Software-Ergonomie-Schulung in der betrieblichen Praxis - das SEE-Konzept

Marc Hassenzahl, Edmund Buchbinder, Jochen Prümper

Zusammenfassung

Die Qualifizierung von Software-Entwicklern und Entwicklerinnen im Bereich Software-Ergonomie ist ein Ansatz zum Transfer entsprechender Inhalte und Methoden in die betriebliche Praxis. Die hier vorgestellte „Software-Ergonomie-Schulung für Entwickler" (SEE) besitzt ein an die Bedürfnisse und Beschränkungen des betrieblichen Entwicklungsalltags angepaßtes Schulungskonzept. Es betont sowohl software-ergonomische Praxis als auch Theorie. Im Zentrum stehen die Entwickler, ihre Kompetenzen und ihre Bereitschaft sich mit software-ergonomischen Inhalten zu beschäftigen und sie anzuwenden. Es werden alltagstaugliche Methoden vermittelt und geübt. In jedem modernen Schulungskonzept muß ein entsprechendes Evaluationskonzept von Anfang an berücksichtigt werden. SEE betont die Evaluation auf mehreren Ebenen: auf der Ebene der unmittelbaren Reaktionen der Schulungsteilnehmer, ihrem objektiven Lernerfolg und dem nachfolgenden Transfer des Gelernten in die betriebliche Praxis. Nur so kann die Qualität der Schulung überprüft und weiter verbessert werden.

1 Einleitung

Die Gebrauchstauglichkeit von Software ist inzwischen ein allgemein akzeptiertes Qualitätsmerkmal. In der betrieblichen Praxis allerdings klafft meistens eine deutliche Lücke zwischen Anspruch und Umsetzung. Um gebrauchstaugliche Software in breitem Maße Wirklichkeit werden zu lassen, ist es notwendig, software-ergonomische Inhalte und Methoden in die Praxis der Software-Entwicklung zu transferieren (Vossen 1998).

Ein Ansatz ist die *Qualifizierung der Software-Entwickler*. Alternativ bzw. ergänzend sind der *Beratungsansatz*, bei dem ein Experte die software-ergonomische Qualitätssicherung übernimmt (Hassenzahl, Prümper & Buchbinder 1998) und/oder der *Prozeßansatz*, bei dem im Sinne einer „Organisationsentwicklung" ein adäquater (d.h. benutzer-orientierter) Software-Entwicklungsprozeß eingeführt wird.

Beim *Qualifizierungsansatz* muß langfristig bei der universitären (und beruflichen) Ausbildung der Software-Entwickler angesetzt werden, die trotz ausgearbeitetem Software-Ergonomie-Curriculum (vgl. Maaß et al. 1993) immer noch stiefmütterlich behandelt wird (Brennecke & Keil-Slawik 1997).

Daneben muß aber auch eine Qualifizierung der bereits im beruflichen Alltag stehenden Entwickler erfolgen. In diesem Sinne spielt die *organisationale/ betriebliche Weiterbildung* von Software-Entwicklern im Bereich Software-Ergonomie eine zentrale Rolle. Ziel des vorliegenden Beitrags ist es, einen Einblick in die Vorüberlegungen und Gestaltung einer solchen betrieblichen Weiterbildungsmaßnahme zu geben. Dazu sollen zunächst Schulungsziele, ein Schulungskonzept und Leitmotive entwickelt werden (Abschnitt 2). Danach wird ein darauf aufbauendes zweitägiges Schulungsprogramm vorgestellt (Abschnitt 3). Abschließend wird das Evaluationskonzept der Schulung erläutert (Abschnitt 4).

2 Schulungsziel, Konzept und Leitmotive

Grundlegendes *Ziel* einer organisationalen Weiterbildung zum Thema Software-Ergonomie ist es, *Software-Entwicklern software-ergonomische Inhalte und Methoden zu vermitteln, die in einer Organisation eingesetzt werden können und zur*

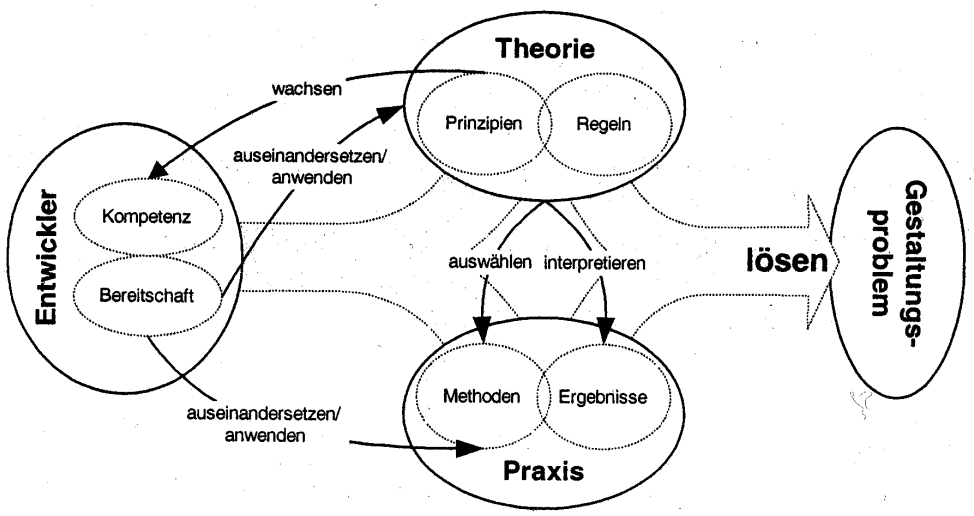

Abbildung 1: Schulungskonzept

Entwicklung gebrauchstauglicherer Software führen.

Neben dem Ziel benötigt eine Schulung ein *Konzept*, auf dessen Basis relevante Inhalte und eine angemessene Lehrform ausgewählt werden können.

Abbildung 1 stellt das Schulungskonzept mit seinen Elementen und den relevanten Beziehungen zwischen diesen Elementen dar.

Die grundlegende Aufgabe eines Software-Entwicklers ist es, *Gestaltungsprobleme* zu lösen. Dabei soll durch die Schulung gewährleistet werden, daß die erarbeitete Lösung software-ergonomischen Kriterien entspricht. Bei der Bearbeitung des Gestaltungsproblems kann auf zwei Informationsquellen zurückgegriffen werden: software-ergonomische *Praxis* und *Theorie*.

Das erfolgreiche Anwenden software-ergonomischer Methoden wird bei gegebener Einbettung in eine „ergonomiefreundliche" Organisationskultur in hohem Maße durch Merkmale des Entwicklers bestimmt. Dabei spielt zum einen seine software-ergonomische *Gestaltungskompetenz* eine entscheidende Rolle. Gestaltungskompetenz setzt sich sowohl aus praktischem als auch theoretischem Wissen zusammen.

Zum anderen ist die *Bereitschaft* der Entwickler zentral, sich mit Software-Ergonomie zu beschäftigen und sich deren grundlegenden Ziele zu eigen zu machen. Sie bestimmt während der Schulung, ob und wie der Entwickler sich mit den angebotenen Inhalten auseinandersetzt. In der Entwicklungspraxis bestimmt sie neben anderem, inwiefern vermitteltes Wissen zur Bearbeitung eines aktuellen Gestaltungsproblems herangezogen wird.

Unter software-ergonomischem *Praxiswissen* verstehen wir eine Sammlung von Methoden, wie z.B. das Erstellen eines Papierprototypen, die vom Entwickler in seinem Arbeitsalltag möglichst einfach und direkt zur Lösung des Gestaltungsproblems eingesetzt werden können.

Ebenso wichtig ist es, relevantes *Theoriewissen* zu vermitteln. Dies ist ein Punkt, der durchaus kontrovers diskutiert werden kann. Wir meinen, daß selbst bei der vorherrschenden starken Betonung von Praxisrelevanz auf theoretisches Wissen in Form von Prinzipien oder Regeln aus drei Gründen nicht verzichtet werden kann: Erstens ist theoretisches Wissen notwendig, um zu entscheiden, welche Methoden für eine Situation angemessen sind (und welche nicht). Zweitens kann man im Einzelfall nur unter Rückgriff auf theoretisches Wissen zu einer sinnvollen Interpretation der Ergebnisse angewendeter Methoden kommen. Drittens eröffnet theoretisches Wissen in Form von Prinzipien dem Entwickler die Chance, seine software-ergonomische Gestaltungskompetenz zu erweitern, d.h. zu „wachsen".

Auf der Basis dieses allgemeinen Schulungskonzepts müssen nun Inhalte ausgewählt werden, die die einzelnen Elemente zum Thema haben, d.h. den Entwickler selbst, software-ergonomische Theorie und Praxis. Da betriebliche Weiterbildung zum Thema Software-Ergonomie weitere spezifische Anforderungen stellt, haben wir jedes der zentralen Elemente unter ein *Leitmotiv* gestellt. Diese Leitmotive werden im folgenden kurz erläutert.

2.1 Bereitschaft: Positive Einstellung zur Software-Ergonomie erzeugen

Bevor Inhalte und Methoden effektiv vermittelt werden können, muß dazu eine Bereitschaft auf der Seiten des Lernenden bestehen. Eine solche Bereitschaft, ausgedrückt in einer *positiven* (nicht zu verwechseln mit einer unkritischen) *Einstellung* zur Software-Ergonomie ist Voraussetzung für die erforderliche Mitarbeit und einen nachfolgenden Transfer der Inhalte in die tägliche Praxis.

Gemäß unseren Erfahrungen entsteht ein Bedarf nach Software-Ergonomie-Qualifizierung - zumindest momentan - aufgrund der Anforderungen der „neuen" Bildschirmarbeitsverordnung (BildscharbV) und entsprechender Normen zur Gebrauchstauglichkeit von Software (DIN EN ISO 9241 Teile 10-17). Die Umsetzung dieser Anforderungen wird im Allgemeinen durch Personal- bzw. Betriebsräte eingefordert, was nicht immer konfliktfrei abläuft. Dementsprechend müssen „schwelende" Konflikte in der Organisation und Ressentiments gegenüber Software-Ergonomie bei der Konzeption einer Schulung antizipiert werden.

2.2 Praxis: An das Alltagswissen der Entwickler anknüpfen

Für Entwickler interaktiver Softwaresysteme ist Gebrauchstauglichkeit kein neues Thema. Manche verwenden bereits typische Methoden, wie z.B. die Analyse von Anforderungen in Zusammenarbeit mit dem Benutzer oder Prototyping. Dies geschieht allerdings meist in einer eher unsystematischen Form. Andere besitzen „naive" Theorien zur Software-Ergonomie, die aber durchaus fehlerhaft sein können. Ein Beispiel für eine solche „naive", aber fehlerhafte Theorie berichten Maaß, Rosson und Kellogg (1987) in einer Fußnote (S. 423): Der Entwickler eines Online-Tutorials gab an, bei Informationsdarstellung und Interaktionsarten gezielt auf Konsistenz zu *verzichten*, da dies den Benutzer langweile. Natürlich hat die Reduktion der Gebrauchstauglichkeit keinen motivierenden Effekt, vielmehr sollten Inhalte und didaktisches Konzept motivierend wirken.

Software-ergonomische Überlegungen sind also für Entwickler kein Neuland. Allerdings greifen sie dabei meist auf unsystematisches oder fehlerhaftes Wissen zurück. Eine Schulung muß dies berücksichtigen, korrigierend und systematisierend eingreifen. Dabei sollte das „Prozeßhafte" von Software-Entwicklung betont und wenn möglich Schwierigkeiten beim Anwenden der Methoden antizipiert werden.

2.3 Theorie: Gestaltungsprinzipien statt Gestaltungsregeln

Software-Entwicklung ist ein kreativer Prozeß. Während der Entwicklung müssen vom Entwickler hunderte einzelner Gestaltungsentscheidungen getroffen werden. Da die technisch-inhaltlichen Anforderungen und Rahmenbedingungen von System zu System verschieden sein können und sich über die Zeit verändern, erscheint es sinnvoll, abstrakte Gestaltungsprinzipien statt konkreter Gestaltungsregeln zu vermitteln. Gestaltungsprinzipien ermöglichen es dem Entwickler im Anwendungsfall neue und angemessene Regeln abzuleiten.

Auch ein inhaltliches Argument spricht eher dafür, auf - zugegebenermaßen vom Entwickler erwartete - konkrete Regeln weitestgehend zu verzichten. Gemäß DIN EN ISO 9241-11 (ISO 1996b) ist Gebrauchstauglichkeit nicht als Produktqualität, sondern als **Nutzungsqualität** („quality in use") zu verstehen. Gebrauchstauglichkeit entsteht erst im konkreten Zusammenspiel von Produkt und Nutzungskontext. Dies macht es besonders schwer, allgemeingültige und gleichzeitig konkrete Regeln zur Erstellung gebrauchstauglicher Software abzuleiten. Bevan (1997) beschreibt es so: „Although developers would like to know what attributes to incorporate in the code to reduce the effort required for use, presence or absence of predefined attributes cannot assure usability, as it is usually impossible to know how users will respond until they have actually experienced use of a prototype system".

Gebrauchstauglichkeit macht also einen theoretisch geleiteten, empirischen Ansatz erforderlich, der aufgrund stark variierender technischer bzw. inhaltlicher Anforderungen und Rahmenbedingungen nur schwer durch konkrete Gestaltungsregeln (z.B. in Form von „Checklisten") ersetzt werden kann. Selbst wenn man es versucht und die in DIN EN ISO 9241 aufgeführten, technologie- und kontextfrei formulierten Regeln zur Anwendung bringen will, steht man vor dem Problem, mehr als 400 Regeln kennen und wiederholt anwenden zu müssen (Görner, Burmester & Kaja 1997).

3 Schulungsprogramm

Auf der Basis der im vorherigen Abschnitt beschriebenen Überlegungen haben wir eine zweitägige „Software-Ergonomie-Schulung für Entwickler" (SEE) erarbeitet. In der betrieblichen Praxis muß man oft mit starken zeitlichen Beschränkungen rechnen. Um so wichtiger ist es, ein klares Konzept

	Modul	Inhalt
1	Einführung in die Software-Ergonomie *(Theorie)*	– Gestaltungsziel der Software-Ergonomie – Gebrauchstauglichkeit – Software und ihr Nutzungskontext – Nutzungsprobleme und Funktionsprobleme
2	Software-Ergonomie: Normen und Gesetze *(Theorie)*	– Gesetze und Richtlinien – DIN EN ISO 9241: „Ergonomische Anforderungen an die Büroarbeit mit Bildschirmgeräten" – DIN EN ISO 9241 Teil 11: Gebrauchstauglichkeit – DIN EN ISO 9241 Teil 10: Dialogprinzipien – Nutzen von Software-Ergonomie
3	Einführung in die benutzer-orientierte Software-Entwicklung *(Theorie)*	– ISO 13407 „Benutzer-orientierte Gestaltung interaktiver Systeme" – Benutzer-orientierte vs. „klassische" Software-Entwicklung
4	Beobachten und Analysieren: Anforderungsanalyse mit Benutzern *(Praxis)*	– Psychologisch-orientierte Beschreibung von Arbeitsaufgaben – Wichtige Merkmale der Arbeitsaufgabe – Analysen am Arbeitsplatz planen und durchführen
5	Gestalten und Visualisieren: Psychologische Grundlagen und Papierprototypen *(Praxis)*	– Wahrnehmung: Gestaltgesetze, Visuelle Suche, Farben – Gedächtnis: Erinnern versus Wiedererkennen – Lernen: Strategien – Papierprototypen
6	Bewerten und Verbessern: Walkthrough, Gebrauchstauglichkeitstest und Benutzerbefragung *(Praxis)*	– „Walkthrough": Bewerten mit Benutzern – Gebrauchstauglichkeitstest planen, durchführen und interpretieren – Benutzerbefragung mit dem ISONORM 9241/10 (Prümper 1997) planen, durchführen und interpretieren

Tabelle 1: Module und Inhalte (in Stichworten)

und Leitmotive zu entwickeln, da nur so die Auswahl wirklich relevanter Schulungsinhalte gewährleistet werden kann. Das Schulungsprogramm besteht aus insgesamt sechs Moduln. Der Titel und die Inhalte jedes Moduls in Stichworten findet sich in Tabelle 1.

Das Programm ist inhaltlich in zwei Abschnitte unterteilt: einen theoretischen und einen praktischen Teil. Der theoretische Teil setzt sich mit grundlegenden Fragen, Definitionen, Normen, Gesetzen und dem Nutzen von Software-Ergonomie auseinander. Mit Hilfe kleinerer Übungen (siehe unten) werden theoretische Überlegungen mit der alltäglichen Erfahrungswelt der Entwickler verknüpft. Den Übergang zum praktischen - in unserem Sinne - methodenorientierten Teil bildet eine Einführung in die benutzer-orientierte Software-Entwicklung anhand ISO 13407 (ISO 1996a). Die folgenden Methoden orientieren sich am chronologischen Prozeß der benutzer-orientierten Software-Entwicklung. Im Vordergrund stehen einfache Methoden (z.B. Papierprototypen), die ohne großen Aufwand verwirklicht werden können.

Übungen ermöglichen es, vermitteltes Wissen, angesprochene Probleme und Techniken zu erfahren und dadurch „begreifbar" zu machen. Sie senken ebenfalls die Hemmschwelle zur Anwendung im Berufsalltag. Im Rahmen des hier vorgestellten Schulungsprogramms haben Übungen einen hohen Stellenwert. In der Regel wird jedes der in Tabelle 1 aufgeführten Module mit einer Übung abgeschlossen. Dabei war es zentral, einen inhaltlichen Zusammenhang zwischen den Übungen herzustellen. Tabelle 2 beinhaltet ein Beispiel.

Die Übungen sollen - neben dem Anwenden erworbenen Wissens - auch die Bereitschaft zur Beschäftigung mit software-ergonomischen Themen erhöhen. Dies kann beispielsweise dadurch erreicht werden, daß sich Software-Entwickler in die Rolle des „naiven Benutzers" (siehe Tabelle 2)

Übung:	„Walkthrough": Bewerten eines selbst erstellten Papierprototypen *(Zeitpunkt: Mitte Modul 6)*
Ziel:	Schwierigkeiten bei der Anwendung des „Walkthroughs" (kooperatives Entdecken) erkennen. Auch erhöhte Kritikfähigkeit ist ein Ziel dieser Übung. Weiterhin soll durch die Übernahme der „Benutzerrolle" das Verständnis für Probleme mit einem unbekannten System erreicht werden.
Hintergrund:	Entwickler tendieren beim „Walkthrough" dazu, den Benutzer vom eigenen Entwurf überzeugen zu wollen, anstatt seine Probleme ernst zu nehmen.
Beschreibung:	Jeder Schulungsteilnehmer agiert einmal in der Rolle des „naiven Benutzers" und einmal in der Rolle des „Testleiters". In der Rolle des „naiven Benutzers" werden die Teilnehmer mit dem in einer vorherigen Übung erstellten Papierprototypen eines anderen Schulungsteilnehmers konfrontiert. In der Rolle des „Testleiters" sollen gemeinsam mit dem „naiven Benutzer" ergonomische Schwachstellen des eigenen Entwurfs identifizieren werden. Die Ergebnisse des „Walkthroughs" werden in eine Überarbeitung des ersten Papierprototypen integriert und im Plenum vorgestellt bzw. diskutiert. In einer „Benutzerrunde" berichten die Teilnehmer von eventuellen Schwierigkeiten, sich in die Rolle des Benutzers hineinzuversetzen und von Problemen beim Verständnis des fremden Entwurfs.

Tabelle 2: Beispiel für eine Übung

hineinversetzen müssen und anhand der dabei erlebten Schwierigkeiten feststellen, wie wenig sie eigentlich von und über ihre „Kunden" (die Benutzer) wissen.

4 Schulungsevaluation

Betriebliche Weiterbildung ist kein Selbstzweck. Sie ist nur dann sinnvoll, wenn sie auf ein Ziel ausgerichtet ist und dieses Ziel auch erreicht wird. Das Überprüfen der Zielerreichung kann unter dem Begriff *Schulungsevaluation* zusammengefaßt werden.

Evaluation ist aber auch ein Instrument zur kontinuierlichen Verbesserung des Schulungskonzepts. Dabei müssen die Organisation und Durchführung der Schulung, die Schulungsinhalte und die organisationale Einbindung der Schulungsmaßnahme berücksichtigt werden. Da Schulungskonzept und Evaluationskonzept eng verbunden sind, ist es notwendig, die Evaluation von Anfang an in das Schulungskonzept zu integrieren (Phillips 1997).

Wir leiten das Evaluationskonzept von dem anerkannten Vier-Ebenen-Modell von Kirkpatrick (1996) ab. Die vier Evaluationsebenen sind in Tabelle 3 kurz dargestellt.

Das Schulungskonzept sieht eine Evaluation auf den Ebenen „Reaktionen", „Lernen" und „Verhalten" vor. Eine Evaluation auf der Ebene „Resultate" erfordert einen tiefen Einblick in die Struktur der Organisation. Sie ist daher nur von der Organisation selbst zu leisten und hier nicht explizit berücksichtigt.

Reaktionen	*subjektive Beurteilung der Schulung durch die Teilnehmer*
	Hat die Schulung den Teilnehmern gefallen? Hat sie ihnen Spaß gemacht? Wurde sie für sinnvoll erachtet? Hatten die Teilnehmer das Gefühl etwas gelernt zu haben?
Lernen	*objektive Beurteilung des Lernerfolgs*
	Wurden die vermittelten Prinzipien, Fakten und Techniken erlernt?
Verhalten	*Beurteilung des Lerntransfers*
	Wird das Gelernte im Arbeitsalltag eingesetzt?
Resultate	*Beurteilung des Nutzens durch die Organisation*
	Hatte die Schulung aus organisationaler Sicht den gewünschten Erfolg (z.B. Verbesserung der software-ergonomischen Qualität der entwickelten Software)?

Tabelle 3: Das Vier-Ebenen-Modell nach Kirkpatrick (Kirkpatrick 1996)

	Stimme nicht zu				Stimme zu
Ich war insgesamt zufrieden mit dem Training.	☐	☐	☐	☐	☐
Das Training hat Spaß gemacht.	☐	☐	☐	☐	☐

Abbildung 2: Beispiele aus dem Reaktionsfragebogen (Dimension „Affektive Reaktionen")

4.1 Reaktionen

Die Erhebung der Reaktionen erfolgt direkt im Anschluß an die Schulung mit einem speziell entwickelten Fragebogen. Dieser *Reaktionsfragebogen* wurde aus der Trainings- und Transferforschung (Alliger et al. 1997; Baldwin & Ford 1988; Kirkpatrick 1996; Tannenbaum et al. 1991; Warr & Bunce 1995) abgeleitet und beinhaltet die folgenden Dimensionen: Schulungsgestaltung (z.B. Lerntempo), Schulungsleitung, Einschätzung des subjektiven Lernerfolgs, Motivation (z.B. Motivation zur Umsetzung des Gelernten), subjektiv beurteilte Rahmenbedingungen für Transfer (z.B. Unterstützung durch Vorgesetzte), Affektive Reaktionen (z.B. Zufriedenheit, Spaß), Anforderungsniveau der Schulung und Erfüllung der persönlichen Erwartungen an die Schulung.

Abbildung 2 zeigt einen kurzen Ausschnitt aus dem Reaktionsfragebogen.

4.2 Lernen

Wie in Abschnitt 2 dargestellt, ist es das Ziel der Schulung, sowohl *Gestaltungskompetenz* (in Form von theoretischem und praktischem Wissen) als auch *Bereitschaft* (in Form einer positiven Einstellung zur Software-Ergonomie) zu vermitteln.

Anders als bei den Reaktionen ist hier die *Veränderung* von Wissen und Einstellung relevant. Entsprechend muß beides sowohl vor als auch nach der Schulung mittels auf die Schulungsinhalte abgestimmter Fragebögen erhoben werden. Abbildung 3 zeigt jeweils ein Beispiel für zu beurteilende Wissensaussagen (a) und Einstellungsaussagen (b).

a) Wissensaussage

Je mehr Funktionen eine Software bietet, desto aufgabenangemessener ist sie.

Diese Aussage ist ...		Ich bin mir bei der Antwort...				
...richtig	...falsch	...unsicher				...sicher
☐	☐	☐	☐	☐	☐	☐

b) Einstellungsaussage

	Stimme nicht zu				Stimme zu
Software-Ergonomie spart Geld.	☐	☐	☐	☐	☐
Zuerst muß die Software laufen, dann ist Zeit für Software-Ergonomie.	☐	☐	☐	☐	☐

Abbildung 3: Beispiele aus dem Lernfragebogen

4.3 Verhalten

Die Beurteilung des Transfers erfolgt einige Zeit (mindestens 3 Monate) nach der Schulung mit Hilfe eines *Transferfragebogens*. Dieser beinhaltet die folgenden Aspekte „Anwenden der Schulungsinhalte im Entwicklungsalltag" sowie „positive und negative Erfahrungen bei der Anwendung".

Dieser Fragebogen soll von den Schulungsteilnehmern ausgefüllt werden. Da eine subjektive Einschätzung der eigenen Tätigkeit immer mit Verzerrungen behaftet ist, müssen sowohl die Vorgesetzten als auch die „Kunden" (d.h. die Benutzer) in die Beurteilung des Transfers einbezogen werden.

5 Schlußfolgerung

Die vorliegende Arbeit gibt Einblick in die Vorüberlegungen und Inhalte einer betrieblichen Weiterbildungsmaßnahme zum Thema Software-Ergonomie. Die Wichtigkeit der engen Verschränkung praktischen und theoretischen Wissens, die angemessene Berücksichtigung der Bereitschaft auf Seiten der Entwickler und die starke Integration von Schulungs- und Evaluationskonzept wurden betont. Auf dieser Basis wurden Schulungsmaterialien, Foliensätze und Evaluationsmaterialien für eine zweitägige Weiterbildungsmaßnahme erstellt. Erste Erfahrungen im Praxiseinsatz erscheinen erfolgsversprechend (Buchbinder 1998). Detaillierte Analysen und weitere Praxiseinsätze sollen diesen ersten positiven Eindruck festigen.

6 Literatur

Alliger, G.M., Tannenbaum, S.I., Bennett, W., Traver, H. & Shotland, A. (1997). A meta-analysis of the relations among training criteria. *Personnel Psychology*, **50**, S. 341-358.

Baldwin, T.T. & Ford, J.K. (1988). Transfer of training: A review and directions for future research. *Personnel Psychology*, **41**, S. 63-105.

Bevan, N. (1997). Quality and usability: A new framework. In E. van Veenendaal & J. McMullan (Hrsg.), *Achieving software product quality*. Niederlande: Tutein Nolthenius.

Brennecke, A. & Keil-Slawik, R. (1997). Einsatz elektronischer Lehr- und Lernumgebungen in der Software-Ergonomie-Ausbildung. In R. Liskowsky, B. M. Velichkovsky & W. Wünschmann (Hrsg.), *Software-Ergonomie '97. Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung*, S. 83-92. Stuttgart: B.G. Teubner.

Buchbinder, E. (1998). Evaluation von Software-Trainings in der betrieblichen Praxis. Darmstadt: Technische Universität, unveröffentlichte Diplomarbeit.

Görner, C., Burmester, M. & Kaja, M. (1997). Dialogbausteine: Ein Konzept zur Verbesserung der Konformität von Benutzungsschnittstellen mit internationalen Standards. In R. Liskowsky, B. M. Velichkovsky & W. Wünschmann (Hrsg.), *Software-Ergonomie '97. Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung*, S. 157-165. Stuttgart: B.G. Teubner.

Hassenzahl, M., Prümper, J. & Buchbinder, E. (1998). Software ergonomics in practice: the importance of acceptance-related issues. In M. Sikorski & M. Rauterberg (Hrsg.), *Transferring usability engineering to industry. International workshop proceedings*, S. 9-13. Gdansk: Technical University of Gdansk.

ISO (1996a). ISO CD 13407: Human-centred design, commitee draft. *International Organization for Standardization.*

ISO (1996b). ISO 9241: Ergonomic requirements for office work with visual display terminals. Part 11: Guidance on usability. *International Organization for Standardization.*

Kirkpatrick, D.L. (1996). Evaluating Training Programs: The Four Levels. San Francisco: Berret-Koehler Publishers.

Maaß, S., Ackermann, D., Dzida, W., Gorny, P., Oberquelle, H., Rödiger, K.H., Rupietta, W. & Streitz, N. (1993). Software-Ergonomie-Ausbildung in Informatik-Studiengängen bundesdeutscher Universitäten. *Informatik-Spektrum*, **16**, S. 25-30.

Maaß, S., Rosson, M.B. & Kellog, W.A. (1987). Benutzerfreundlichkeit, Systemkonsistenz und andere schwer definierbare Prinzipien: Interviews mit Systementwicklern. In W. Schönpflug & M. Wittstock (Hrsg.), *Software-Ergonomie '87: Nützen Informationssysteme dem Benutzer?*, S. 417-427. Stuttgart: B.G. Teubner.

Phillips, J.J. (1997). Handbook of Training Evaluation and Measurement Methods. Houston: Gulf Publishing Company.

Prümper, J. (1997). Der Benutzungsfragebogen ISONORM 9241/10: Ergebnisse zur Reliabilität und Validität. In R. Liskowsky, B. M. Velichkovsky & W. Wünschmann (Hrsg.), *Software-Ergonomie '97. Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung*, S. 253-262. Stuttgart: B.G. Teubner.

Tannenbaum, S.I., Mathieu, J.E., Salas, E. & Cannon-Bowers, J.A. (1991). Meeting trainees' expectations: The influence of training fulfillment on the development of commitment, self-efficacy, and motivation. *Journal of Applied Psychology*, **76[6]**, S. 759-769.

Vossen, P.H. (1998). Usability contracts and other strategies of usability transfer in industrial settings. In M. Sikorski & M. Rauterberg (Hrsg.), *Transferring usability engineering to industry. International workshop proceedings*, S. 28-32. Gdansk: Technical University of Gdansk.

Warr, P. & Bunce, D. (1995). Trainee characteristics and the outcomes of open learning. *Personnel Psychology*, **48**, S. 347-375.