

## Pattern-Template

Deliverable E2.1

Projekt	USecureD – Usable Security by Design
Förderinitiative	Einfach intuitiv – Usability für den Mittelstand
Förderkennzeichen	01MU14002
Arbeitspaket	AP 2.1
Fälligkeit	31.01.2016
Autoren	Luigi Lo Iacono (TH Köln) Hartmut Schmitt (HK Business Solutions) Peter Nehren (TH Köln)
Status	Entwurf
Klassifikation	Intern



**HK Business Solutions GmbH**  
Hartmut Schmitt  
Mellinweg 20  
66280 Sulzbach  
schmitt@hk-bs.de

KMU  
(Konsortialführer)

---

**Technology**  
**Arts Sciences**  
**TH Köln**

**Technische Hochschule Köln**  
Prof. Dr.-Ing. Luigi Lo Iacono  
Betzdorfer Straße 2  
50679 Köln  
luigi.lo\_iacono@th-koeln.de

Hochschule  
(Konsortialpartner)

Mittelstand-  
Digital 

The logo for Mittelstand-Digital consists of three overlapping squares: a black one at the top, a yellow one at the bottom left, and a red one at the bottom right.

Gefördert durch:



aufgrund eines Beschlusses  
des Deutschen Bundestages

## Abstract

Im Projekt USecureD werden Musterlösungen und praxistaugliche Werkzeuge entwickelt, die kleine und mittlere Unternehmen (KMU) bei der Entwicklung bzw. bei der Auswahl betrieblicher Anwendungssoftware mit dem Qualitätsmerkmal „Usable Security“ unterstützen.

Ziel von Teilarbeitspaket AP 2.1 war es, Entwurfswerkzeuge wie Prinzipien, Entwicklungsrichtlinien und Patterns zu entwickeln, die Softwarearchitekten und Entwickler bei der Auswahl und Umsetzung benutzerfreundlicher Sicherheitsfunktionen und -mechanismen im Kontext betrieblicher Anwendungssoftware unterstützen. Im Rahmen dieser Arbeiten wurde eine Beschreibungstemplate für wiederverwendbare und übertragbare Gestaltungslösungen mit dem Qualitätsmerkmal Usable Security entwickelt.

## Schlagworte

Template, Beschreibungsvorlage, Schablone, Pattern, Entwurfsmuster, Gestaltungsmuster, Musterlösung

## Inhaltsverzeichnis

1	Vorgehensweise.....	4
2	Pattern-Template.....	5
3	Quellen.....	7

## 1 Vorgehensweise

Für jeden Softwarearchitekten und -entwickler ist es hilfreich, wenn er beim Lösen von Entwurfsproblemen auf bewährte, gut dokumentierte und wiederverwendbare Gestaltungsmuster oder Entwurfsmuster, sog. Patterns, zurückgreifen kann. Das Konzept der Patterns, also die Dokumentation bewährter Lösungsschablonen für wiederkehrende Entwurfsprobleme, kommt ursprünglich aus der Architektur [ALEXANDER ET AL. 1977] und wurde von Gamma [GAMMA 1995] und Beck [BECK 1996] erfolgreich auf den Bereich Softwarearchitektur übertragen. Norman [NORMAN & DRAPER 1986] machte erstmals den Vorschlag, Patterns auch im Bereich der Mensch-Computer-Interaktion einzusetzen; seitdem haben Patterns auch dort zunehmendes Interesse gefunden ([BORCHERS 2001], [TIDWELL 2005]). Bei der Gestaltung von Benutzerschnittstellen können Interaktionspatterns und Usability Patterns genutzt werden, um bewährte Lösungen zu definierten Problemen der Interaktionsgestaltung zu dokumentieren. Sie tragen damit zur Erhöhung der Konsistenz von Benutzeroberflächen, zur Steigerung der Entwicklungseffizienz und zur Einhaltung von Entwicklungsrichtlinien bei. Es gibt zahlreiche bekannte Sammlungen von Usability- und Interaktionspatterns ([VAN WELIE 2008], [TOXBOE 2016], [YAHOO 2016]).

Für die Domäne Usable Security gab es bisher keine Sammlung entsprechender Patterns. Durch eine umfassende Literaturrecherche des Konsortiums konnten insgesamt 47 Usable-Security-Patterns zusammengetragen werden ([EGELMAN 2009], [GARFINKEL 2005], [ARTEAGA 2009]).

Für die Dokumentation der Usable-Security-Patterns wurde im USecureD-Projekt ein Beschreibungstemplate entwickelt. Hierdurch sollte sichergestellt werden, dass alle erarbeiteten Usable-Security-Patterns in geeigneter und einheitlicher Form beschrieben sind und dass diese möglichst einfach auf unterschiedliche Geschäftsdomänen übertragen und nahtlos in beliebige Softwareentwicklungsprozesse integriert werden können. Zusätzlich zu bekannten Pattern-Attributen (Name, Kontext, Problem, Lösung usw., vgl. z. B. [E-TEACHING 2016]), mit denen grundlegende Informationen zum Pattern erfasst werden, wurden neue Attribute eingeführt (Abhängigkeiten, Beziehungen, Prinzipien, Use Cases, Check Listen, Entwicklungsrichtlinien, Tags, Log History); diese Attribute sollen unter anderem eine Verknüpfung mit den entsprechenden Projektergebnissen ermöglichen, die in USecureD erarbeitet werden.

Durch eine umfassende Analyse wurden Verknüpfungen und Abhängigkeiten der Patterns untereinander hergestellt. Dadurch wurde der Katalog zu einer sogenannten Mustersprache bzw. Pattern Language weiterentwickelt. Besonders hervorzuheben ist daher das Attribut *Beziehungen*, wodurch diese Verknüpfungen beschrieben werden. Auf der USecureD-Plattform werden diese als Links implementiert und als interaktive Grafik visualisiert, wodurch eine schnelle Navigation zwischen Patterns, die in einer Beziehung zueinanderstehen, gewährleistet werden kann. Im Rahmen dieser Analyse wurden den Patterns des Weiteren Prinzipien zugeordnet, wodurch in dem Attribut *Prinzipien* direkt ersichtlich wird, welchen abstrakteren Zielen bzw. Grundsätzen das Pattern dient.

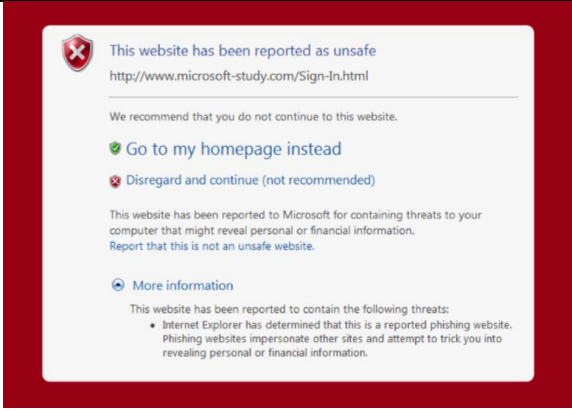
Im Hinblick auf eine effektive Durchsuchbarkeit der Pattern-Datenbank wurden unter dem Attribut *Tag* Schlagworte gesammelt, die dem jeweiligen Pattern zuzuordnen sind. Diese Schlagworte entstammen größtenteils aus dem USecureD-Qualitätsmodell. Damit wird gewährleistet, dass bei der Suche nach Patterns zur Verbesserung einzelner Merkmale des Qualitätsmodells, auch passende Patterns in der Datenbank gefunden werden können.

Darüber hinaus wurde eine *Log History*, d. h. ein Feld zur Dokumentation von Protokollereignissen, hinzugefügt. In diesem können vorgenommene Änderungen und Bearbeitungszeiten dokumentiert werden.

## 2 Pattern-Template

Name	<i>eindeutiger Name des Patterns</i>
Quellen	<i>Quellenangaben und Literaturhinweise zu dem beschriebenen Pattern</i>
Synonyme	<i>bekannte Synonyme bzw. anderssprachige Namen für das beschriebene Pattern</i>
Kontext	<i>Beschreibung der wiederkehrenden Situation bzw. Ausgangslage, in der sich ein bestimmtes Problem ergibt und in der das Pattern anwendbar ist</i>
Problem	<i>Beschreibung des akuten Problems im gegebenen Kontext einschließlich der Einflussfaktoren und Anforderungen (z. B. Sicherheits- oder Usabilityziele, Bedingungen, Einschränkungen), die sich hierdurch ergeben</i>
Lösung	<i>Beschreibung einer bewährten Entwurfsform, die eine Auflösung der beschriebenen Anforderungen ermöglicht</i>
Beispiele	<i>bekannte Verwendungen und Illustrationen des beschriebenen Patterns</i>
Implementierung	<i>detailliertere Informationen zu dem beschriebenen Pattern, z. B. Spezifikation von funktionalen/nichtfunktionalen Anforderungen oder Hinweise zu Architekturkonzepten oder zur Implementierung</i>
Konsequenzen	<i>Vor- und Nachteile des Patterns, die z. B. durch Einflussfaktoren oder Anforderungen entstehen, die miteinander in Konflikt stehen, und die vor der Verwendung des Patterns abgewogen werden sollten</i>
Abhängigkeiten	<i>Abhängigkeiten von anderen Patterns</i>
Beziehungen	<i>Patterns, die ein ähnliches Problem adressieren wie das beschriebene Pattern oder die in Kombination mit dem beschriebenen Pattern verwendet werden können</i>
Prinzipien	<i>Prinzipien, die ein übergeordnetes Ziel darstellen, zu dessen Erreichen das Pattern beiträgt</i>
Richtlinien	<i>USecureD-Entwicklungsrichtlinien, bei deren Umsetzung das Pattern berücksichtigt werden kann</i>
Check Listen	<i>Checklisten, mit denen überprüft werden kann, ob das Pattern korrekt umgesetzt wurde</i>
Use Cases	<i>USecureD-Use-Cases, bei deren Umsetzung das Pattern verwendet werden kann</i>
Tags	<i>Zum Pattern passende Schlagworte, um die Durchsuchbarkeit des Katalogs zu verbessern</i>
Log History	<i>Feld zur Protokollierung von Ereignissen, wie zum Beispiel das aktuelle Updatedatum des Pattern</i>

### 3 Beispiel Pattern

Name	<i>Empfehlungen bereitstellen</i>
Quellen	<i>Egelman, Serge. 2009. Trust Me: Design Patterns for Constructing Trustworthy Trust Indicators.</i>
Synonyme	<i>keine</i>
Kontext	<i>Wird eine Gefahr erkannt, sollte dem Nutzer eine klare Empfehlung zum sicheren Fortfahren präsentiert werden und eine Liste mit anderen möglichen Aktionen angezeigt werden.</i>
Problem	<i>Viele Warnungen schlagen fehl, und zwar nicht, weil der Nutzer nicht versteht, was die Gefahr ist, sondern weil ihm keine klaren Vorschläge gegeben werden, wie er die Gefahr umgehen kann.</i>
Lösung	<i>Warnmitteilungen müssen dem Nutzer eine weitere Vorgehensweise vorschlagen und eine Anleitung geben, wie er diese umsetzen kann.</i>
Beispiel	
Implementierung	<i>Die empfohlene Aktion sollte markanter sein als alle anderen Optionen. Dadurch wird dem Nutzer schnell verständlich, was die empfohlene Aktion ist, auch wenn er nicht die detaillierte Beschreibung liest und direkt zu den Optionen springt. Die verfügbaren Optionen sollten so gestaltet werden, dass es einfach ist, zwischen der empfohlenen Option und den übrigen Optionen zu unterscheiden.</i>
Konsequenzen	<i>In vielen Fällen wird der Nutzer nur den Titel der Warnung lesen und dann direkt zu den verfügbaren Optionen springen. Sind keine Optionen verfügbar, wird der Nutzer sehr wahrscheinlich keine fundierte Entscheidung treffen (beispielsweise die Warnung ignorieren, wenn keine Empfehlungen angezeigt wurden).</i>
Abhängigkeiten	<i>keine</i>
Beziehungen	<p><i>[Dialoge mit Handlungsempfehlungen]</i></p> <p><i>[Empfohlene Optionen]</i></p> <p><i>[Bemerkbare kontextabhängige Hinweise]</i></p>
Prinzipien	<i>[Sichtbarkeit]</i>
Tags	<i>Fehlersicherheit, Erlernbarkeit</i>
Log History	<i>[01.01.2016]: Translated to german</i>

## 4 Quellen

- [ALEXANDER ET AL. 1977] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, Shlomo Angel (1977): A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York
- [BECK 1996] Kent Beck (1996): Smalltalk Best Practice Patterns. Prentice Hall, Upper Saddle River
- [BORCHERS 2001] Jan O. Borchers (2001): A Pattern Approach to Interaction Design. John Wiley & Sons, Chichester
- [E-TEACHING 2016] e-teaching.org Redaktion (2015): Bestandteile eines Entwurfsmusters. Verfügbar unter: [https://www.e-teaching.org/didaktik/konzeption/entwurfsmuster/beschreibungsformat/index\\_html](https://www.e-teaching.org/didaktik/konzeption/entwurfsmuster/beschreibungsformat/index_html). [09.03.2016]
- [GAMMA 1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston
- [NORMAN & DRAPER 1986] Donald A. Norman, Steven W. Draper (1986): User Centered System Design: New Perspectives on Human-computer Interaction. CRC Press, Boca Raton
- [TIDWELL 2005] Jenifer Tidwell (2005): Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly, Sebastopol
- [TOXBOE 2016] Anders Toxboe (2014): UI Patterns - User Interface Design Pattern Library. Verfügbar unter: <http://ui-patterns.com> [09.03.2016]
- [VAN WELIE 2008] Martijn van Welie (2008): A Pattern Library for Interaction Design. Verfügbar unter: <http://www.welie.com> [09.03.2016]
- [YAHOO 2016] Yahoo Developer Network (2016): Yahoo Design Pattern Library. Verfügbar unter: <http://developer.yahoo.com/ypatterns> [09.03.2016]
- [EGELMAN 2009] Egelman, Serge. 2009. Trust Me: Design Patterns for Constructing Trustworthy Trust Indicators
- [GARFINKEL 2005] Garfinkel, Simson L. 2005. „Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable“. Dissertation, Massachusetts Institute of Technology
- [ARTEAGA 2009] Muñoz-Arteaga, Jaime et al. 2009. „A Method to Design Information Security Feedback Using Patterns and HCI-Security Criteria“. In Computer-Aided Design of User Interfaces VI, 283–94. Springer London. doi:10.1007/978-1-84882-206-1\_26