

# Usability greifbar machen: Interaktionsanforderungen

Holger Röder

Universität Stuttgart, Institut für Softwaretechnologie, Abteilung Software Engineering

roeder@informatik.uni-stuttgart.de

## Motivation und Zielgruppe

Gute Usability ist ein allgemein akzeptiertes Ziel bei der Entwicklung von Software. Betrachtet man Prozessmodelle für die Software-Entwicklung und die Situation in der Praxis, fällt jedoch auf, dass dieses Ziel nur selten systematisch verfolgt wird [1]. Stattdessen wird darauf vertraut, dass die Beteiligten implizit zu einer ergonomischen Lösung kommen – ein Vorgehen mit ungewissem Ausgang, das der Bedeutung von Usability nicht gerecht wird.

Ansätze zur Verbesserung dieser Situation können nur erfolgreich sein, wenn sie die typischen Bedingungen heutiger Software-Entwicklung berücksichtigen. Dies heißt insbesondere, dass keine eigenständige Usability-Organisation im Unternehmen existiert, keine (oder nur wenige) Experten verfügbar sind und der Fokus bei der Entwicklung auf der Funktionalität der Software liegt. Der hier vorgestellte Ansatz ist als Beitrag zur wissenschaftlichen Diskussion über die Frage gedacht, welche Rolle Usability-Aspekte, speziell Vorgaben für die Interaktionsgestaltung, unter den genannten Bedingungen im Rahmen der Anforderungsanalyse spielen können. Gleichzeitig liegt dem Ansatz auch der Anspruch praktischer Anwendbarkeit zugrunde; er richtet sich also ausdrücklich auch an Praktiker, die mit der Entwicklung interaktiver Softwaresysteme betraut sind. Dies zeigt sich auch in dem Versuch, mit Use Cases eine in der Praxis verbreitete Spezifikationstechnik als Ausgangspunkt für die Berücksichtigung von Usability-Aspekten heranzuziehen, um auch Entwicklern ohne Usability-Expertise die ergonomische Interaktionsgestaltung zu erleichtern.

## Usability-Anforderungen

Im Requirements Engineering werden Anforderungen an die Usability traditionell den nichtfunktionalen Anforderungen zugerechnet. Lehrbücher und Normen propagieren die Spezifikation quantifizierter Anforderungen auf Grundlage eines Qualitätsmodells von Usability, das messbare Teilqualitäten wie Effizienz oder Benutzerzufriedenheit vereint. In der Praxis steht diesem Vorgehen nicht zuletzt das Problem der Operationalisierung entsprechender Anforderungen entgegen: die Forderung, ein Benutzer müsse eine bestimmte Aufgabe in einer definier-

ten Zeit erledigen können, enthält für Entwickler kaum Hinweise auf eine mögliche Umsetzung.

Rückt man von der Maximalforderung quantifizierter Usability-Anforderungen ab, können auch solche Anforderungen betrachtet werden, denen der Versuch einer Operationalisierung von Usability zugrunde liegt. Beispiele für diese Art von Anforderungen sind etwa Styleguides oder UI-Prototypen, die sich auf die *physische* Gestaltung der Benutzungsschnittstelle (z. B. Farben, Schriftarten oder die Anordnung von Dialogelementen), beziehen und deren Einhaltung zu ergonomisch gestalteten Dialogen und – in der erhofften Konsequenz – zu hoher Usability führt. Auch für die *logische* Interaktionsgestaltung, also z. B. die Konzeption der Benutzerführung, die Festlegung zu verwendender Metaphern oder die Auswahl darzustellender Informationen, lassen sich Vorgaben formulieren. Existierende Methoden sehen eine systematische Entwicklung derartiger *Interaktionsanforderungen* jedoch nicht vor. Der im folgenden Abschnitt vorgestellte Ansatz versucht, eine solche Brücke zwischen Requirements Engineering und Usability Engineering zu schlagen. Er ist dabei als Ergänzung, nicht als Alternative zu existierenden Techniken wie z. B. UI-Prototypen zu sehen.

## Entwicklung von Interaktionsanforderungen

Prinzipien guter Interaktionsgestaltung, aus denen sich Vorgaben für die Entwicklung ableiten lassen, sind vielfach dokumentiert, etwa in Form von *Interaktionsmustern* [2]. Allerdings finden sich in der Literatur kaum Hinweise, wie Entwickler die für eine Problemstellung geeigneten Interaktionsmuster erkennen und auswählen können, und wie die Berücksichtigung von Interaktionsmustern spezifiziert werden kann.

Ein möglicher Ansatz, der diese Lücke schließt, besteht in der Ergänzung einer funktionalen Spezifikation durch systematisch entwickelte Interaktionsanforderungen, die Interaktionsmuster und Spezifikationsbestandteile verknüpfen. Grundlage ist hierbei ein Katalog von Interaktionsmustern. Jedes Interaktionsmuster enthält eine formalisierte Beschreibung des Problems, auf das das Muster anwendbar ist. Als Problem werden dabei *Interaktionselemente*, d. h. gestaltbare Bestandteile der Inter-

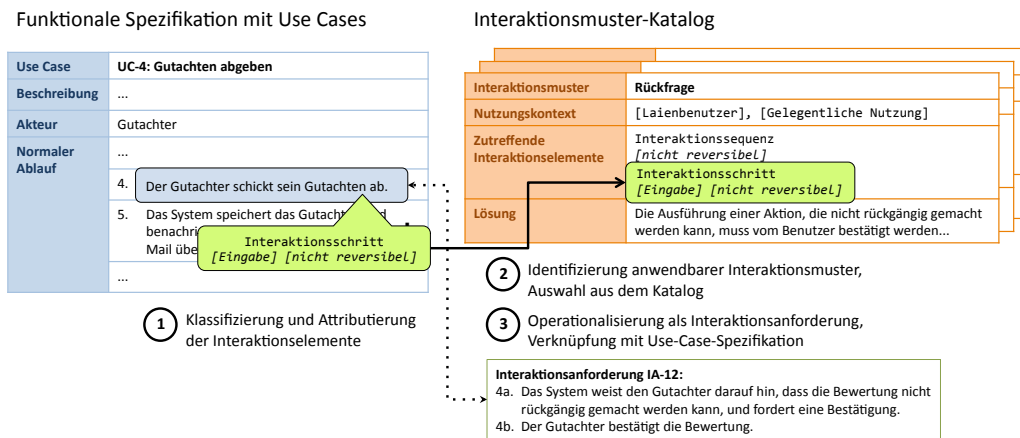


Abbildung 1: Beispielhafte Entwicklung einer Interaktionsanforderung

aktion (z. B. einzelne *Interaktionsschritte*), mit bestimmten *Eigenschaften* verstanden. Beispielsweise kann ein Interaktionsmuster eine Lösung für die ergonomische Gestaltung einer Abfolge von Interaktionsschritten bieten, die der Benutzer unterbrechen können soll (d. h. für eine Interaktionssequenz mit der Eigenschaft *UNTERBRECHBAR*). Auch ein bestimmter *Nutzungskontext* des Systems kann Teil der Problemstellung sein. Die Problemstellung bildet zusammen mit der Beschreibung einer passenden ergonomischen Gestaltungslösung den Kern jedes Interaktionsmusters.

Die eigentliche Entwicklung der Interaktionsanforderungen geht von der Spezifikation der Systemfunktionalität aus. Hier wird vorausgesetzt, dass eine funktionale Spezifikation bereits existiert und den Charakter einer essentiellen Interaktionsbeschreibung hat; dies ist etwa bei Use Cases, die Interaktionsabläufe zwischen Akteuren und System spezifizieren, der Fall. Die einzelnen Bestandteile der Interaktionsbeschreibung können in diesem Fall als Interaktionselemente klassifiziert und mit den entsprechenden Eigenschaften attribuiert werden. Für die so definierten Problemstellungen können dann passende Interaktionsmuster aus dem Katalog ausgewählt und deren Anwendung anschließend als Interaktionsanforderungen spezifiziert werden.

Beispielhaft sei hier das Vorgehen bei der Entwicklung einer Interaktionsanforderung für ein Konferenzverwaltungssystem beschrieben (Abb. 1).

1. Klassifizierung und Attributierung der Interaktionselemente: Im Beispiel entspricht Schritt 4 im Use Case GUTACHTEN ABGEBEN einem Interaktionsschritt mit den Eigenschaften *EINGABE* und *NICHT REVERSIBEL*, da es sich um eine Benutzereingabe handelt, die der Benutzer nicht rückgängig machen kann.
2. Auswahl passender Interaktionsmuster für die im ersten Schritt identifizierten Problemstellungen: Im Beispiel enthält der Katalog das In-

teraktionsmuster *RÜCKFRAGE*, das u. a. auf *NICHT REVERSIBEL EINGABEN* zutrifft.

3. Spezifikation von Interaktionsanforderungen durch Operationalisierung der anzuwendenden Interaktionsmuster: Die spezifische Anwendung des Musters *RÜCKFRAGE* für Schritt 4 des Use Case wird als Interaktionsanforderung IA-12 spezifiziert, die in diesem Fall die Form zusätzlicher Use-Case-Schritte hat; die Anforderung verknüpft Use Case und Interaktionsmuster.

### Fazit

In dieser Arbeit wird ein Ansatz zur Interaktionsgestaltung skizziert; neu ist hierbei die Verknüpfung einzelner Bestandteile der funktionalen Spezifikation (z. B. einzelner Schritte in einem Use Case) mit Vorgaben zur Interaktionsgestaltung, die auf Grundlage eines Katalogs von Interaktionsmustern entwickelt und als Interaktionsanforderungen spezifiziert werden. Gute Gründe sprechen für die systematische Entwicklung von Interaktionsanforderungen. Entwickler werden bei der Auswahl relevanter Prinzipien der Interaktionsgestaltung unterstützt. Die Operationalisierung in Form greifbarer Anforderungen ermöglicht die Berücksichtigung beim Architekturentwurf und bei der Implementierung. Die Behandlung als Anforderungen erlaubt zudem die Analyse und Priorisierung der Interaktionsanforderungen sowie eine spätere Prüfung der Umsetzung, und bindet die Interaktionsgestaltung so stärker in traditionelle Entwicklungsprozesse ein.

### Referenzen

- [1] M. Breitfeld: *Usability-Anforderungen in der Praxis*. Diplomarbeit Nr. 2869, Institut für Softwaretechnologie, Universität Stuttgart, 2009.
- [2] A. Dearden, J. Finlay: *Pattern languages in HCI: A Critical Review*. Human-Computer Interaction, 21(1), 2006.