Usability-Engineering für Sicherheitskritische Mensch-Maschine-Systeme

Prof. Dr. Michael Herczeg, Institut für Multimediale und Interaktive Systeme, Universität zu Lübeck

Kurzfassung

Usability-Engineering zielt auf die systematische Entwicklung anwendungs- und benutzergerechter interaktiver Computeranwendungen. Für den Bereich der sicherheitskritischen Mensch-Maschine-Systeme sind Besonderheiten und Ergänzungen für das Usability-Engineering vorzusehen. Diese werden in diesem Beitrag diskutiert.

1 Einleitung

Zunächst sollen die Methodik und die generelle Bedeutung von Usability-Engineering und sicherheitskritischen Systemen dargestellt werden.

1.1 Usability-Engineering

Usability-Engineering, auch *benutzerzentrierte Systementwicklung* genannt, verbindet die Methoden des *Software-Engineering* mit denen des *Cognitive-Engineering* [9, 10] und der *Software-Ergonomie* [5]. Zunächst soll eine systematische und ökonomische Vorgehensweise bei der Entwicklung von Software-Systemen durch phasenorientierte Entwicklungsprozesse gewährleistet werden. Der Entwicklungsprozess soll dabei aber nicht an den möglichen Funktionen, sondern an den durchzuführenden Aufgaben, dem Anwendungskontext sowie den Fähigkeiten und Erwartungen der Benutzer ausgerichtet werden.

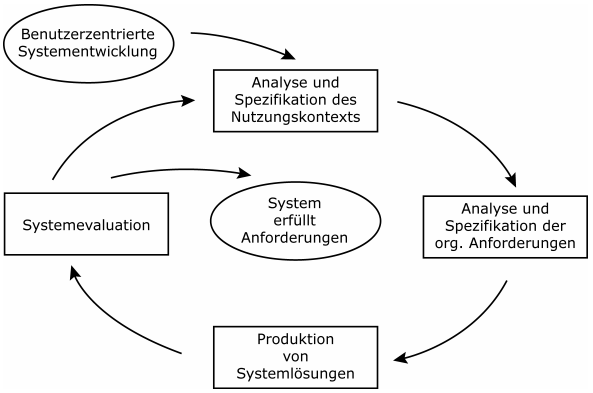

Bild 1 Usability-Engineering nach ISO 13407

Beim Usability-Engineering ist man bemüht, die realen Kontexte und Bedürfnisse der Benutzer zu erfassen und die Systemlösungen daraufhin zu konzipieren und zu bewerten (siehe Beispiel aus der Prozessnorm ISO 13407 in **Bild 1**). Im Ergebnis sollen die Systeme *gebrauchstauglich*, d.h. *effektiv*, *effizient* und *zufriedenstellend* sein. Diese und weitere verfeinerte Kriterien finden sich in der Produktnorm ISO 9241.

1.2 Sicherheitskritische Systeme

Der Einsatz sicherheitskritischer Systeme hat direkten Einfluss auf die Sicherheit und Unversehrtheit von Mensch und Umwelt. Beispiele solcher Systeme sind Steuerungs- und Kontrollsysteme für Fahrzeuge, Produktionsanlagen, Kraftwerke und medizintechnische Systeme. Mit diesen Systemen werden technische oder biomedizinische Prozesse gesteuert. Damit ist ein beträchtliches Risikopotenzial verbunden. Sicherheitskritische Mensch-Maschine-Systeme müssen daher mit besonderer Sorgfalt entwickelt und mit Schutzmechanismen und Hinweisen gegen menschliche oder technische Fehlfunktionen ausgestattet werden.

2 Entwicklung sicherheitskritischer Mensch-Maschine-Systeme

Im Rahmen des Usability-Engineering werden zunächst allgemeine Kriterien verwendet, um die Gebrauchstauglichkeit software-basierter Systeme zu gewährleisten. Bei sicherheitskritischen Mensch-Maschine-Systemen müssen die Prozesselemente des Usability-Engineering erweitert und zusätzliche Kriterien einbezogen werden. Die basalen Prozessphasen des Software- bzw. dem Usability-Engineering, nämlich *Analyse*, *Design (Entwurf)*, *Implementierung* und *Validierung (Evaluation)* bilden auch die Grundlage bei der Entwicklung sicherheitskritischer Mensch-Maschine-Systeme. Allerdings müssen die einzelnen Phasen spezifisch ausgeprägt werden [4].

2.1 Analyse

Die Analysephase im Usability-Engineering sieht vor, *Kontext-*, *Organisations-*, *Aufgaben-* sowie *Benutzeranalysen* vorzunehmen. Bei sicherheitskritischen Systemen müssen diese durch eine Analyse möglicher *Fehler und deren Folgen* (z.B. mittels FTA, FMEA,

HAZOP) ergänzt werden. Dabei sind *Risikoanalysen* vorzunehmen, die aufzeigen, mit welcher Wahrscheinlichkeit und Tragweite *Gefährdungen* zu erwarten sind und wie sie beherrscht werden können. Für medizintechnische Systeme findet sich Näheres zum *Risikomanagement* in der Norm ISO 14971. Bei *Zuverlässigkeitsanalysen* zeigt sich, dass sich technische und menschliche Zuverlässigkeit in der Fehlerrate gravierend unterscheiden [3]. Menschen begehen sehr leicht Fehler unterschiedlichster Art [11]. Technische Systeme auf der anderen Seite können zwar mit niedrigen Fehlerraten arbeiten, besitzen allerdings nur sehr begrenzte ganzheitliche Strategien. In der Analysephase müssen die spezifischen Fähigkeiten und Grenzen von Mensch und Technik im Anwendungskontext für die Konzeption der *Arbeitsteilung* erarbeitet werden.

2.2 Design (Entwurf)

In der Designphase müssen die Erkenntnisse aus der Analysephase in eine arbeitsteilige Funktionsstruktur für Mensch und Maschine übersetzt werden [6]. Dabei sind in sicherheitskritischen Handlungsabschnitten komplementäre Redundanzen zwischen Mensch und Maschine vorzusehen. Wichtige Prinzipien sind das Vorhersehen und Zulassen von Fehlhandlungen, ohne die Sicherheit des Gesamtsystems zu gefährden [7, 8]. Dieses Prinzip, auch *Design for Error* genannt, geht von realistischen Fähigkeiten und Grenzen von Mensch und Maschine aus und bringt beide bestmöglich zur Wirkung. Neben diesem Prinzip ist es für den menschlichen Operateur wichtig, ein System nutzen zu können, das seine *Situation Awareness* gewährleistet, d.h. die drei kognitiven Teilprozesse *Wahrnehmen*, *Verstehen* und *in die Zukunft Projizieren* unterstützt.

2.3 Implementierung

Obwohl auch die Implementierung sicherheitskritischer Systeme Besonderheiten aufweist, soll an dieser Stelle nur auf die besondere Herausforderung hinsichtlich der *Korrektheit*, *Echtzeitfähigkeit*, *Stabilität* und *Robustheit* der Systeme hingewiesen werden, für die es bereits diverse Methoden im Rahmen des Software-Engineering gibt. Besonders die *Verifikation*, der umfassende Test der einzelnen Komponenten, ist für die spätere Zuverlässigkeit der Systeme von Bedeutung.

2.4 Validierung (Evaluation)

Im Rahmen der Validierung sind für sicherheitskritische Mensch-Maschine-Systeme zunächst die Evaluationsmethoden des Usability-Engineering vorzusehen. Die *theoriebasierte Evaluation* stützt sich auf allgemeine Kriterien für gebrauchstaugliche Systeme. Die *aufgabenbasierte Evaluation* nutzt die Aufgabenanalysen und zeigt, wie mit dem System reale Problemstellungen bewältigt werden können. Die *benutzerbasierte Evaluation* muss nachweisen, dass reale Benutzer mit dem System erfolgreich und sicher arbeiten können. Die Evaluationen sind sowohl *summativ*, d.h. am Ende der Entwicklung, als auch *formativ*, d.h. während der beschriebenen anderen Entwicklungsphasen vorzunehmen.

Sicherheitskritische Systeme sollten mit Hilfe der beschriebenen Methoden eher *evolutionär und konservativ* als *revolutionär und innovativ* entwickelt und optimiert werden. Kurzfristige und marktorientierte Optimierungen stellen sich schnell als schädlich heraus.

3 Literatur

[1] Endsley, M.R. & Garland, D.J. (Eds.): *Situation Awareness – Analysis and Measurement*. Mahwah: Lawrence Erlbaum Associates, 2000.

[2] Herczeg, M.: *Sicherheitskritische Mensch-Maschine-Systeme*. FOCUS MUL, 17(1), 2000, 6-12.

[3] Herczeg, M.. *Interaktions- und Kommunikationsversagen in Mensch-Maschine-Systemen als Analyse- und Modellierungskonzept zur Verbesserung sicherheitskritischer Technologien*. In Grandt, M. (Hrsg.), Verlässlichkeit der Mensch-Maschine-Interaktion, DGLR-Bericht 2004-03. Bonn: Deutsche Gesellschaft für Luft- und Raumfahrt, 2004, 73-86.

[4] Herczeg, M.: *A Task Analysis and Design Framework for Management Systems and Decision Support Systems*. ACIS International Journal of Computer & Information Science, 2(3), September, 2001, 127-138.

[5] Herczeg, M.: *Software-Ergonomie*. München: Oldenbourg Wissenschaftsverlag, 2005.

[6] Herczeg, M.: *Interaktionsdesign*. München: Oldenbourg Wissenschaftsverlag, 2006.

[7] Hollnagel, E., Woods, D.D. & Leveson, N. (Eds.): *Resilience Engineering*. Hampshire: Ashgate, 2006.

[8] Hollnagel, E.: *Barriers and Accident Prevention*. Hampshire: Ashgate, 2004.

[9] Norman, D.A. & Draper, S.W. (Eds.): *User Centered System Design*. Hillsdale: Lawrence Erlbaum Associates, 1986.

[10] Rasmussen, J., Pejtersen, A.M. & Goodstein, L.P.: *Cognitive Systems Engineering*. New York: Wiley & Sons, 1994.

[11] Reason, J.: *Human Error*. Cambridge: Cambridge University Press, 1990.