

Wie sag ich's meinem Anwender?

Vorgehen und Empfehlungen für die Gestaltung von Meldungen

Richard Oed
Daimler AG
GR/ESP, Prozessgestaltung
Postfach 2360
89013 Ulm
richard.oed@daimler.com

Heiko Ziegler
Daimler TSS GmbH
Bereich Consulting
Wilhelm-Runge-Str. 11
89081 Ulm
heiko.ziegler@daimler.com

Abstract

Das richtige Feedback an den Anwender kann die Usability eines Systems deutlich beeinflussen. Leider wird diesem Aspekt bei der Softwareentwicklung häufig zu wenig Beachtung geschenkt. Welche Erfahrungen wir dabei in einem aktuellen Projekt gemacht haben und welche Lösungsmöglichkeiten erarbeitet

wurden, zeigt dieser Erfahrungsbericht auf. Erarbeitet wurden Guidelines zur Erstellung von Meldungen sowie ein Gestaltungskonzept zur textuellen und grafischen Gestaltung aller Hinweis-, Warn- und Fehlermeldungen im zu entwickelnden System.

Keywords

Feedback, Fehlermeldungen, Guidelines, Styleguide, Usability Engineering

1.0 Motivation

Wer soll das verstehen? Das war die Frage eines Anwenders an das Projektteam bezogen auf die folgende Fehlermeldung:



Die ausgeführte Aktion führte zu folgendem Ergebnis:

Die an der Quell-Verwendungsversion hängende Quell-Stammversion ist keine Version des der Ziel-Verwendung zugeordneten (Ziel-)Stamms (BATTERIE SENSOR). Kopieren daher nicht möglich. Bitte selektieren Sie eine Versionsversion, die eine korrekte Stammversion referenziert.

Wie konnte das passieren?

Fachexperten, Anwendungsprogrammierer und andere Personen ohne ausreichende Usability-Erfahrung hatten nach bestem Wissen und Gewissen Meldungen und Fehlermeldungen für eine umfangreiche und komplexe Anwendung geschrieben. Aus ihrer Sicht fachlich korrekt, völlig klar und verständlich und alle Eventualitäten bedenkend.

Die Anwender verstanden allerdings nur „Bahnhof“.

Da erst wurde dem Projektteam bewusst, dass dem Aspekt der Anwendungsmeldungen zu wenig Beachtung geschenkt wurde, die Ersteller der Meldungstexte völlig allein gelassen wurden und im Projekt keinerlei Vorgaben für die Erstellung von Meldungstexten existierten, bzw. nicht bekannt waren.

So erhielten die Autoren den Auftrag, sich um dieses Thema zu kümmern und einen Vorschlag für die Erstellung und Gestaltung von Fehlermeldungen auszuarbeiten.

2.0 Erfahrungen

Die Erfahrung aus verschiedenen Projekten zeigt, dass bei der Erstellung von Anwendungen das Feedback in Form von Meldungen ein häufig vernachlässigtes Element darstellt. Hierzu gehören insbesondere Hinweis-, Warn- und Fehlermeldungen die an den Anwender gerichtet sind.

Diese Meldungen sind oft

- in der Wortwahl inkonsistent,
- in der Gestaltung uneinheitlich.
- in schlechtem Deutsch geschrieben und deshalb
- nicht oder kaum verständlich.

In den Entwicklungsprojekten wird oft der Fehler gemacht, dass die Erstellung der Meldungen erst

- in letzter Minute angegangen,
- dem Programmier allein überlassen und
- der Inhalt praktisch keinem Review unterzogen wird.

Das Outsourcing der Softwareentwicklung bringt eine weitere Herausforderung mit sich. So werden Meldungstexte oft während der Programmentwicklung von Nicht-Muttersprachlern in Englisch erstellt und anschließend ins Deutsche rückübersetzt. Auch dabei kann es zu erheblichen Verständnisproblemen kommen.

3.0 Lessons Learned

Dieses wiederkehrende Problem und unser Lösungsbeitrag in einem aktuellen Projekt, werden in diesem Erfahrungsbericht adressiert.

Unsere grundsätzlichen Lessons Learned sind:

- Es ist wichtig, in einem Projekt frühzeitig das Vorgehen und die Regeln für das Verfassen von Meldungen zu definieren und zu kommunizieren.
- Von Anfang an ein Glossar aufzusetzen und durchgängig zu verwenden vermeidet unterschiedliche Begrifflichkeiten.
- Ab der (fachlichen) Konzeptionsphase sollten Meldungen standardisiert erfasst werden.
- Die Überprüfung sowohl der fachlichen als auch der Systemmeldungen sollte sowohl durch Usability-Experten als auch durch Experten aus der Anwendungsdomäne erfolgen.

4.0 Erstellung eines Meldungskonzepts

Grundsätzlich empfehlen wir, einen Prozess zur Erstellung von Meldungen frühzeitig zu definieren und den beteiligten Rollen zu kommunizieren.

4.1 Rollen

In großen Projekten sind oft die folgenden **Rollen** beteiligt:

- **Fachbereich**
Er kennt die fachlichen Anforderungen und aus seinem Kreis kommen oft die System-Anwender. Er ist während der Erhebung der Fachanforderungen beteiligt. Dabei entstehen oft bereits fachliche Meldungen. Auch steht er bei den späteren Reviews der Meldungen ggf. für Rückfragen zur Verfügung.
- **Requirements Engineer**
Er entwickelt die Systemanforderungen in Abstimmung mit dem Fachbereich und erstellt dabei oft auch Systemkonzepte. In dieser Phase entstehen mit die meisten Meldungen.
- **Implementierer/IT-Dienstleister**
Er implementiert das System. Während der Spezifikation und der Imp-

lementierung entstehen technische System-Meldungen.

- **Usability Engineer**
Er ist verantwortlich für die Qualitätsabsicherung der Meldungen und erarbeitet daher das Meldungskonzept und die Guidelines. Er führt das Review der erarbeiteten Meldungen durch.
- **Endanwender oder deren Vertreter**
Stehen zur Nutzerabsicherung bei den späteren Reviews der Meldungen für Usability Tests und ggf. Rückfragen zur Verfügung.

4.2 Prozess

Der Usability Engineer hat u.a. die Aufgabe, im Entwicklungsprojekt einen Prozess zur Erstellung von Meldungen aufzusetzen und auf seine Einhaltung zu achten. Dabei hat sich folgendes Vorgehen bewährt:

1. Erarbeitung eines projektspezifischen Meldungskonzepts und Erstellung von Guidelines
2. Aufsetzen von Glossar und Meldungs-Tabelle/Datenbank
3. Kommunikation der verbindlichen Vorgehensweise an alle beteiligten Rollen
4. Schulung der beteiligten Rollen (insb. Requirements Engineer)
5. Planung und Durchführung der Reviews der Meldungsdatenbank

Exemplarisch sei hier der in unseren Projekten empfohlene Prozess skizziert:

- Standardisierte Erfassung von Meldungen ab der Erhebung der Fachanforderungen
- Standardisierte Erfassung von Meldungen durch den Requirements Engineer während der Entwicklung der Systemanforderungen unter Berücksichtigung des Glossars
- Standardisierte Erfassung von technischen Systemmeldungen während der Systemspezifikation und der Umsetzung durch den IT-

Dienstleister unter Berücksichtigung des Glossars

- Review der Meldungen durch den Usability Engineer auf die Einhaltung der Guidelines. Ggf. Rückgriff auf Experten aus der Anwendungsdomäne (Fachbereich), Requirements Engineer oder Endanwender.

5.0 Anforderungen an die Benutzerführung insbesondere Meldungen

In der Norm DIN EN ISO 9241-13 [ISO9241-13] werden Verfahren zur Benutzerführung und Anforderungen an gute Meldungen beschrieben. Sie enthält Empfehlungen zur:

- Benutzerführung allgemein
- sprachlichen Gestaltung der Benutzerführung
- Eingabeaufforderung
- Rückmeldung
- Statusinformation
- Fehlermanagement
- Online-Hilfe

Alle diese Aspekte der Benutzerführung müssen beim Entwurf des Interaktionskonzepts berücksichtigt werden. Wegen ihrer speziellen Problematik beschränken wir uns in diesem Beitrag auf die Gestaltung von Meldungen. Unsere Empfehlungen basieren auf den Normen, der Literatur und unseren Erfahrungen aus zahlreichen Projekten.

5.1 Allgemeine Empfehlungen

Benutzerführung sollte den Benutzer nicht bei der Aufgabenerledigung stören oder den Fluss des Dialogs unterbrechen. Grundsätzlich sollten so wenige Meldungen wie möglich und so viele wie nötig generiert werden. Dabei ist auf folgendes zu achten:

- Klare Unterscheidbarkeit zwischen unterschiedlichen Systemmeldungs-typen → Abgrenzung: Hinweis-/Warn-/Fehlermeldungen
- Systemweit einheitlicher Aufbau und Verhalten. → Interaktionskonzept und Styleguide

- Systemweit einheitliche Syntax und Semantik → Glossar und Formulierungsleitfaden
- Die Formulierung der Meldungen muss möglichst konkret, präzise, spezifisch und trotzdem verständlich sein. → Nutzerabsicherung

5.2 Grundsätzlicher Aufbau

Eine Systemmeldung an den Nutzer sollte grundsätzlich kommunizieren:

- Was ist los?
Die Meldung, Beschreibung des Ereignisses bzw. des Zustandes.
- Warum?
Erläuterung der Gründe oder möglicher Gründe.
- Was kann ich tun?
Mögliche Handlungsoptionen des Anwenders, falls der Zustand ein Problem darstellt.

5.3 Sprachliche Formulierungen

In der Norm und der Literatur [Heinke 2004] werden zahlreiche Beispiele für die sprachliche Formulierung von Meldungen gegeben. Dazu gehören u.a.:

- Das Ergebnis einer Aktion sollte angegeben werden, bevor beschrieben wird, wie die Aktion ausgeführt wird.
Bsp.: „Um die Eingaben zu verwerfen, drücken Sie die Esc-Taste.“ anstelle von „Drücken Sie die Esc-Taste, um die Eingaben zu verwerfen.“
- Meldungen der Benutzerführung sollten so formuliert sein, dass sie mehr die Benutzerkontrolle als die Systemkontrolle betonen.
Bsp.: „Zum Speichern Ihrer Änderungen bitte [OK]-Taste drücken.“ anstelle von „Das System speichert Ihre Änderungen nur, wenn die [OK]-Taste gedrückt wird.“
- Im Allgemeinen sollten Nachrichten der Benutzerführung positiv formuliert werden und betonen, was zu tun ist, statt was zu vermeiden ist.
- Meldungen sollten grammatikalisch einheitlich, in kurzen einfachen Sät-

zen und emotional neutralen Worten formuliert werden.

- Meldungen sollten entsprechend der Terminologie formuliert werden, die Benutzer bei der Erledigung der Arbeitsaufgaben typischerweise verwenden.

5.4 Einheitliches Vokabular: Glossar

Qualitätsaspekte wie Lernförderlichkeit oder Erwartungskonformität erfordern Konsistenz in Syntax und Semantik an der Bedienschnittstelle.

Ein einheitlich verwendetes Vokabular ist eine Grundvoraussetzung dafür, und zwar nicht nur für die Anwender sondern auch für das gemeinsame Verständnis in größeren Entwicklungsteams. Ein Glossar ist eine Liste von Wörtern mit Erklärungen, definiert dieses Vokabular zentral und ermöglicht so eine einheitliche Begriffsverwendung.

6.0 Standardisierte Erfassung von Meldungen

Die in Kapitel 5.2 aufgeführten Bestandteile einer Meldung (Ereignis, Gründe, Handlungsoptionen) sind jeweils sehr fachspezifisch und können nachträglich nicht oder nur sehr schwer nacherhoben werden.

Aus diesem Grund muss bereits bei der Erhebung der Meldungen diese Struktur berücksichtigt werden.

6.1 Meldungsdatenbank

Hierfür erweist sich die zentrale und standardisierte Erfassung aller Meldungen in einem einheitlichen Format in einer Meldungstabelle oder Meldungsdatenbank als nützlich.

Da bereits bei der Erhebung von fachlichen Anforderungen Fehlerfälle und damit Gründe und Lösungsmöglichkeiten beschrieben werden, sollte mit der

standardisierten Erfassung von Meldungen bereits mit der Anforderungsanalyse begonnen werden. Gegebenenfalls werden diese früh erfassten Meldungen im Laufe der Entwicklung auf mehrere Systemmeldungen aufgeteilt oder umformuliert. Jedoch liegt dann zumindest der fachliche Hintergrund bereits vor.

Bei der Formulierung der Texte können Platzhalter sinnvoll sein (z.B. „Datei %1 konnte nicht geöffnet werden.“). Hierfür ist ein einheitliches Format zur Verwendung von Platzhaltern in den Texten zu definieren. Dies schließt eine Erläuterung der verwendeten Platzhalter mit ein (z.B. „%1 = Dateiname der Datei inklusive vollständiger Pfadangabe und Dateityp-Suffix“).

In allen entwicklungsrelevanten Dokumenten (z.B. Lasten-/Pflichtenheft, IT-Spezifikation, Schulungsunterlagen) sollte auf diese Meldungen mit einer Meldungs-ID referenziert werden. Dies gewährleistet die Rückverfolgbarkeit (Traceability) der Meldung zu ihrem Vorkommen im Anwendungskontext. Diese Information kann bei geplanten Änderungen an den Meldungstexten oder der Übersetzung in weitere Sprachen sehr wertvoll sein.

Die standardisierte Erfassung der Meldungen an zentraler Stelle bietet auch Vorteile bei der Qualitätssicherung, da der Stand der Reviews bzw. dessen Ergebnisse direkt in der Datenbank abgeleitet werden kann.

6.2 Beispielhafte Struktur

Im Folgenden werden die möglichen Attribute einer Meldungsdatenbank beschrieben, wie sie sich bewährt hat.

- Meldungs-ID
- Meldung
 - Text der Ereignismeldung
 - Text der Gründe
 - Text der Handlungsoptionen
- Variablendefinition (Platzhalter)

- Schweregrad der Meldung [Fehler, Warnung, Information]
- direkte Interaktion oder sofortige Aufmerksamkeit durch den Nutzer erforderlich
- Autor/Fachlicher Ansprechpartner
- Review-Status [OK, Beanstandet durch Nutzer/Fachbereich, Korrigiert]
- Review Gegenvorschlag

Wird das System mehrsprachig entwickelt, empfiehlt es sich, für jede Sprache in der Datenbank eine eigene Spalte für den Meldungstext vorzusehen.

7.0 Interaktionskonzept

Unterschiedliche Kategorien von Meldungen erfordern unterschiedliche Behandlung durch den Anwender.

Grob kann man Systemrückmeldungen in folgende Kategorien unterteilen:

- Meldungen in Folge von Aktionen/Funktionsaufrufen oder durch Änderungen am Systemzustand (die klassische „Fehlermeldung“)
- Meldungen in Folge von Nutzereingaben (schon während der Eingabe)
 - durch Formatprüfungen (z.B. Telefonnummer, Autokennzeichen)
 - Meldungen in Folge von Pflichtfeldprüfungen. Hier kann es auch komplexere Formen geben, z.B. mit Abhängigkeiten zu anderen Attributen, mit Abhängigkeiten zum Anwendungskontext.
- Bestätigungsaufforderungen
- Statusinformationen, die den aktuellen Zustand von Systemkomponenten anzeigen.

Zusätzlich zu den Kategorien muss auch der Schweregrad der Meldungen berücksichtigt werden. Üblicherweise unterscheidet man Fehler, Warnungen und Hinweise.

Auch kann zwischen Meldungen unterschieden werden, die die sofortige Aufmerksamkeit oder Interaktion des Anwenders erfordern oder im Ablauf auch später behandelt werden können.

Die Umsetzung im Interaktionskonzept der Anwendung ist oft abhängig von der verwendeten Client-Technologie. So erfordern z.B. klassische Web-Anwendungen zur Auswertung der Nutzereingabe einen Request-Response-Zyklus mit dem Server, wohingegen geeignet programmierte Web-Anwendungen oder Rich-Clients dem Nutzer schon während der Eingabe Rückmeldung geben können. Verwendet man Client-Frameworks lohnt ein Blick in die entsprechende Dokumentation oder Guidelines des Frameworks. Exemplarisch seien hier die „Eclipse User Interface Guidelines“ [Eclipse UI Guidelines] genannt. Diese empfehlen die Kommunikation von Fehlern, die eine explizite Nutzereingabe oder sofortige Aufmerksamkeit des Nutzers erfordern, mit einem modalen Dialog. Dies zwingt den Nutzer, davon Notiz zu nehmen und sich dem Problem anzunehmen. Fehler die dies nicht erfordern, werden in einer Fehlerliste zusammengestellt. Der Nutzer kann sich zu geeigneter Zeit darum kümmern.

8.0 Styleguide und Guidelines zur Erstellung von Meldungen

Um allen beteiligten Erstellern von Meldungen konkrete Hilfen an die Hand zu geben, wurde für das Projekt ein Styleguide und eine Reihe von Empfehlungen für die Formulierung von Meldungen fachlicher Art und für Systemmeldungen entwickelt und kommuniziert.

Ziel ist es, die visuelle und sprachliche Konsistenz und damit verbunden die Selbsterklärungsfähigkeit und Lernför-

derlichkeit des Systems sicherzustellen.

8.1 Styleguide

Der Styleguide soll die visuelle Konsistenz und den einheitlichen Aufbau der Meldungen an der Benutzungsoberfläche sicherstellen.

In jedem Fall sollten Schweregrad, Meldung, Grund und Lösungsmöglichkeiten kommuniziert werden

8.1.1 Syntax der Meldungen

Jede Nachricht ist nach folgender Struktur aufgebaut:

- Icon
- Meldung
- Grund
- Lösungsmöglichkeit

Für jedes dieser Strukturelemente gibt es entsprechende Empfehlungen zur Gestaltung und zur Syntax.

Icon

Das Icon visualisiert auf den ersten Blick, um welche Art von Meldung es sich handelt:



Hinweis

Für reine Mitteilungen und Systemfeedback an den Anwender.



Warnung

Für Meldungen des Anwendungsprogramms, die eine Reaktion des Anwenders benötigen.



Fehler

Für Systemfehler und Anwendungsfehler, die ein Weiterarbeiten ohne Eingriffe des Anwenders nicht erlauben.

Meldung

[Objekt] [Qualifizierer] [negierte Beziehung].

Beispiel:

Kopieren von „Batterie Sensor“ nicht möglich.

Grund

[Objekt] [Qualifizierer] [negierte Beziehung].

Beispiel:

Es ist nicht möglich, eine Komponente zu kopieren, die zu einem anderen Stamm gehört.

Lösungsmöglichkeit:

[Optional] [Lösungsmöglichkeit] [Objekt] [Qualifizierer] [muss Aktion]

Beispiel:

Bitte wählen Sie eine Version von „Batterie Sensor“ aus, die zum gleichen Stamm gehört.

8.1.2 GUI

Auch hier gilt, wie beim Interaktionskonzept, dass die grafische Gestaltung oft von der verwendeten Client-Technologie abhängig ist. Verwendet man Client-Frameworks lohnt sich daher ein Blick in die entsprechende Dokumentation oder in die Guidelines des Frameworks.

Für die aktuelle Anwendung wurde die klassische Form der Meldung gewählt und im Styleguide wie folgt dokumentiert:



Abb. 1: Beispiel einer Warnmeldung mit einfacher Interaktion.

8.2 Guidelines zur Formulierung der Meldungen

Den Entwicklern wurden einige wenige „Goldene Regeln“ zum Verfassen von Meldungen vorgegeben.

- Im Aktiv, nicht im Passiv formulieren.
- Keine doppelte Verneinung.
- Einfache Konstruktion der Sätze (Subjekt, Prädikat, Objekt).
- einheitliche Syntax (nicht: „SNR“, „Sachnummer“, „SNr.“), Glossar beachten.
- Grammatikalisch richtige und vollständige Sätze schreiben. Satzzeichen, Rechtschreibung beachten.
- Keine Anzeige von im Kontext überflüssigen Informationen.
- Besonders relevante Merkmale *konsistent* hervorheben bzw. kennzeichnen (beispielsweise Sachnummern in „Anführungszeichen“ setzen).
- Sowohl Meldung, Ursache als auch die Lösungsmöglichkeit jeweils in einem eigenen Satz beschreiben.

9.0 Fazit

Gute und verständliche Meldungen und Fehlermeldungen unterstützen und verbessern die Interaktion zwischen Mensch und Maschine. Dabei ist es besonders wichtig, bereits zur Anforderungsanalyse das Meldungs-

konzept zu entwerfen und an die Projektbeteiligten zu kommunizieren.

Das in diesem Beitrag vorgestellte Konzept könnte dafür die Basis bilden und kann projektspezifisch adaptiert werden. Die Guidelines zur Formulierung von Meldungen sind allgemein gültig und stellen eine Best Practice dar.

Häufig sind durch die Wahl der Client-Technologie, z.B. Java-Swing, .NET oder Web, das Interaktionskonzept und der Styleguide schon vorgezeichnet. Dann kann es aus Gründen der Erwartungskonformität wichtig sein, wesentliche Teile davon zu übernehmen.

Das Schreiben der Meldungen wird einem aber von keinem System abgenommen. Da sind die Kreativität und das Einfühlungsvermögen der Mitarbeiter gefragt. Wenn wir versuchen, uns in die Lage des Anwenders zu versetzen, werden wir Meldungen auch so formulieren, dass sie für den Anwender verständlich sind.

10.0 Literaturverzeichnis

Eclipse UI Guidelines, 2008, http://wiki.eclipse.org/User_Interface_Guidelines

Heineke, A. M. (2004): Mensch-Computer-Interaktion. Leipzig: Fachbuchverlag Leipzig.

ISO 9241-13, Ergonomische Anforderungen für Büroarbeiten mit Bildschirmgeräten, Teil 13: Benutzerführung, Beuth Verlag, 1000-08