Dabei mögen einige Menüs neu sortiert worden sein, aber schon die relative Ähnlichkeit der Benutzeroberfläche suggeriert eine gewisse Vertrautheit. Dem Nutzer ist es in solchen Fällen mit geringer Gewöhnungszeit möglich, sich den neuen Gegebenheiten anzupassen und sich mit der Zeit auch die neuen Funktionalitäten anzueignen. Dem gegenüber steht der Wechsel von einem AWS zu einem gänzlich anderen (Konkurrenz-)System. In vielen Fällen ist hier eine komplette Umstellung des Nutzerverhaltens erforderlich. Keine Schaltfläsche sieht mehr so aus wie zuvor und selbst versierte Nutzer brauchen mitunter Stunden, um sich alleine die Basisfunktionalitäten zu erschließen. Der Nutzer kann hierbei durch die fehlende Vertrautheit relativ früh verzweifeln, da der Verlust seiner Nutzungskompetenz ihm angesichts der völlig neuen Oberfläche besonders schwer aufzuholen erscheint.

Modellhaft lässt sich dabei unterstellen, dass jede einzelne Veränderung des Neusystems im Vergleich zum Altsystem einen eigenen, identifizierbaren Nutzungskompetenzverlust- bzw. -erosions-Beitrag zur Folge hat. Es ist plausibel anzunehmen, dass dieser Verlust sich anhand zweier Kriterien beurteilen lässt: *Verlusthöhe* und *Verlustdauerhaftigkeit*.

- **Verlusthöhe:** Wie schwerwiegend ist der Verlust an bzw. die Erosion von Nutzungskompetenz? Diese ist umso höher, desto verringerter die Produktivität im Anschluss an den Systemwechsel ist, also desto eher die verlorenen Nutzungskompetenzen für die spezifischen Aufgabenstellungen an diesen Nutzer relevant sind. Dieser kann jeglicher Produktivität beraubt sein, da er die beabsichtigten Ergebnisse seines Tuns gar nicht mehr erreichen kann, weil er kein Know-how abrufen kann, wie er diese erzielen könnte. Alternativ kann der Nutzungskompetenzverlust ihn lediglich dazu zwingen, etwas umständlicher als zuvor zu arbeiten.
- **Verlustdauerhaftigkeit:** Wie schwer ist es, die eingetretenen Nutzungskompetenzverluste wieder wettzumachen? Gelingt es dem Anwender schnell, sich an das neue AWS anzupassen und sein altes Nutzungskompetenzniveau wieder zu erreichen oder sogar zu übertreffen, dann ist die Dauerhaftigkeit eher gering. Ist dieses hingegen ein aufwendiger Prozess oder aber mit dem neuen AWS sogar objektiv unmöglich, so ist die Dauerhaftigkeit als hoch einzustufen.

Dieses Modell der Nutzungskompetenz beim AWS-Wechsel ist dazu geeignet, Empfehlungen für die Entwicklung von AWS zu entwickeln – unter dem Gesichtspunkt der AWS-Wechselfreundlichkeit im Speziellen, sicherlich mit etwas höherem Abstraktionsniveau für die Gestaltung von AWS im Allgemeinen. So lassen sich deutlich zwei verschiedene Ansatzpunkte für eine Erhöhung der Systemwechsel-Freundlichkeit ableiten:

- Verringerung der Verlusthöhe und -häufigkeit: Dieses bedeutet, dass sich das Neusystem von dem Altsystem erstens in möglichst wenigen Punkten hinsichtlich der Mensch-Maschine-Schnittstelle unterscheiden sollte. Zweitens sollten die Unterschiede gerade in den Bereichen, die besonders wichtig sind, um überhaupt produktiv arbeiten zu können, also bei den Basisfunktionalitäten, besonders gering sein.
- Verringerung der Verlustdauerhaftigkeit: Der Anwender muss möglichst dazu in der Lage sein, systemwechselbedingte Verluste seiner Nutzungskompetenz schnell wieder wettzumachen. Dieses kann dann erreicht werden, wenn beispielsweise die neuen Funktionalitäten beziehungsweise die veränderten Bedienelemente sich weitestgehend an gewohnte Umsetzungen anlehnen. Erste Erfolge bei dieser Umstellung durch derartige Vertrautheiten schaffen das Vertrauen, sich auch dann in tiefer gehende Möglichkeiten des neuen AWS einzuarbeiten.

6 Migration als Alltagstauglichkeitsbewahrer

In diesem Beitrag wurde ein Modell dargestellt, anhand dessen die Problematik eines Umstiegs auf ein neues AWS aus der Sicht des Anwenders dargestellt werden kann. Eine Beschäftigung mit den Elementen des Modells lässt Rückschlüsse zu auf die Wirkungen von Systemwechseln und auf mögliche Problemvermeidungs- beziehungsweise Problemreduzierungs-Optionen.

Alltagstauglichkeit stellt ein Gestaltungsparadigma dar, demzufolge die Anlehnung an andernorts bewährte Problemlösungsmechanismen, auch außerhalb des computerisierten Umfeldes, Wert dadurch schafft, dass der Benutzer Vertrauen in seine Problemlösungskompetenz gewinnt und so schneller und intuitiver seine individuelle Nutzungskompetenz erweitern kann. Der Alltagstauglichkeits-Ansatz wurde bestätigt und dahingehend präzisiert, dass diese Alltagstauglichkeit, dieses Anlehnen an alltägliche Gegebenheiten auch die Anlehnung an das gewohnte Systemumfeld eines umsteigenden Anwenders umfasst. Wiederkehrende Elemente der Mensch-Maschine-Schnittstelle, zum Beispiel die gleichen Buttons für die Basisfunktionen, die ähnlich aufgebauten Menus, etc. können dem Umsteiger dabei helfen, weniger Kompetenzverluste zu erfahren. Zugleich können bei veränderten Elementen der Mensch-Maschine-Schnittstelle gestalterische Rückgriffe auf andere, im Alltag möglichst vieler Anwender vertraute, Darstellungselemente helfen, um die Auswirkungen von systemwechselbedingten Kompetenzverlusten gering zu halten, indem dem Anwender ein möglichst schnelles Erlernen der neuen Elemente ermöglicht wird.

Des Weiteren wurde die Behandlung des Systemwechsels im Zuge von Migrationsstrategien dargestellt, sowie die Basisprämissen, auf denen solche Migrationsstrategien aufbauen. Einer dieser Ansatzpunkte ist der Investitionsschutz, auch im Sinne von Human Capital, also dem Erhalt von Know-how seitens der Anwender. Unter Verwendung von der in diesem Beitrag verwendeten Terminologie kann festgehalten werden, dass auch die individuellen Nutzungskompetenzen der Anwender bei einem AWS-Wechsel zu den erhaltenswerten Ressourcen gehören. Es kann somit betont werden, dass bei einem AWS-Wechsel nicht nur der Erhalt beziehungsweise Transfer von Inhalten, beispielsweise gespeicherten Daten, erforderlich ist, sondern eben auch der möglichst weitestgehende Erhalt von Elementen der Mensch-Maschine-Schnittstelle.

Ein möglicher Ansatz, der Entwicklern bei der Gestaltung systemwechselfreundlicher AWS hilfreich sein könnte, ist der der verschiedenen "Sichten", also die Möglichkeit, ein gegebenes AWS auf verschiedene Weisen hinsichtlich der Mensch-Maschine-Schnittstelle zu präsentieren. Hiermit könnte Umsteigern unter Umständen ermöglicht werden, erst einmal in einer vertrauten, nämlich an ihr bisheriges AWS angelehnten, Umgebung zu arbeiten, bevor dann unter Umständen der Wechsel auf ein neue, unter Umständen leistungsfähigere Bedienoberfläche von staten geht. Dann, wenn nur eine Sicht möglich ist, sollte eine Anlehnung an vertraute AWS erfolgen, je nach avisierte Kundenzielgruppe. Primär als Upgrades vertriebene AWS, also neue Versionen existierender AWS, sollten sich möglichst an den Vorgängerversionen orientieren. Neue AWS sollten sich demgemäß an weit verbreiteten Konkurrenzprodukten, Standardsoftware oder dem Betriebssystem orientieren. Das dargestellte Modell lässt sogar den Schluss zu, dass derartige Ähnlichkeiten und Anlehnungen unter Umständen tatsächlichen, faktischen Verbesserungen der Mensch-Maschine-Schnittstelle vorzuziehen sind, zumindest in Hinsicht auf Anwender, welche bereits Erfahrungen mit anderen, älteren AWS haben. Hier könnte sich dann auch ein möglicher Zielkonflikt ergeben, denn Neueinsteigern dürfte die gänzlich neue, aber einfachere Bedienung eher zum Vorteil gereichen.

Wo genau nun die Grenze zwischen dieser wünschenswerten Anlehnung und dem weniger wünschenswerten, vor allem aber juristisch heiklen, Plagiat liegt ist nicht Gegenstand dieses Beitrages, aber sicher eine interessante Fragestellung. Auch könnte untersucht werden, inwieweit die Anlehnung an existierende Produkte Differenzierungsstrategien zuwider läuft. Sicherlich existiert hier auch ein noch aufzulösender Widerspruch. Aus einer reinen Anwender-Perspektive jedenfalls hat eine gewisse Uniformität hier deutliche Vorteile [Po83].

Literaturverzeichnis

- [BMW07] Bohl, O.; Manouchehri, S.; Winand, U.: Migration als Ansatz zur Gestaltung mobiler Services. In: (Oberweis, A. et. al. Hrsg.): eOrganisation: Service-, Prozess-, Market-Engineering - 8. Internationale Tagung Wirtschaftsinformatik. Band 1, Universitätsverlag Karlsruhe, Karlsruhe, 2007; S. 309-326.
- [Dö97] Dörner, D.: Die Logik des Misslingens. Strategisches Denken in komplexen Situationen. Rowohlt Tb. Januar 1997.
- [Dö98] Dömer, F.: Migration von Informationssystemen – Erfolgsfaktoren für das Management. Deutscher Universitäts-Verlag, Wiesbaden 1998.
- [Du07] Duden - Deutsches Universalwörterbuch: 6., überarbeitete Auflage. Mannheim, Leipzig, Wien, Zürich: Dudenverlag 2007.
- [Gr89] Grudin, J.: The Case Against User Interface Consistency. In: Communications of the ACM, October 1989, Volume 32, Number 10. S. 1164-1173.
- [Ke00a] Keil-Slawik, R.: AUDIO VIDEO DISCO - Alltagstaugliche Konzepte für das Lernen mit neuen Medien. In: (Mehlhorn, K., Snelting, G. (Hrsg.): Informatik 2000. Neue Horizonte im neuen Jahrhundert. Berlin: Springer, 2000, S. 32 – 43.
- [Ke00b] Keil-Slawik, R.: Rahmenbedingungen für den Einsatz von Multimedia in der Hochschullehre. Unveröffentlichtes Manuskript zur Tagung der FB 8 der GI. Bederkesa. 2000. Verfügbar unter: <http://iug.uni-paderborn.de/rks//Publikationen.2007-09-28>.
- [Po83] Porter, M. E.: Wettbewerbsstrategie, Methoden zur Analyse von Branchen und Konkurrenten. 1. A. Campus, Frankfurt (Main), 1983.
- [Si92] Simon, A. R.: Systems Migration: A Complete Reference. Van Nostrand Reinhold, New York 1992.
- [St98] Staehle, W. H.: Management. Vahlen, München 1998.
- [Ta03] Tarasevich, P.: Designing Mobile Commerce Applications. In: Communications of the ACM, December 2003, Volume 46, Number 12. S. 57- 60.
- [Wi00] Winand, U.: Entwurfsprinzip Alltagstauglichkeit – Konzepte zur Gestaltung von medien-, kommunikations- und informationstechnischgestützten Anwendungssystemen und Diensten. Arbeitsbericht Nr. 27/Wirtschaftsinformatik. Universität Kassel. Kassel 2000.