

# Usability-Patterns verwenden

**Bereich**  
Planung und Design

**Aktivität**  
Oberflächenentwurf erstellen

## Ziele

- Usabilitymängel vermeiden
- Konsistenz von Benutzeroberflächen erhöhen
- Entwicklungseffizienz steigern

**schnell durchführbar** ★★★

**einfach durchführbar** ★★★

**agil einsetzbar** ★★★

## Motivation/Problemstellung

Bei der Entwicklung der Benutzeroberfläche treten immer wieder Gestaltungsfragen auf, die unmittelbare Auswirkungen auf die Usability des Softwareprodukts haben. Die zugrunde liegenden Entwurfsprobleme sind in der Regel weder einmalig noch neu. Insbesondere für weniger erfahrene UI-Designer bzw. Softwareentwickler ist es hilfreich, wenn sie in diesen Fällen auf bereits bewährte und gut dokumentierte Gestaltungslösungen, sog. Patterns, zurückgreifen können.

## Kurzbeschreibung

Usability-Patterns sind bewährte Lösungsmuster für typische Gestaltungs- bzw. Interaktionsprobleme, die in bestimmten Nutzungskontexten auftreten. Sie können beim Entwurf oder der Entwicklung von Software eingesetzt werden. Usability-Patterns sind in strukturierter, relativ abstrakter Form beschrieben. Sie ermöglichen es, die für ein Produkt relevanten Usability-Merkmale bereits in frühen Phasen der Produktentwicklung und ohne weitere Usability-Expertise systematisch zu betrachten und besser zu berücksichtigen.

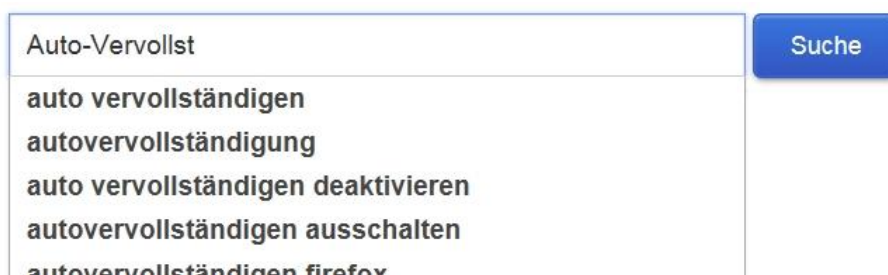


Abbildung 1: Beispielhafte Umsetzung des UsabilityPatterns „Auto-Vervollständigung“

## Input

- Identifizierte Gestaltungs- bzw. Interaktionsprobleme

## Output

- Passende, bewährte Musterlösungen

## Rahmenbedingungen

### Ausführender

UI- bzw. Softwaredesigner und -entwickler

### Werkzeuge, Hilfsmittel

Patternbeschreibung/-sammlung

### Vorkenntnisse/Erfahrungen

–

### Ort/Umgebung

Bildschirm-/Büroarbeitsplatz

### Weitere Teilnehmer

–

### Voraussichtliche Dauer

Abhängig von der jeweiligen Problemstellung

## Vorgehensweise

### Vorbereitung

Zunächst recherchieren Sie nach geeigneten Patterns und überprüfen, ob diese in Ihrem Kontext, also zur Lösung Ihrer konkreten Probleme beim Oberflächendesign und der Interaktionsgestaltung, anwendbar sind. Bekannte Patterns sind z. B. „Wizard“, „Fortschrittsanzeige“ oder „Auto-Vervollständigung“. Es gibt eine Vielzahl gut dokumentierter Sammlungen von UI Patterns, Interaction Patterns, UX Patterns und Design Patterns, die Sie bei der Gestaltung von Benutzungsschnittstellen nutzen können (s. weiterführende Informationen). Meist sind die Patterns in diesen Sammlungen nach verschiedenen Kategorien unterteilt (z. B. „Navigation“, „Suche“ oder „Layout“). Da die Patterns in der Regel einheitlich dokumentiert sind und in Beziehung zueinander stehen, werden diese Sammlungen auch als *Pattern languages* bezeichnet. Die dort beschriebenen Patterns haben sich in der Praxis als Musterlösungen für häufig auftretende Entwurfsprobleme herausgebildet. Die Patterns sind so abstrakt beschrieben, dass sie auf unterschiedliche Geschäftsdomänen übertragbar sind und in verschiedenartige Softwareentwicklungsprozesse integriert werden können. Für die Beschreibung der Patterns gibt es keinen verbindlichen Standard, aber es hat sich ein bestimmter Aufbau durchgesetzt. So enthalten die Patterns neben dem Namen in der Regel Informationen zum Kontext, zum Problem, zur vorgeschlagenen Lösung und zu bestimmten Anforderungen, die erfüllt sein müssen, damit das Pattern angewendet werden kann. Oft enthalten die Patterns auch konkrete Umsetzungsbeispiele, z. B. in Form von Bildern oder Pseudocode. Die Beschreibung der Patterns ist zum einen deskriptiv, denn sie dokumentiert und illustriert vorhandenes Erfahrungs- bzw. Expertenwissen; zum anderen ist sie präskriptiv, denn sie macht konkrete Vorgaben bzw. Empfehlungen, wie bei bestimmte Produkten bestimmte Usability-Merkmale erreicht werden können.

### Durchführung

Nach der Auswahl setzen Sie das beschriebene Usability-Pattern im eigenen Kontext um. Die konkrete Umsetzung instanziiert also das abstrakte Lösungsmuster. Die spezifischen Auswirkungen der Umsetzung hängen vom jeweiligen Usability-Pattern ab. Dies können einfache Anpassungen der Oberfläche sein; es kann aber auch die Anpassung von Funktionen bzw. Funktionsschritten oder die Entwicklung zusätzlicher Funktionen notwendig sein. Die Usability-Merkmale, die durch die Verwendung des Patterns gefördert werden, haben aus Benutzersicht in der Regel keine eigenständige Bedeutung, sondern ergänzenden Charakter. Sie verbessern oder erleichtern die Benutzung anderer (fachlicher) Funktionen des Produkts.

Die Verwendung von Usability-Patterns unterstützt Sie in diesen Bereichen:

- **Entwicklungseffizienz:** Prototypen bzw. Softwarelösungen können schneller entwickelt werden, wenn auf Patterns zurückgegriffen wird und keine eigenen Lösungen konzipiert werden müssen. Existierende Lösungen bzw. Lösungsansätze für viele Aufgabenstellungen machen komplexe Entwicklungsprojekte zudem besser beherrschbar.
- **Konsistenz:** Für gleiche Problemstellungen werden gleiche Lösungen herangezogen; dies sorgt für konsistente User Interfaces, die für den Nutzer verständlich und vorhersagbar sind. Insbesondere bei großen Entwicklungsteams wird Wildwuchs vermieden, die Synchronisierung von Arbeiten erleichtert (keine unnötigen Doppelarbeiten) und die Einhaltung von Entwicklungsrichtlinien unterstützt.
- **Code-Qualität:** Einmal implementierte Patterns können leicht wiederverwendet werden; dies vermeidet die Entwicklung von redundantem Code.
- **Qualitätssicherung:** Durch die Verwendung bewährter Patterns (bzw. getesteter, robuster Implementierungen) wird die Qualität verbessert und die Qualitätssicherung vereinfacht.
- **Kommunikation:** Verwenden alle Beteiligten bei der Produktentwicklung dieselben Patterns, so entsteht ein gemeinsames Verständnis für Probleme bzw. deren Lösung und es werden Missverständnisse vermieden.
- **Einarbeitung neuer Mitarbeiter:** Eine Patternsammlung, in der man auf einen Blick alle verfügbaren Elemente und Lösungen für Standardprobleme vorfindet, ist ein guter Einstieg für neue Designer und Entwickler.

### **Nachbereitung/Auswertung**

Bei Ihrer Arbeit können Sie Patterns aus zahlreichen öffentlichen Sammlungen nutzen. Unter Umständen ist es aber auch sinnvoll, für Ihre Projekte bzw. Produkte eine eigene Pattern Library aufzubauen. In einer solchen Library können Sie Musterlösungen zur Erreichung vielfältiger (Usability-)Aspekte dokumentieren. Diese Lösungen sind dann für alle am Produktentwicklungsprozess Beteiligten – z. B. Produktmanager, UI-Designer und Entwickler – zentral verfügbar und können von diesen einfach wiederverwendet werden.

### **Gütekriterien/Empfehlungen**

Usability-Patterns können nicht nur in frühen Entwicklungsphasen zum Lösen von Entwurfsproblemen eingesetzt werden, sondern sie können auch als Evaluierungswerkzeug angewendet werden, z. B. im Rahmen einer Usability-Pattern-Inspektion. Eine solche Inspektion dient dazu, Usability-Schwachstellen aufzudecken und konkrete Empfehlungen für die Verbesserung der Benutzeroberfläche zu liefern. Ausgehend von Benutzeraufgaben bzw. Aktivitäten wird während der Usability-Pattern-Inspektion überprüft, ob die entwickelten Masken und Dialoge mit einem oder mehreren auf die Situation passenden Patterns übereinstimmen. Da die Patterns allgemein verständlich beschrieben sind, kann diese Form der Usability-Evaluierung auch von Nichtexperten durchgeführt werden.

### **Risiken**

Verwenden Sie bei der Gestaltung von Oberflächen und Interaktionen immer wieder dieselben Entwurfsmuster, so besteht die Gefahr, dass die Nutzer das Produkt als schablonenhaft und langweilig empfinden. Andererseits können Sie die eingesparte Zeit nutzen, um kreative Lösungen für die wirklich herausfordernden Problemstellungen bei der UI-Gestaltung zu erarbeiten.

## Einordnung in den agilen Referenzprozess

### Mögliche Vorgänger

- Best Practices (PQ4Agile): –
- Sonstige Praktiken: (Kunden-)Anforderungen einarbeiten, Lösungskonzept entwerfen

### Mögliche Nachfolger

- Best Practices (PQ4Agile): Usability-Review durchführen
- Sonstige Praktiken: Prototypen erstellen, Softwareinkrement entwerfen, Softwareinkrement implementieren

### Mögliche Alternativen, verwandte Praktiken

- Best Practices (PQ4Agile): Experimentelles und exploratives Prototyping durchführen
- Sonstige Praktiken: Oberflächenentwurf erstellen

## Einordnung in das PQ4Agile-Qualitätsmodell

Beeinflusste Qualitätsmerkmale und -teilmerkmale (Softwarequalität):

Gebrauchstauglichkeit, Reife, Wartbarkeit, Effektivität, Effizienz, Zufriedenheit

Beeinflusste Qualitätsmerkmale und -teilmerkmale (Prozessqualität): Wiederverwendung, Einhaltung interner Standards, Skalierbarkeit

## Schlagworte

Entwurfsmuster, Gestaltungsmuster, Musterlösung, Lösungsschablone, Patternbibliothek

## Weiterführende Informationen

### Informationen im Internet

<http://www.usabilitypatterns.info/catalog/catalog.html>

<http://www.welie.com> (Patternsammlung und weiterführende Links)

### Literatur

Eelke Folmer & Jan Bosch (2003): Usability Patterns in Software Architecture. In: Constantine Stephanidis (Hrsg.): Proceedings of the Tenth International Conference on Human-Computer Interaction, S. 93–97. Lawrence Erlbaum Associates, New Jersey

Holger Röder (2012): Usability Patterns – Eine Technik zur Spezifikation funktionaler Usability-Merkmale. Cuvillier, Göttingen

Jenifer Tidwell (2010): Designing Interfaces. 2. Auflage. O'Reilly, Sebastopol

Best Practice „Usability-Patterns verwenden“

Version 2.0 – 16.09.2015 – Autor: Hartmut Schmitt, HK Business Solutions

Das Projekt PQ4Agile wird vom Bundesministerium für Bildung und Forschung im Rahmen der Maßnahme KMU-innovativ: IKT (01 | S13032) gefördert.