entwickler.press
shortcuts

Usable Security und Privacy by Design

Hartmut Schmitt, Peter Nehren,
Luigi Lo Iacono und Peter Leo Gorski

Hartmut Schmitt, Peter Nehren, Luigi Lo Iacono und Peter Leo Gorski

Usable Security und Privacy by Design

ISBN: 978-3-86802-759-4

© 2017 entwickler.press

Ein Imprint der Software & Support Media GmbH

1 Benutzerzentrierte Entwicklung von Sicherheitsfunktionen

Der Begriff „Usable Security and Privacy by Design“ bezeichnet Methoden und Verfahrensweisen in der Entwicklung von Software und technischen Produkten, bei denen der Benutzer im Mittelpunkt der Entwicklung von Sicherheits- bzw. Datenschutzkomponenten steht.

„Benutzer“ meint in diesem Zusammenhang nicht nur den Anwender der Software, sondern auch deren Entwickler. Letzterem kommt dabei eine besondere Bedeutung zu. Denn die komplexen Sicherheits- und Datenschutzsachverhalte müssen in verständlicher Art und Weise in Methoden, Werkzeugen, APIs, Bibliotheken und Frameworks integriert sein, damit nicht funktionale Anforderungen wie Sicherheit korrekt umgesetzt werden können – korrekt im Sinne der effektiven Implementierung der Sicherheitsmechanismen, aber auch der Benutzbarkeit durch den Zielanwender. Das erste Kapitel des shortcuts zum Thema „Usable Security“ dient als Einführung in die Thematik. Es soll insbesondere für Softwareentwickler einen Überblick bieten und für die Probleme bei der Entwicklung gebrauchstauglicher Sicherheitsfunktionen sensibilisieren.

Die Digitalisierung schreitet stetig voran und hält in allen Bereichen des Lebens Einzug – ob im vernetzten Zuhause (Smart Home), in unzähligen Gegenständen (Internet of Things) oder im großen Rahmen in der Industrie (Industrie 4.0). Die Vernetzung durch Cloud Computing und neue Technologien zur Verarbeitung und Analyse von Big Data können einen entscheidenden Beitrag dazu leisten, die alltäglichen Dinge des digital vernetzten Zeitalters einfacher zu gestalten. Hierbei werden jedoch online

riesige Datenmengen abgelegt und bearbeitet, darunter zunehmend sensible Daten von Unternehmen, Organisationen und Privatpersonen. Um die damit einhergehenden Risiken zu minimieren, fordern Benutzer verstärkt geeignete Sicherheitskomponenten. Die Folge: Zu dem enormen Angebot an Endgeräten, Software und Onlineservices gesellt sich eine ebenso große Anzahl von (häufig optionalen) Sicherheitswerkzeugen. Die Anwender sehen sich mit Passwörtern und Passwortmanagement, Virenscannern, Malwareschutz, Firewalls, Kommunikations- und Datenverschlüsselung konfrontiert, ebenso mit den damit verbundenen Updates, Einstellungen und Warnungen. Für Entwickler bringt das die zusätzliche Herausforderung mit sich, die essenziellen Sicherheitsmechanismen in einer Weise auszugestalten, wie es allgemein von gebrauchstauglichen Systemen erwartet wird [1], das heißt, so, dass sie von den Zielanwendern effektiv, effizient und zufriedenstellend verwendet werden können.

Usability vs. Security?

Die eigene Erfahrung lässt erahnen, dass bei der Nutzung von Sicherheitssoftware bzw. von Systemkomponenten, die Sicherheitsfunktionen bereitstellen, die Usabilitykriterien „effektiv“, „effizient“ und „zufriedenstellend“ oft nicht erfüllt werden. Einschlägige wissenschaftliche Studien belegen diesen Eindruck [2], [3], [4]. Sicherheit und Usability werden häufig sogar als widersprüchliche Komponenten eingeschätzt, die nicht unter einen Hut zu bekommen sind. Wenn Sicherheitslösungen ohne Berücksichtigung der menschlichen Eigenschaften und Fähigkeiten sowie des Anwendungskontexts entwickelt werden, stellen sie eine Barriere dar, die beim Erledigen der eigentlichen primären Aufgaben im Weg steht [4]. Das kann dazu führen, dass Sicherheitssysteme oder Sicherheitsvorgaben umgangen werden,

wodurch Sicherheitskonzepte zu Fall gebracht werden können. Das folgende Beispiel aus der Praxis [3] verdeutlicht das: Mobile fahrbare Terminals sind in Krankenhäusern ein verbreitetes Mittel zur Unterstützung der Visite. Zur Authentifizierung des Benutzers kommen hier häufig Verfahren zum Einsatz, die einer Interaktion des Arztes bedürfen (z. B. Passwort eingeben oder Krankenhaus-Smartcard einlesen). Das Abmelden erfolgt entweder explizit oder automatisch nach Verstreichen einer festgelegten Zeitspanne. Insbesondere Letzteres erweist sich für den Arzt im Nutzungskontext als ineffizient, da z. B. bei längeren Gesprächen mit dem Patienten während der Visite ein wiederholtes Anmelden am System erforderlich ist. Das Erweitern der impliziten Abmeldung durch ein Gerät zur Entfernungsmessung sollte hier eine Verbesserung bewirken. Außer Reichweite geraten, wird der angemeldete Anwender automatisch abgemeldet. Dieser Ansatz erwies sich in der Praxis allerdings als absolut untauglich. Abhilfe wurde von den frustrierten Anwendern mittels einfacher Styroporbecher geschaffen, die über den Entfernungsmesser gestülpt wurden. So konnte das Gerät nicht mehr feststellen, ob man sich vom Computer entfernt hat. Auch für das zeitlich gesteuerte Abmelden ist eine „Lösung“ gefunden worden: Die Assistenzärzte waren dafür verantwortlich, regelmäßig auf die Space-Tasten der Tastauren zu drücken, um den Log-out zu verhindern.

In den letzten Jahren hat sich gezeigt, dass Sicherheitskomponenten ein großes Problem für viele Verbraucher darstellen. Im IT-Grundschutzkatalog des Bundesamts für Sicherheit in der Informationstechnik (BSI) befindet sich eine Liste mit über 100 verschiedenen menschlichen Fehlhandlungen [5]. Dabei rühren viele Fehlhandlungen daher, dass festgelegte Sicherheitsvorschriften und Sicherheitsmechanismen bzw. die Entwickler dieser

Mechanismen Anforderungen an die Anwender stellen, die diese kognitiv nicht in der Lage sind zu lösen [6]. Software soll für den Benutzer auf der einen Seite möglichst einfach und intuitiv sein – mit dem Ziel, dass er seine Aufgabe effektiv und effizient bearbeiten kann. Auf der anderen Seite sollen die erzeugten oder bearbeiteten Informationen und Daten sicher sein. Somit lautet die Kernfrage: Wie lässt sich ein höherer Grad an Sicherheit erreichen, ohne das Nutzererlebnis zu beeinträchtigen? Genau dieser Problemstellung widmet sich der Forschungszweig „Usable Security and Privacy“.

Usable Security and Privacy

Gebrauchstaugliche Informationssicherheit zeichnet sich durch interdisziplinäre benutzerzentrierte Forschung und Zusammenarbeit aus: Modelle aus der Psychologie, die die Möglichkeiten und Grenzen der kognitiven Leistungsfähigkeit aufzeigen, sind ebenso wichtig wie Ergebnisse aus den Forschungsbereichen der Mensch-Computer-Interaktion oder des Designs. Hier wird zum Beispiel in Studien evaluiert, an welchen Stellen und warum Benutzer Probleme bei der Benutzung von Sicherheitssoftware haben. Die daraus gewonnenen Erkenntnisse und Anforderungen fließen in die interdisziplinäre Forschung zurück.

In den letzten Jahren haben sich verschiedene Konferenzen und Workshops etabliert, die ein Forum für den direkten fachübergreifenden Austausch zu den hier dargestellten Themen bieten. Ein reger Wissensaustausch auf internationaler Ebene findet bei dem vom „Usable Privacy and Security Laboratory“ der Carnegie Mellon University ins Leben gerufenen „Symposium On Usable Privacy and Security“ (SOUPS, *http://cups.cs.cmu.edu/soups*) statt, ebenso bei dem von der IFIP Technical Committee 11 Arbeitsgruppe 12 geförderten

„International Symposium on Human Aspects of Information Security & Assurance“ (HAISA, *http://haisa.org/*) und der „International Conference on Human Aspects of Information Security, Privacy and Trust“ (HAS, *http://2017.hci.international/has*). Auf Workshops wie dem „Workshop on Socio-Technical Aspects in Security and Trust“ (STAST, *http://stast.uni.lu/*), dem „New Security Paradigms Workshop“ (NSPW, *http://www.nspw.org/*) oder dem „Usable Security and Privacy“ Workshop auf der „Mensch und Computer“ Konferenz (*http://www.mensch-und-computer.de/*) lassen sich ebenfalls gezielt Wissen und Fähigkeiten in dem Themengebiet erweitern. „Usable Security and Privacy“ ist aber längst auch auf den großen Topkonferenzen wie der ACM, der CCS und der USENIX anerkannt und mit einzelnen Sessions oder Workshops vertreten.

Auf Grundlage der bereits geleisteten Arbeiten und wissenschaftlichen Erkenntnissen kann ein konkretes Porträt des im Fokus stehenden Benutzers gezeichnet werden. Dabei betrachten wir in diesem Kapitel den Anwender getrennt vom Entwickler, da er eine besondere Rolle im Entwicklungsprozess benutzertauglicher Informationssicherheit einnimmt.

Benutzerzentriertes Security Engineering

Als Voraussetzung für ein benutzerzentriertes Security Engineering ist ein gewisses Maß an Verständnis über die Bedürfnisse der Anwender wichtig. Lange Zeit wurden diese zu Unrecht als schwächstes Glied in der Kette betrachtet, denn Informations- und Kommunikationstechnologie-(IKT-)Sicherheitssysteme stellen Anforderungen, die von durchschnittlichen Nutzern nur schwer gänzlich erfüllt werden können [4]. Wie erfolgreich ein Nutzer Sicherheitsfunktionen von Software anwendet, ist von mehreren Faktoren abhängig. Einen

systematischen Ansatz zur Identifizierung von Fehlerquellen eines Sicherheitssystems verfolgt Lorry Cranors Sicherheitsframework „human-in-the-loop“ [7]. Der Name leitet sich davon ab, dass bisherige Sicherheitssysteme nicht ohne die Interaktion mit Menschen auskommen (**Abb. 1.1**).

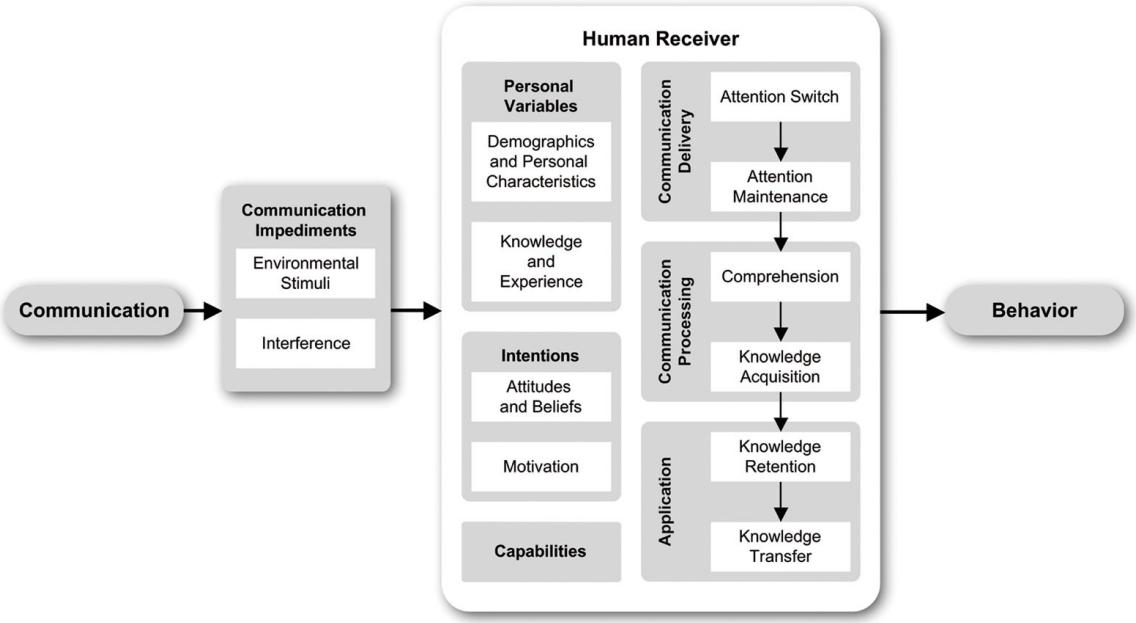


Abbildung 1.1: Das „human-in-the-loop“-Sicherheitsframework von Lorry Cranor [7]

Aus dem Modell geht hervor, dass das Verhalten eines Menschen, ausgelöst durch eine sicherheitsrelevante Kommunikation, in erster Linie beeinflusst ist

- von individuell geprägten Variablen wie demografischen Faktoren, der Persönlichkeit, gemachten Erfahrungen sowie erworbenen Kompetenzen,
- von der Absicht dieses Menschen, also seiner Einstellung und Motivation, und

- von den Fähigkeiten des Menschen.

Der „DsiN-Sicherheitsindex 2015“ unter der Schirmherrschaft des Bundesministeriums des Innern hat die Verbraucher in vier Typen differenziert: die souveränen, gutgläubigen, fatalistischen und außenstehenden Nutzer [8]. Sie unterscheiden sich besonders offensichtlich in Bezug auf die drei genannten Variablen. Welche Ansätze existieren also, um diese Stellschrauben in eine positive Richtung zu drehen?

Die Steigerung der Sicherheitskompetenz von Anwendern und die Förderung von Bewusstseinsbildung bezeichnet der DsiN-Sicherheitsindex als „Digitale Aufklärung 2.0“ [8], jedoch nicht im speziellen Bezug auf gebrauchstaugliche Informationssicherheit. Sie umfasst die Aus- und Weiterbildung von Personal in Form von Schulungen und durch die Thematisierung in Ausbildungen und Studiengängen. Es bedarf allerdings auch Arbeitshilfen, um ein Verständnis für gebrauchstaugliche Sicherheit aufzubauen, denn es existiert bis dato nur sehr wenig Fachliteratur, die zudem vornehmlich in englischer Sprache verfasst ist [9], [10]. Zudem sind die Entwicklung und Bekanntmachung von Software mit Vorbildcharakter wichtig, damit sich auf Seiten der Nutzer ein gewisser Anspruch an Software etablieren kann.

Auf der anderen Seite müssen Sicherheitsmechanismen von IKT-Systemen die ihnen zugrunde liegenden Sicherheitskonzepte verständlich und transparent machen. Bei der Verarbeitung der Kommunikation (Communication Processing) sowie beim Schritt der Anwendung (Application) durch den menschlichen Empfänger (**Abb. 1.1**), wird dieser in der Regel von einem mentalen Modell geleitet [11]. Das ist abhängig von den oben genannten Faktoren und aus diesem Grund sehr individuell. Dabei muss das Modell keineswegs richtig, logisch oder für andere Menschen nachvollziehbar sein. Ein

Sicherheitsmechanismus sollte aus diesem Grund so transparent sein, dass die Anwender sich ein möglichst passendes mentales Modell des Konzepts machen können. Die Integration von asymmetrischen Verschlüsselungsverfahren in Software hat sich in der Vergangenheit als nicht plausibel genug für die Nutzer erwiesen [2]. Die Unterstützung von mentalen Modellen ist besonders schwierig, wenn sich für das zugrunde liegende Sicherheitskonzept kein Pendant in der bekannten realen Welt finden lässt [12].

Die Nichtberücksichtigung von menschlichen Fähigkeiten, aber auch von deren zeitlichen Ressourcen hat einen negativen Effekt auf die Motivation und damit auch auf die Produktivität von Benutzern [13].

Bereits angesprochen wurden die kognitiven Fähigkeiten, die eine natürliche Grenze für jeden einzelnen darstellt. Authentifizierungsverfahren sind eines der Kernthemen der Usable-Security-Forschung und erweisen sich an dieser Stelle bis heute als Herausforderung. Trotz zahlreicher Vorschläge alternativer Authentifizierungsverfahren werden in der Praxis immer noch überwiegend Passwörter eingesetzt. Das gilt auch für Geräte mit Touchscreens, bei denen der Vorgang des Eintippens von Sonderzeichen dreimal länger dauert als bei Tastaturen [13]. Besonders die steigende Anzahl und notwendige Komplexität überfordert die kognitiven Fähigkeiten der Menschen [4], [14]. Die derzeit vielversprechendsten Lösungsansätze basieren auf „impliziter Authentifizierung“ [13], bei denen der Authentifizierungsvorgang z. B. durch biometrische Merkmale oder Nutzungsprofile erfolgt und kein geheimes Passwort eingefordert wird. Prominente Beispiele sind das Nymi-Band [15], das die Wellen eines Elektrokardiogramms auswertet, oder die FIDO-Spezifikationen [16], die verschiedene

biometrische Erkennungsmerkmale berücksichtigen.

Entwicklerzentriertes Security Engineering

Es lässt sich beobachten, dass bisher vor allem der Anwender im Fokus der Usable-Security-Forschungen steht. Die Gruppe der zu berücksichtigenden Benutzer von Softwaresicherheitskomponenten reicht jedoch deutlich weiter. Softwareentwickler müssen bei ihren primären Aufgaben mit sehr viel speziellerer Software arbeiten – mit Software, die von Entwicklern für Entwickler entwickelt wurde. Bereits existierende Softwarebausteine sind ein elementares Werkzeug bei dieser Arbeit. Nur so ist es oft möglich, der steigenden Komplexität von Software und den verkürzten Entwicklungszeiten gerecht zu werden. Bei diesen Bausteinen muss die Gebrauchstauglichkeit eine besondere Berücksichtigung finden. Besonders die Implementierung von grundlegenden Sicherheitsmechanismen wie Verschlüsselung, digitale Signatur oder das Management kryptografischer Schlüssel ist aufgrund ihrer Komplexität enorm fehleranfällig, weshalb viele Entwickler auf Implementierungen von spezialisierten Programmierern aufbauen. Eine Studie des „European Center for Security and Privacy by Design“ hat längst den Bedarf an mehr von solchen Securitybausteinen identifiziert [17]. Gebrauchstaugliche Sicherheit muss an dieser Stelle entwicklerzentriert greifen, um einer falschen oder fehlerhaften Verwendung z. B. von APIs, Konfigurationen oder Patterns entgegenzuwirken.

Für die Anwender steht die Funktionalität eines Systems im Vordergrund. Das spiegelt sich entsprechend im Prozess der Softwareentwicklung wider: Die Konzentration liegt hier klar auf funktionalen Anforderungen. Die nicht funktionalen Anforderungen Usability und Sicherheit stehen in der Regel nicht im Mittelpunkt der Entwicklungsprozesse, sondern finden oft erst

am Ende Berücksichtigung. Fehler, die bereits im Laufe der Entwicklung passieren, führen jedoch unter Umständen dazu, dass Anwender überhaupt nicht mehr in der Lage sind, Sicherheitsfunktionen anzuwenden. Eine gravierende softwareseitige Sicherheitslücke kann dafür sorgen, dass die Bemühungen der Anwender umsonst sind, wie folgende zwei Beispiele zeigen.

Fahl et al. [18] fanden heraus, dass die bisherige Herangehensweise sowie der Mangel an gebrauchstauglichen Securitybausteinen beim Betriebssystem Android dazu geführt haben, dass die SSL-Validierung in einer Vielzahl von Applikationen nicht existent ist. Die Gruppe deutscher Wissenschaftler stellte fest, dass von den Android-Entwicklern eine Reihe von „Trust Manager“- und „SSL Socket Factory“-Klassen eingesetzt wurden, die zwar vom Namen her (z. B. „SimpleX509TrustM“ oder „EasySSLSocketF“) eine einfache Anwendung versprechen, tatsächlich aber dazu führen, dass die Applikationen ohne Ausnahme jedem Zertifikat vertrauen. Mangels Zeit, Expertise oder auch aus blindem Vertrauen validierten die Entwickler die Implementierungen der adaptierten Klassen nicht.

Ein weiteres Beispiel stammt vom August 2015 [19], [20]: In den USA wurden kritische Sicherheitslücken in den Sicherheitskontrollen von Flughäfen aufgedeckt. Unabhängige Wissenschaftler fanden hartkodierte Zugangsdaten bei Röntgendetektoren, Sprengstoffdetektoren und Stechuhren. Benutzername und Passwort sind bei allen Geräten der gleichen Modelle identisch. Dabei ist das Risiko von im Quellcode festgelegten Zugangsdaten ein altbekanntes Problem, das z. B. auf der 2011 veröffentlichten CWE/SANS-Liste der 25 gefährlichsten Softwarefehler auf Platz 7 geführt wurde [21]. Solche Beispiele

zeigen eindrücklich, dass auch Entwickler als Anwender in Zukunft stärkere Berücksichtigung in der Usabilityforschung finden müssen.

By Design

Im Rahmen des Projekts „USecureD“ (Kasten: „Über USecureD“) wird an einem Qualitätsmerkmal als Indikator für Usable Security and Privacy by Design gearbeitet. Das Merkmal wird Softwarelösungen auszeichnen, die das Ergebnis kontinuierlicher und zyklischer Zusammenarbeit von interdisziplinärer Wissenschaft, praktischem Know-how von Softwareentwicklern und der Evaluation der Gebrauchstauglichkeit durch den Anwender sind.

Dieser Ansatz geht damit einen Schritt weiter als bisherige Herangehensweisen, denn er bezieht insbesondere die gesamte Produktionskette der Softwareentwicklung (by Design) mit ein. Erklärtes Ziel sind insbesondere Lösungen, die es auch Entwicklern in kleinen und mittelständischen Unternehmen ermöglichen, mithilfe eines Vorgehensmodells gebrauchstaugliche Informationssicherheit in ihren Produkten zu implementieren. Praxistaugliche Werkzeuge, die sich passgenau in etablierte Prozesse der Softwareentwicklung integrieren lassen, sollen dabei helfen, bewährte Musterlösungen praktisch einzusetzen. Dazu wird das Projekt auch die Anforderungen von Unternehmen und Entwicklern systematisch analysieren, die diese an Entwicklungsprozesse für sichere und gebrauchstaugliche Software stellen.

Über USecureD

USecureD – Usable Security by Design

Bewährte Musterlösungen und praxistaugliche Werkzeuge für die Entwicklung und Auswahl betrieblicher Software mit dem

Qualitätsmerkmal Usable Security. Die Projektpartner sind:

- HK Business Solutions GmbH, Sulzbach (Gesamtprojektleitung)
- Technische Hochschule Köln

Assoziierte Partner:

- TU Berlin, Quality and Usability Lab am Institut für Softwaretechnik und Theoretische Informatik
- Fraunhofer-Institut für Experimentelles Software Engineering IESE
- Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn
- saar.is – saarland.innovation&standort e. V.
- Ha-Ra Umwelt- und Reinigungstechnik GmbH
- Bruno Zimmer e. K.

Laufzeit: Mai 2015 – April 2017

Das Projekt wird vom Bundesministerium für Wirtschaft und Energie im Rahmen der Förderinitiative „Einfach intuitiv – Usability für den Mittelstand“ gefördert (01MU14002).

Mehr Informationen: *www.usecured.de*

Fazit

Technische Produkte und Software kommen aufgrund des erreichten Grads der Vernetzung nicht mehr ohne Sicherheitskomponenten aus. Bei der Entwicklung solcher Systeme müssen die Anforderungen und Bedürfnisse der Benutzer besonders berücksichtigt werden. Dadurch soll ein höherer Grad an Sicherheit ermöglicht werden, ohne das Nutzererlebnis zu beeinträchtigen. Softwareentwickler spielen dabei eine besonders zentrale Rolle im Entwicklungsprozess gebrauchstauglicher Sicherheit, da sie auf der einen Seite

Sicherheit für Anwender implementieren und auf der anderen Seite ebenfalls Anwender von Sicherheitskomponenten zur Softwareentwicklung sind. Aus diesem Grund muss diese spezielle Nutzergruppe stärker in die Usable-Security-Forschung miteinbezogen werden. Insbesondere gebrauchstaugliche und effektive Softwarewerkzeuge werden dringend benötigt. Der shortcut möchte besonders Softwareentwicklern eine Einführung in das Thema Usable Security and Privacy geben und auf Probleme bei der Entwicklung gebrauchstauglicher Sicherheitsfunktionen aufmerksam machen.

Links & Literatur

[1] ISO 9241-11 (1998): „Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 11: Anforderungen an die Gebrauchstauglichkeit – Leitsätze“

[2] Whitten A.; Tygar, J. D.: „Why Johnny Can’t Encrypt“, in: Proceedings of the 8th USENIX Security Symposium, 1999, S. 169–184

[3] Blythe, J.; Koppel, R.; Smith, S. W.: „Circumvention of Security: Good Users Do Bad Things“, IEEE Security & Privacy, Vol.11, No. 5, 2013, S. 80–83

[4] Adams, A.; Sasse, M. A.: „Users are not the enemy“, Communications of the ACM, Vol. 42 No. 12, 1999, S. 40–46

[5] BSI (2014): „G 3 Menschliche Fehlhandlungen“, Online: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschu

[6] Howe, A. E.; Ray, I.; Roberts, M.; Urbanska, M.; Byrne, Z.: „The Psychology of Security for the Home Computer User“, IEEE Symposium on Security and Privacy 2012, S. 209–223

[7] Cranor, L. F.: „A Framework for Reasoning About the Human in the Loop“, in: Proceedings of the 1st Conference on Usability,

Psychology, and Security 2008

[8] Deutschland sicher im Netz e.V.: „DsiN-Sicherheitsindex 2017 | Digitale Sicherheitslage der Verbraucher in Deutschland“, 2017

[9] Cranor, L. F.; Garfinkel, S.: „Security and Usability: Designing Secure Systems that People Can Use“, O’Reilly and Associates, 2005

[10] Garfinkel, S.; Richter Lipford, H.: „Usable Security (Synthesis Lectures on Information Security, Privacy, and Trust)“, Morgan & Claypool, 2014

[11] Gentner, D.; Stevens, A. L.: „Mental Models“, Hillsdale NJ: Lawrence Erlenbaum Associates, 1983

[12] Fischer-Hübner, Simone; Iacono, Luigi Lo; Möller, Sebastian: „Usable Security und Privacy“, DuD • Datenschutz und Datensicherheit 11, 2010

[13] Sasse, M. A.: „Technology should be smarter than this!: A vision for overcoming the great authentication fatigue“, Secure Data Management: 10th VLDB Workshop, 2014, S. 33–36

[14] Komanduri, S.; Shay, R.; Kelley, P. G.; Mazurek, M. L; Bauer, L.; Christin, N.; Cranor L. F.; Egelman, S.: „Of passwords and people: measuring the effect of password-composition policies“, in CHI ’11: Proceeding of the 29th SIGCHI Conference on Human Factors in Computing Systems, 2011

[15] Nymi Inc, 2015: „Introducing the Nymi Band“, https://www.nymi.com/

[16] FIDO Alliance, 2017: „Specifications“, https://fidoalliance.org/

[17] Ochs, C., 2014: „Emerging Trends in Software Development & Implications for IT Security: An Explorative Study“, EC SPRIDE, http://www.ec-spride.tu-darmstadt.de/fileadmin/user_upload/Group_EC_Spride/files/TR_So

[18] Fahl, S.; Harbach, M.; Muders, T.; Smith, M.; Baumgärtner, L.; Freisleben, B.: „Why Eve and Mallory Love Android: An Analysis of Android SSL (In)Security“, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, 2012, S. 50–61

[19] Dienst, J.; Nious, K.: „I-Team: Local Airports Vulnerable to Cyberattacks, Experts Warn“. NBCUniversal Media, 2015, http://www.nbcnewyork.com/news/local/hackers-airport-JFK-Newark-LaGuardia-Cyber-security-320674852.html

[20] ICS-CERT, 2014: „Advisory (ICSA-14-205-01) – Morpho Itemiser 3 Hard-Coded Credential“, https://ics-cert.us-cert.gov/advisories/ICSA-14-205-01

[21] CWE, 2011: „CWE/SANS Top 25 Most Dangerous Software Errors“, http://cwe.mitre.org/top25/

2 Anwendungsfälle und Musterlösungen für Unternehmenssoftware

Gerade im Bereich betrieblicher Anwendungssoftware gibt es einen großen Nachholbedarf, was die Gebrauchstauglichkeit von Schutzmechanismen und Sicherheitsfunktionen angeht. Mangelnde Usability ist in diesem Fall kein Schönheitsfehler. Sie kann, wenn Sicherheitsfunktionen nicht richtig bedient werden und dadurch der Schutz sensibler Daten versagt, drastische Folgen für die Unternehmen haben. Um das Problem einzugrenzen, lohnt es sich, zu schauen, welche Anwendungsfälle konkret betroffen sind und vor allem, welche Musterlösungen geeignet sind, um Abhilfe zu schaffen und Anwenderunternehmen einen effektiven Schutz zu gewährleisten.

Die digitale Transformation unseres beruflichen Alltags schreitet immer weiter voran. Durch die damit einhergehende Vernetzung aller Wirtschaftsbereiche ergeben sich für die Unternehmen neue Geschäftsmodelle und Chancen der Wertschöpfung; allerdings sind auch viele Anpassungen an die neuen Gegebenheiten erforderlich. Beispielsweise fallen täglich riesige Mengen teils sensibler Daten an, die vor unberechtigtem Zugriff geschützt werden müssen. Damit dieser Schutz funktioniert, müssen adäquate Sicherheitskonzepte für Software, Endgeräte und Onlineservices entwickelt werden. Die implementierten Schutzmechanismen und Sicherheitsfunktionen – z. B. Passwortmanagement, Virenscanner oder Kommunikations- und Datenverschlüsselung – dürfen keine Barriere darstellen, die den betrieblichen Anwendern beim Erledigen ihrer eigentlichen Aufgaben im Wege steht. Ansonsten werden die Mechanismen

bewusst oder unbewusst umgangen, wodurch letztendlich das gesamte Sicherheitskonzept zu Fall gebracht werden kann.

Usable Security and Privacy

In Kapitel 1 haben wir einen Ansatz vorgestellt, der seit einigen Jahren unter dem Schlagwort Usable Security and Privacy diskutiert wird. Er integriert Methoden und Werkzeuge des Security Engineerings mit Modellen der Psychologie und Erkenntnissen aus der Mensch-Maschine-Interaktion sowie der Designforschung. Ziel dieses interdisziplinären Ansatzes ist es, Software und interaktive Produkte so zu gestalten, dass sie den Benutzer bei seinen sicherheits- und datenschutzrelevanten Zielen und Vorhaben bestmöglich unterstützen. Der Nutzer soll in die Lage versetzt werden, Sicherheitselemente und deren Notwendigkeit zumindest grundlegend zu verstehen, damit er diese in der dafür vorgesehenen Weise verwenden kann. Bei betrieblichen Anwendungen ist hier ein besonderes Maß an Gebrauchstauglichkeit vonnöten, damit auch Newbies, sporadische Nutzer oder Anwender in Stresssituationen mit den Sicherheitselementen ihres Systems zurechtkommen.

Dem Softwareentwickler kommt in dem Entwicklungsprozess solcher Produkte eine Schlüsselposition zu. Auf der einen Seite implementiert er die Sicherheitsfunktionen, die von den späteren Endanwendern des Systems genutzt werden. Auf der anderen Seite ist er bei der Softwareentwicklung selbst Anwender von Sicherheitskomponenten. Ihm müssen Methoden, Tools, APIs, Bibliotheken und Frameworks zur Verfügung stehen, bei denen komplexe Sicherheits-, Datenschutz- und Usabilityanforderungen in verständlicher Art und Weise integriert sind. Nur dann ist er in der Lage, die entsprechenden nicht funktionalen Anforderungen an ein zu entwickelndes Produkt systematisch und korrekt umzusetzen.

Anwendungsbereiche und Anwendungsfälle für Usable Security

Um die methodischen Grundlagen für das USecureD-Projekt zu schaffen, hat HK Business Solutions zunächst untersucht, welche typischen Anwendungsbereiche und Anwendungsfälle es im Bereich Usable Security gibt. Im Fokus des Projekts steht Unternehmenssoftware; dies umfasst alle Programme, die betriebliche Anwender bei ihrer täglichen Arbeit unterstützen und diesen dabei helfen, Prozesse in ihrem Unternehmen zu optimieren [1]. Hierbei kann zwischen folgenden Arten von Unternehmenssoftware unterschieden werden:

- Betriebswirtschaftliche Anwendungen koordinieren die Geschäftsprozesse in wichtigen betriebswirtschaftlichen Aufgabenbereichen der Unternehmen (z. B. Einkauf, Produktion, Vertrieb und Finanzbuchhaltung) und unterstützen diese abteilungsübergreifend.
- Technische Anwendungen sind ingenieurtechnisch orientierte Anwendungen (z. B. CAD-Programme), die insbesondere in den technischen Bereichen von Unternehmen eingesetzt werden.
- Management- und Informationssysteme (z. B. Data Warehouse) dienen der Erfassung, Verarbeitung und Analyse von Information bzw. Daten und unterstützen die Lenkungsentscheidungen von Unternehmen.
- Anwendungen zur Unterstützung der betrieblichen Abläufe (z. B. Office-Software, E-Mail-Programme oder Groupware) helfen bei der bereichsübergreifenden Abstimmung von Arbeitsabläufen bzw. bei der Zusammenarbeit in Gruppen.

Den Kern der Unternehmenssoftware bilden nach dem Verständnis vieler Nutzer die betriebswirtschaftlichen Anwendungen, die oft auch verallgemeinernd als

Unternehmenssoftware bezeichnet werden. Diese betriebswirtschaftlichen Anwendungen werden von vielen Herstellern zu Produktsuites zusammengefasst und als Enterprise-Resource-Planning-(ERP-)Systeme vermarktet. ERP-Standardprodukte sind bei mittelständischen Anwenderunternehmen sehr beliebt, zudem werden sie oft von einer Nutzergruppe verwendet, die hinsichtlich Vorerfahrung, Securityexpertise und Usabilityanforderungen sehr heterogen ist. Daher bildeten diese Produkte den Schwerpunkt der Untersuchung, um mit den späteren Projektergebnissen einen möglichst großen Nutzen zu erzielen. Von den Management- und Informationssystemen bzw. bei den Anwendungen zur Unterstützung der betrieblichen Abläufe haben wir bei der Untersuchung jeweils bestimmte Teilbereiche betrachtet. Analysiert wurden Anwendungsbereiche, die einen starken Securitybezug haben bzw. die für eine Vielzahl von Unternehmen relevant sind (z. B. Projektverwaltung, Controlling).

Im nächsten Schritt hat HK Business Solutions acht ERP-Produkte analysiert und hierbei insbesondere typische Anwendungsfälle identifiziert. Als Untersuchungsgegenstand haben wir die Produkte der Marktführer SAP Business One, Microsoft Dynamics NAV, Sage und Oracle sowie zwei kleinere kommerzielle Produkte für die Zielgruppe KMU und zwei Open-Source-Systeme ausgewählt. Unterstützend wurden verschiedene Fachbücher und Studien zum Funktionsumfang bzw. Einsatz von ERP-Systemen analysiert. Anschließend haben wir die gesammelten Anwendungsfälle in mehreren Expertenworkshops aufbereitet und in Form von Use-Case-Diagrammen [2] dokumentiert.

Use Cases bzw. Use-Case-Diagramme beschreiben die Interaktion der Nutzer mit einem System – abstrahiert von konkreten

technischen Lösungen und aus Sicht der jeweiligen Akteure. Daher sind Use Cases sehr gut dazu geeignet, die gesamte Konzeption und Entwicklung eines Systems an den Bedürfnissen der späteren Nutzer auszurichten. **Abbildung 2.1** zeigt eines der zweiundzwanzig im Projekt erstellten Use-Case-Diagramme. Als Bezeichnung für die Akteure in den Use-Case-Diagrammen wurden typische Rollenbezeichnungen verwendet, wie sie in kleinen und mittleren Unternehmen anzutreffen sind, in diesem Fall z. B. Einkäufer, Vertriebsleiter und Vertriebsmitarbeiter.

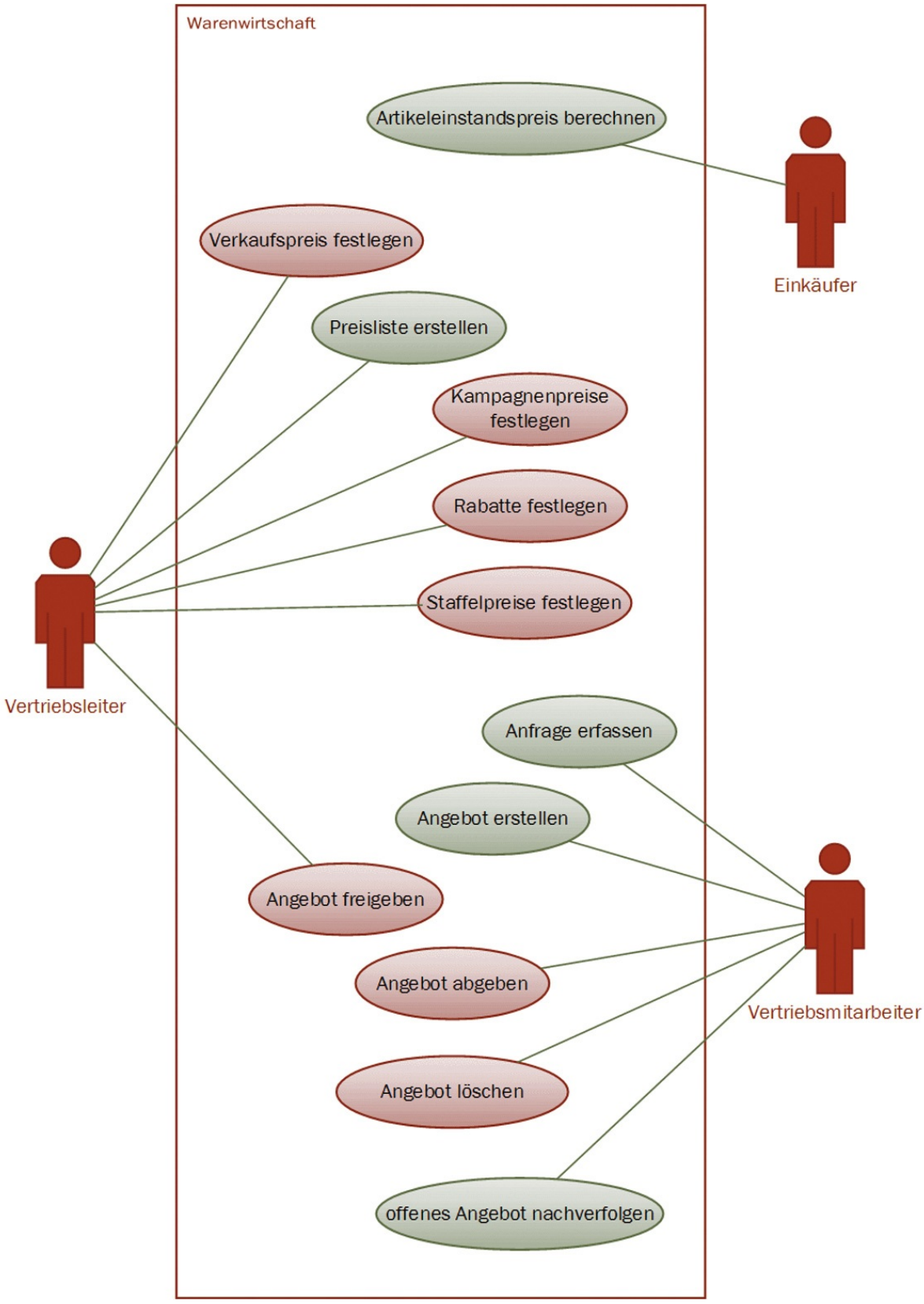


Abbildung 2.1: Use-Case-Diagramm für den Bereich „Preiskalkulation und Angebote“

Anhand der Use-Case-Diagramme haben wir sämtliche Anwendungsfälle identifiziert, bei denen ein erhöhter

Sicherheitsbedarf besteht und die insofern einen besonderen Bezug zum Thema Usable Security haben. Dies waren insbesondere

- Anwendungsfälle, bei denen Fakturaänderungen stattfinden (z. B. wertmäßige Änderung eines Belegs)
- Anwendungsfälle, bei denen Lagerbewegungen stattfinden (z. B. mengenmäßige Änderung von Waren)
- Anwendungsfälle, bei denen das betrachtete System mit einem Drittsystem kommuniziert (z. B. Import oder Export von Daten)

Sechs dieser insgesamt 321 Anwendungsfälle haben wir exemplarisch ausgearbeitet. Bei der Auswahl dieser Beispiele haben wir darauf geachtet, dass diese aus unterschiedlichen Bereichen stammen und die Sicht verschiedener Akteure (Endanwender, Systemadministrator und Entwickler) auf das System widerspiegeln. Tabelle 2.1 zeigt den Use Case „Benutzerberechtigung ändern“ in Form eines Auszugs.

In der Literatur gibt es eine Vielzahl von Schablonen, Vorlagen und Empfehlungen zur Ausarbeitung von Use Cases in unterschiedlichen Detaillierungsgraden (vgl. [3], [4], [5]). Aufbauend auf diesen Arbeiten haben wir ein Beschreibungstemplate für die Dokumentation der Anwendungsfälle mit Usable-Security-Bezug erstellt. Zusätzlich zu bekannten Use-Case-Attributen wie ID, Beschreibung, Akteur usw., mit denen grundlegende Informationen zum Use Case erfasst werden, haben wir fünf neue Attribute (z. B. Sicherheitsrisiken und Gefährdung) eingeführt, die dem Thema „Usable Security“ Rechnung tragen.

Use Case „Benutzerberechtigung ändern“

Beschreibung	Dieser Use Case beschreibt, wie ein Systemadministrator im Administrationsbereich die Berechtigungen eines Benutzers ändert
Akteur	Systemadministrator
System	Administrationsbereich
Nutzungskontext	Der Anwender ist am Administrationsbereich angemeldet und hat die Benutzerverwaltung geöffnet
Ziel	Anwender: Benutzerberechtigungen können schnell und komfortabel administriert werden Geschäftsleitung: Geschäftsdaten sind vor unberechtigtem Zugriff geschützt Zugriff haben lediglich Personen, die diesen für die Erledigung der ihnen übertragenen Aufgaben benötigen und denen der Zugriff ordnungsgemäß übertragen worden ist
Vorbedingung	Dem Anwender ist bekannt, ob und in welcher Weise die Benutzerberechtigung eines Benutzers/einer Benutzergruppe geändert werden soll
Standardablauf	1. Systemadministrator öffnet Benutzerkonsole 2. System zeigt Benutzerkonsole an 3. Systemadministrator sucht Benutzer, dessen Benutzerberechtigung geändert werden soll 4. System zeigt Benutzermaske an 5. Systemadministrator nimmt Rechteänderung vor (Verzeichnisse bzw. Dateien, Lesen/Schreiben/Löschen ...) 6. System prüft Plausibilität der Rechtevergabe und ändert Benutzerberechtigung
Nachbedingung/Ergebnis	Die Benutzerberechtigungen sind geändert. Optional kann die Berechtigungsliste des Benutzers/der Benutzergruppe ausgedruckt werden, und es können die geänderten Berechtigungen anhand identischer Einstellungen eines Standardnutzers überprüft werden
Sicherheitsrisiken	Sicherheitsrelevante Fehlkonfiguration Verlust der Vertraulichkeit sensibler Daten Fehlerhafte Autorisierung auf Anwendungsebene

Gefährdung	Unberechtigte Personen haben Zugriff auf Geschäftsdaten
Qualitätsmerkmale	Gebrauchstauglichkeit: Fehlertoleranz, Erwartungskonformität, Steuerbarkeit, Fehlersicherheit Sicherheit: Vertraulichkeit, Integrität, Nachweisbarkeit, Verantwortlichkeit, Authentizität, Zugriffskontrolle, Nichtabstreitbarkeit Nutzungsqualität: Effektivität, Effizienz, Minderung des wirtschaftlichen Risikos

Tabelle 2.1: Beispiel für einen Use Case („Benutzerberechtigung ändern“, Auszug)

Musterlösungen für Usable Security

Bei der Planung und Implementierung von Sicherheitsfunktionen für die oben genannten Use Cases stoßen Softwarearchitekten und -entwickler von betrieblicher Anwendungssoftware auf wiederkehrende Probleme. Beim Lösen solcher Problemstellungen profitieren sie von gut dokumentierten, bewährten und wiederverwendbaren Musterlösungen, so genannten Patterns. Der von Christopher Alexander [6] eingeführte Begriff stammt ursprünglich aus dem Bereich der Architektur und wurde mit der Zeit auf den Softwareentwicklungsprozess übertragen ([7], [8]). Patterns sind seitdem ein fester Bestandteil der Softwarebranche geworden und transferieren Wissen über erfolgserprobte Problemlösungen bei Design- und Implementierungsfragen. Dies zeigt auch eine Vielzahl an Literatur und Datenbanken, in denen Patterns verschiedener Softwareentwicklungsbereiche gesammelt werden. Das Spektrum an Themenbereichen ist dabei sehr umfangreich und reicht von grundsätzlichen, strukturellen Problemstellungen der Softwarearchitektur, bis hin zu speziellen Themen wie Security [9], User Interface Design [10] und Usability [11].

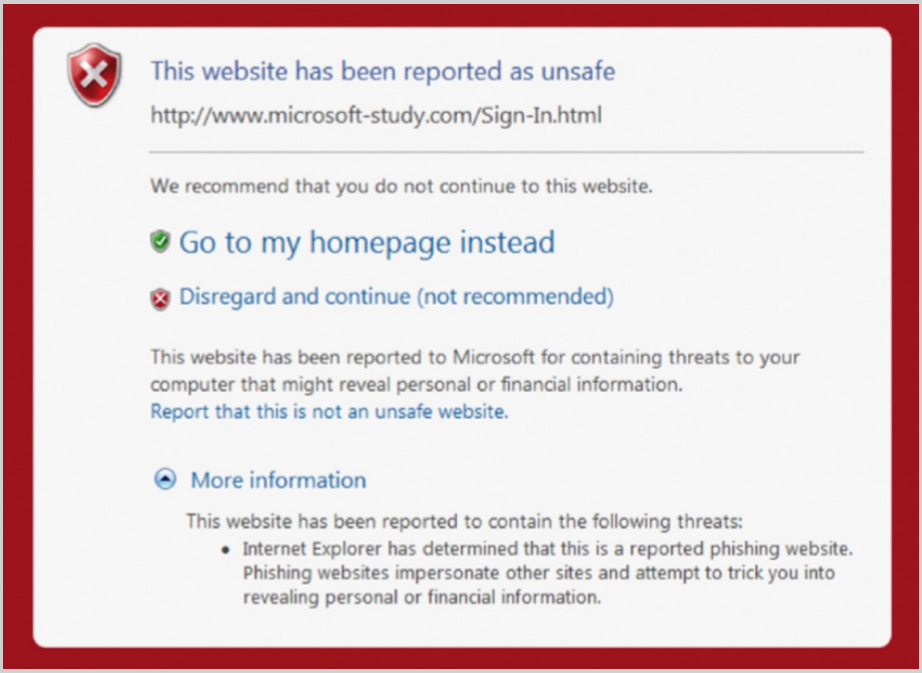
Für den gesonderten Nutzungskontext der Usable Security gab es bisher keine umfassende Sammlung bzw. Datenbank solcher

Musterlösungen, insbesondere nicht in deutscher Sprache. Aus diesem Grund hat die Technische Hochschule Köln im Rahmen des USecureD-Projekts eine umfassende Literaturrecherche durchgeführt, durch die eine Sammlung von bisher 47 englischsprachigen Usable Security Patterns für Probleme unterschiedlichster Themenbereiche zusammengestellt werden konnte. Diese Themenbereiche umfassen unter anderem:

- Authentifizierung, Autorisierung und Schlüsselmanagement
- Signatur und Verschlüsselung von E-Mails bzw. Kommunikationskanälen
- Sicheres Löschen von Daten und Anlegen von Sicherungskopien
- Gebrauchstaugliche Bedienelemente für Sicherheitsfunktionen
- Gestaltung von Hinweisen, Warnungen und Sicherheitsstatus

Die Patterns verschiedener Autoren ([12], [13], [14]) wurden anschließend in ein selbstentwickeltes einheitliches Format (Patterntemplate) übertragen und von HK Business Solutions ins Deutsche übertragen. Das Template enthält neben den üblichen Attributen „Kontext“, „Problem“ und „Lösung“ viele weitere Attribute wie beispielsweise Implementierungshinweise und resultierende Konsequenzen bei der Anwendung des Patterns. Beispielhaft an dem Pattern „Empfehlungen bereitstellen“ von Egelman [12] zeigt Tabelle 2.2 die Umsetzung des Patterntemplates.

Durch die umfassende Analyse der Technischen Hochschule Köln wurden zudem sinnvolle Verknüpfungen der Patterns untereinander hergestellt, wodurch die Sammlung zu einer Mustersprache bzw. Pattern Language weiterentwickelt werden konnte.

Name	Empfehlungen bereitstellen
Quellen	(Egelman 2009)
Synonyme	keine
Kontext	Wird eine Gefahr erkannt, sollte dem Nutzer eine klare Empfehlung zum sicheren Fortfahren präsentiert werden und eine Liste mit anderen möglichen Aktionen angezeigt werden.
Problem	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.
Lösung	Warnmitteilungen müssen dem Nutzer eine weitere Vorgehensweise vorschlagen und eine Anleitung geben, wie er diese umsetzen kann.
Beispiel	 Abb. 2.3: Warnmitteilung (Beispiel)
Implementierung	Die empfohlene Aktion sollte markanter als alle anderen Optionen sein. 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.
Konsequenzen	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).
Abhängigkeiten	keine
Beziehungen	[Dialoge mit Handlungsempfehlungen] [Empfohlene Optionen] [Bemerkbare kontextabhängige Hinweise]
Prinzipien	[Sichtbarkeit]
Tags	Fehlersicherheit, Erlernbarkeit

Tabelle 2.2: Beispiel für ein Usable-Security-Pattern („Empfehlungen bereitstellen“, Auszug)

Um die Arbeit mit der Pattern Language zu vereinfachen, wurden die Verknüpfungen in einer interaktiven Grafik (Dependency Wheel) visualisiert. Wie in Abbildung 2.2 zu sehen ist, ermöglicht dies dem Nutzer ein schnelles Erfassen von Beziehungen und somit ein effizientes Arbeiten mit der Sammlung. Sowohl die interaktive Grafik als auch die englisch- und deutschsprachigen Patterns stehen auf der USecureD-Plattform [15] zur Anwendung bereit.

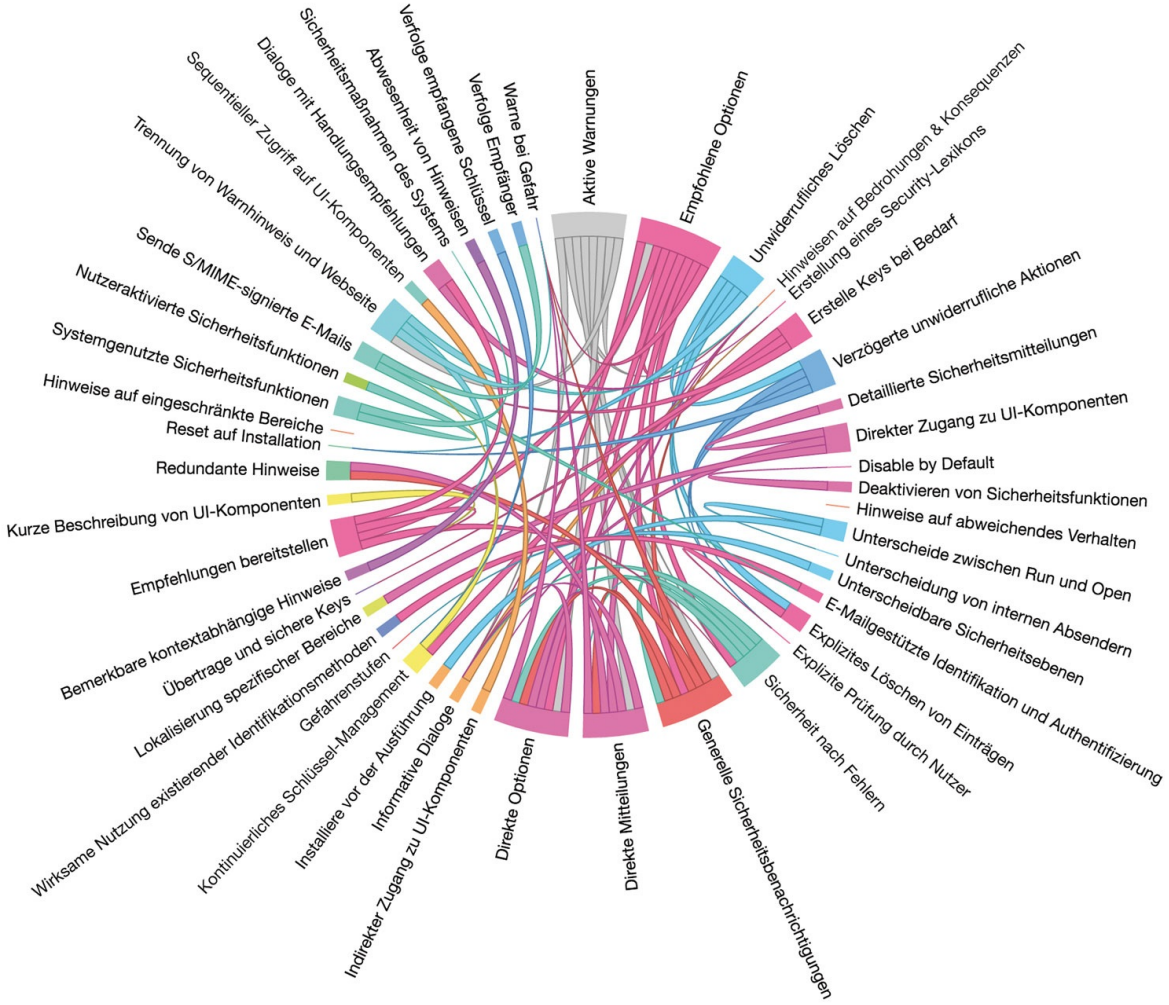


Abbildung 2.2: Abhängigkeiten der Musterlösungen

Links & Literatur

[1] http://www.datev.de/portal/ShowPage.do?pid=dpi&nid=107145

[2] Booch, Grady; Rumbaugh, James; Jacobson, Ivar: „The Unified Modeling Language User Guide“, Addison-Wesley, Boston, 1999

[3] Jacobson, Ivar; Christerson, Magnus; Jonsson, Patrik; Övergaard, Gunnar: „Object Oriented Software Engineering: A Use Case Driven Approach“, Addison-Wesley, Boston, 1992

[4] Cockburn, Alistair: „Writing Effective Use Cases“, Addison-Wesley, Boston, 2000

[5] http://www.re-

wissen.de/opencms/Wissen/Techniken/Guidelines_zur_Erstellung_

[6] Alexander, Christopher; Ishikawa, Sara; Silverstein, Murray; Jacobson, Max; Fiksdahl-King, Ingrid; Angel, Shlomo: A Pattern Language: „Towns, Buildings, Construction“, Oxford University Press, New York, 1977

[7] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: „Design Patterns: Elements of Reusable Object-Oriented Software“, Addison-Wesley, Boston, 1995

[8] Beck, Kent: „Smalltalk Best Practice Patterns“, Prentice Hall, Upper Saddle River, 1996

[9] Steel, Christopher; Nagappan, Ramesh; Lai, Ray: „Core Security Patterns“, Prentice Hall, Upper Saddle River, 2005

[10] http://developer.yahoo.com/ypatterns

[11] http://www.usabilitypatterns.info/catalog/catalog.html

[12] Egelman, Serge: „Trust Me: Design Patterns for Constructing Trustworthy Trust Indicators“, Dissertation, Carnegie Mellon University, Pittsburgh, 2009

[13] Garfinkel, Simson L.: „Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable“, Dissertation, Massachusetts Institute of Technology, Cambridge, 2005

[14] 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, Springer, London,
S. 283–94

[15] https://das.th-koeln.de/usecured

[16] Birolini, Alessandro: „Zuverlässigkeit von Geräten und

Systemen“, Springer, Berlin, 1997

3 Entwicklungsrichtlinien für Produkte mit dem Qualitätsmerkmal „Usable Security“

Wie kann man auf möglichst strukturierte Weise Anwendungssoftware mit dem Qualitätsmerkmal Usable Security entwickeln? Und welche Werkzeuge sind hierfür geeignet? Diesen Fragen geht das dritte Kapitel nach und schlägt Prinzipien und Richtlinien als adäquate Gestaltungswerkzeuge vor. Durch die Verwendung dieser Werkzeuge ist es auch Entwicklern in kleineren Softwareunternehmen möglich, bereits in frühen Entwicklungsphasen eines Produkts die Weichen gezielt in Richtung Usable Security and Privacy zu stellen.

Unter dem Schlagwort Usable Security and Privacy wird seit einigen Jahren ein Ansatz diskutiert, der Methoden und Werkzeuge des Security Engineerings mit Modellen der Psychologie und Erkenntnissen aus der Mensch-Maschine-Interaktion und der Designforschung integriert. Ziel dabei ist, Software und interaktive Produkte so zu gestalten, dass sie den Benutzer bei seinen sicherheits- und datenschutzrelevanten Zielen und Vorhaben bestmöglich unterstützen.

Frühe Qualitätssicherung mit passenden Entwurfswerkzeugen

Um auf möglichst effiziente Weise Produkte mit einem hohen Maß an Usable Security and Privacy zu entwickeln, ist eine Unterstützung in frühen Prozessphasen notwendig. Usabilitymängel oder Schwachstellen, die Security und Privacy betreffen, werden dann bestenfalls von Anfang an vermieden und müssen nicht in späteren aufwendigen Prozessphasen identifiziert und behoben werden. In diesem Kapitel stellen wir

daher weitere Werkzeuge vor, die wir genau zu diesem Zweck im Rahmen des USecureD-Projekts entwickelt haben. Auch diese Werkzeuge – Prinzipien und Richtlinien für Usable Security – stehen auf der genannten USecureD-Plattform kostenlos zur Verfügung.

Unter Prinzipien verstehen wir allgemeine Grundsätze, denen die Entwicklung bzw. Implementierung eines Systems – z. B. einer geschäftlichen Anwendung – folgen sollte. Die Prinzipien für Usable Security haben also einen ähnlichen Zweck wie vergleichbare Prinzipien, die vielen Lesern bereits vertraut sind, etwa die Prinzipien agiler Softwareentwicklung oder der objektorientierten Programmierung. Wie die Prinzipien konkret umgesetzt werden, ist in den Richtlinien für Usable Security beschrieben. Die Richtlinien bewegen sich also auf einer etwas niedrigeren Abstraktionsebene. Sie sind in einer ähnlichen Form beschrieben wie bekannte Design-Guidelines oder Entwicklungsrichtlinien, die von vielen Herstellern und Entwicklerteams als konstruktive Qualitätssicherungsmaßnahme eingesetzt werden. Ein Beispiel veranschaulicht das Zusammenspiel der Usable-Security-Prinzipien und -Richtlinien: Das Prinzip der Sichtbarkeit (vgl. Tabelle 3.1) kann bei der Entwicklung eines Produkts umgesetzt werden, indem die Richtlinien „Sichtbarer Sicherheitszustand“, „Sichtbare Sicherheitsfunktionen“, „Deutliche Kennzeichnung sensibler Daten“ und „Hervorhebung kritischer Inhalte“ angewendet werden. In den entsprechenden Richtlinien ist konkret beschrieben, was der Entwickler z. B. bei der Umsetzung der Benutzeroberfläche beachten sollte, damit alle Sicherheitskomponenten und -mechanismen für den Nutzer gut sichtbar sind.

Prinzipien für Usable Security

Prinzipien basieren auf früheren Erfahrungen, Designs oder wissenschaftlichen Erkenntnissen und sind als abstrakte Entwurfswerkzeuge für viele Anwendungsbereiche gültig. Zum einen dienen Prinzipien der Usable-Security-Forschung als Einstiegsmedium, um einen ersten Überblick über die Thematik Usable Security zu bekommen. Des Weiteren können sie während der Planungsphase von Softwareprojekten eingesetzt werden, um Softwarearchitekten und -entwickler dabei zu unterstützen, wichtige Weichenstellungen für die gesamte Systementwicklung von Anfang an richtig vorzunehmen. Ebenso bieten sie Käufern von Softwareprodukten die Möglichkeit, Anforderungen und Kriterien für ihre konkreten Anwendungsfälle präziser zu definieren.

Um unsere Usable-Security-Prinzipien zu erarbeiten, haben wir zunächst den derzeitigen Stand der Forschung in diesem Bereich erfasst. Im Rahmen einer umfassenden Literaturrecherche haben wir existierende Prinzipien verschiedener Autoren gesammelt und analysiert. Maßgeblich sind in unsere Arbeit die Ergebnisse von Whitten und Tygar [1], Garfinkel [2], Yee [3], Furnell et al. [4], Chiasson et. al [5] und Hof [6] eingeflossen.

Anschließend haben wir die Prinzipiensammlung konsolidiert und eine übersichtliche Struktur für den Prinzipienkatalog geschaffen. Damit die insgesamt 23 Usable-Security-Prinzipien unterschiedlicher Autoren in einem einheitlichen, gut leserlichen Format dokumentiert werden konnten, haben wir eine Beschreibungsvorlage entwickelt. Hierzu haben wir untersucht, welche wiederkehrenden Merkmale in den von unterschiedlichen Autoren beschriebenen Prinzipien vorkommen. Bei der überwiegenden Mehrzahl der Autoren sind das die Attribute „Intention“ und „Motivation“. Sie beschreiben, welche Absicht ein Prinzip hat bzw. welchen Zweck es erfüllen

soll und welche Umstände die Verwendung des Prinzips motivieren können. Die extrahierten Merkmale haben wir durch weitere Attribute ergänzt: Synonyme, unter denen das Prinzip noch bekannt ist, Beispiele, bei denen das Prinzip Anwendung findet, Schlagworte zur besseren Durchsuchbarkeit des Prinzipienkatalogs, Angaben zu den genutzten Quellen und Verknüpfungen zu Richtlinien, die dieses Prinzip umsetzen. Um die Nutzbarkeit der Prinzipien für mittelständische Unternehmen zu verbessern, wurden sie zudem von den englischsprachigen Quellen ins Deutsche übersetzt.

Prinzip „Sichtbarkeit“	
Name	Sichtbarkeit
Quellen	Yee [3]
Synonyme	Sichtbar
Intention	Das System sollte einen klaren Hinweis geben, ob Sicherheitssysteme aktiv sind.
Motivation	Eine angemessene Verwendung von Statusanzeigen und Warnungen wird dem Nutzer helfen, sich zu erinnern, falls er vergessen hat, geeignete Sicherheitsvorkehrungen zu treffen.
Beispiele	Keine
Richtlinien	Sichtbarer Sicherheitszustand Sichtbare Sicherheitsfunktionen Deutliche Kennzeichnung sensibler Daten Hervorhebung kritischer Inhalte
Tags	Selbstbeschreibungsfähigkeit, Zugreifbarkeit

Tabelle 3.1: Beispiel für ein Prinzip („Sichtbarkeit“)

Richtlinien für Usable Security

Entwicklungsrichtlinien (Design-Guidelines) sind wichtig, um bereits bei der Entwicklung von Systemen möglichst viele Ursachen für spätere Schwachstellen zu eliminieren [7]. In der Softwareentwicklung tragen Richtlinien zum einen zur Gewährleistung einer besseren Qualität bei, zum anderen verringern sie die Komplexität der Entwicklungsprojekte, in denen sie angewendet werden.

Unser Ziel war es, eine Sammlung möglichst praxistauglicher Richtlinien für den Bereich Usable Security aufzubauen. Nach Möglichkeit sollten dies bewährte Richtlinien sein, die auch von Entwicklern in kleineren Unternehmen genutzt werden können, um auf systematische Weise Anwendungssoftware mit dem Qualitätsmerkmal Usable Security zu entwickeln. Zugleich wollten wir den Entwicklern eine nachvollziehbare Entscheidungshilfe an die Hand geben und sie bei der Verwendung einzelner Richtlinien bzw. bei der sinnvollen Kombination mehrerer Richtlinien unterstützen. Hierzu wollten wir die Richtlinien mit zusätzlichen Ressourcen und weiterführenden Informationen anreichern, z. B. mit Umsetzungsbeispielen, Hinweisen zur Wirkungsweise und zu vertiefender Literatur.

Im ersten Schritt haben wir allgemeine Informationen zu Richtlinien recherchiert, insbesondere zu Usability Guidelines [8], [9], [10]. Da es bis dahin keine umfassende Sammlung von Richtlinien für den Bereich Usable Security and Privacy gab, haben wir als Nächstes eine eingehende Literatur- und Internetrecherche durchgeführt. Diese Recherche haben wir zum einen genutzt, um Richtlinien für den Aufbau unserer Sammlung zu identifizieren, zum anderen, um – analog zur Vorgehensweise bei den Prinzipien – eine Beschreibungsvorlage zu erstellen, mit

der alle Richtlinien einheitlich und in einem gut lesbaren Format dokumentiert werden konnten. Die Richtlinien wurden also sowohl inhaltlich als auch hinsichtlich ihres Aufbaus und der Strukturierung eingehend untersucht. Die Autoren beziehungsweise Herausgeber der Richtlinien sind Wissenschaftler, (Berufs-)Verbände, Organisationen (Regierungsbehörden, NGOs) und Software- bzw. Systemhersteller. Alle Quellen, die genutzt wurden, sind auf der USecureD-Plattform aufgeführt. Thematisch können die untersuchten Richtlinien unterteilt werden in:

- Usable Security Guidelines: Analysiert wurden kleinere Usable-Security-Guideline-Sets und Arbeiten, die einzelne Guidelines vorstellen beziehungsweise Hinweise zu Guidelines enthalten
- Allgemeine Richtlinien für die Gestaltung von Benutzeroberflächen
- Usability-Guidelines: Betrachtet wurden umfassende Usability-Guideline-Sammlungen, Usability-Heuristiken, Guidelines für Texte in Masken und Meldungen sowie Arbeiten zur Fehlersicherheit und Fehlertoleranz
- Accessibility Guidelines
- User Experience Guidelines
- Trust Design Guidelines
- Privacy Guidelines
- Guidelines für die Erstellung von (Business-)Websites bzw. Onlineshops
- Guidelines für die Herstellung von Apps
- Security Guidelines: Untersucht wurden Security-Guideline-Sammlungen sowie Guidelines zum Umgang mit Passwörtern

Nach dem Zusammentragen der Richtlinien haben wir die Richtlinien identifiziert, die den stärksten Bezug zum Thema Usable Security and Privacy haben (z. B. Usability Guidelines, die für Sicherheitskomponenten anwendbar sind, oder Security Guidelines, die sich durch ein hohes Maß an Usability empfehlen). Diese Richtlinien haben wir strukturiert und mit Schlagworten versehen, außerdem haben wir Themencluster gebildet. Bei der Aufbereitung bildete sich der in Tabelle 3.2 vorgestellte Aufbau heraus, den wir als Beschreibungsvorlage für die anschließende Dokumentation der Richtlinien genutzt haben.

Übersicht – Richtlinien zur Umsetzung des Prinzips „Sichtbarkeit“			
Name	Sichtbarer Sicherheitszustand	Sichtbare Sicherheitsfunktionen	Deutliche Kennzeichnung sensibler Daten
Quellen	Nurse et al. [11]	Nurse et al. [11]	Smith/Mosier [12]
Synonyme	Allow for visibility of system state	Convey available features, Make security functionality visible and accessible	Keine
Richtlinie	Mache den aktuellen Sicherheitszustand des Systems für den Nutzer sichtbar.	Mache Sicherheitsfunktionen für den Nutzer sichtbar und zugänglich.	Sorge dafür, dass der Nutzer deutlich erkennen kann, welche Daten vertraulich zu behandeln sind und welche Daten er nicht bearbeiten darf.
Kontext	Ein System sollte den Nutzer jederzeit angemessen darüber informieren, was gerade passiert. Die Anzeige des aktuellen Sicherheitszustands kann in vielen Fällen genutzt werden, um dem Nutzer ein passives Feedback über die Informationssicherheit zu geben. Die Anzeige des aktuellen Sicherheitszustands	Ebenso wie die übrigen Funktionen und Elemente einer Anwendung sollten auch Sicherheitsfunktionen und -elemente für den Nutzer sichtbar und leicht zugänglich sein. Das Auslagern von Sicherheitsfunktionalität in	Wenn dem Nutzer Daten angezeigt werden, die als „vertraulich“, „intern“ oder „geheim“ eingestuft sind, sollte er einen prominenten Hinweis auf diese

	trägt zum Aufbau von Vertrauen bei und gilt daher als Kriterium für erfolgreiche Mensch-Maschine-Interaktion in Sicherheitsanwendungen. Die Anzeige des aktuellen Zustands darf allerdings nicht aufdringlich sein, z. B. darf der Nutzer nicht die gesamte Zeit mit Sicherheitswarnungen konfrontiert werden.	getrennte bzw. versteckte Bereiche einer Anwendung oder in Bereiche für Fortgeschrittene erschwert hingegen dem Benutzer die Erledigung seiner Aufgabe und beeinträchtigt dadurch die Usability.	Sicherheitsklassifizierung erhalten. Ebenso sollte für den Nutzer klar ersichtlich sein, welche Daten er wegen fehlender Berechtigung nicht bearbeiten darf.
Beispiele	Beispiele für die Anzeige des aktuellen Sicherheitszustands sind Begriffe wie „Passwort-Schutz“ oder „verschlüsselt“ bei passwortgeschützten bzw. verschlüsselten Dokumenten, aktive Symbole, wenn in einem System momentan Sicherheitsfunktionen ausgeführt werden, oder Vorhängeschlösser, um in Browsern die Verwendung von Secure Sockets Layer (SSL) bzw. Transport Layer Security (TLS) anzuzeigen.	Keine	Keine
Verwandte Richtlinien	Sichtbare Sicherheitsfunktionen	Sichtbarer Sicherheitszustand	Fehlern vorbeugen, Hervorhebung kritischer Inhalte
Kategorie	Usable Security	Usable Security	Usable Security
Tags	Sichtbarer Sicherheitszustand, sichtbare Sicherheitsfunktionen, Selbstbeschreibungsfähigkeit, Vertrauen	Sichtbare Sicherheitsfunktionen, sichtbarer Sicherheitszustand, Selbstbeschreibungsfähigkeit, Vertrauen	Deutliche Kennzeichnung sensibler Daten, Fehlersicherheit, Vertraulichkeit, Zugangs- und Zugriffskontrolle, Selbstbeschreibungsfähigkeit

Tabelle 3.2: Richtlinien zur Umsetzung des Prinzips „Sichtbarkeit“

Bei der Erstellung der Sammlung wurden in vielen Fällen Guidelines verschiedener Autoren zusammengeführt. Hierdurch wollten wir sicherstellen, dass eine gut handhabbare Sammlung

entsteht, über die sich Softwareingenieure und -entwickler schnell einen Überblick verschaffen können. Nach Fertigstellung der Sammlung (Stand September 2016: 31 Richtlinien) haben wir sämtliche Richtlinien und Prinzipien gesichtet sowie analysiert, welche Richtlinien und Prinzipien thematisch miteinander in Verbindung stehen. Um solche Verknüpfungen zwischen Richtlinien und Prinzipien zu dokumentieren, haben wir das Attribut „Richtlinien“ genutzt, das hierfür in der Beschreibungsvorlage der Prinzipien vorgesehen ist. Wählt der Besucher der USecureD-Plattform heute die Prinzipien als Einstiegspunkt, so steht ihm also eine durchgängige Werkzeugkette zur Verfügung, die ihm bei jedem der beschriebenen Prinzipien passende Richtlinien empfiehlt. Die Beziehungen aller Prinzipien mit den dazugehörigen Richtlinien sind in **Abbildung 3.1** visualisiert.

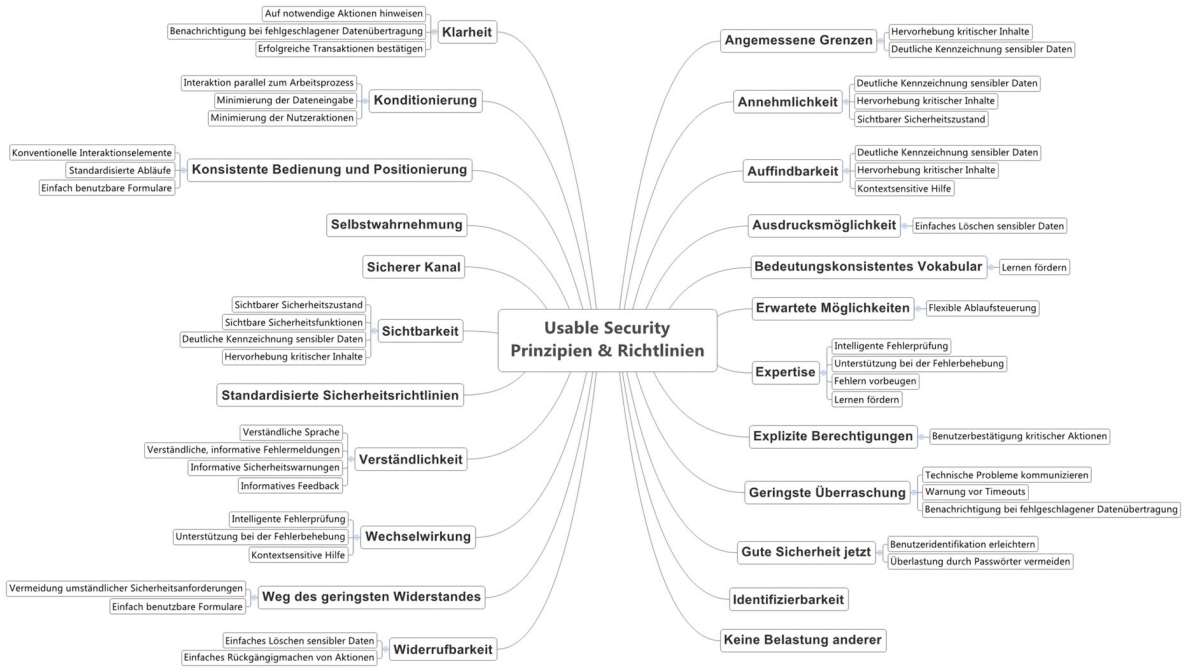


Abbildung 3.1: Übersicht und Beziehungen aller Usable-Security-Prinzipien und -Richtlinien

Die Richtlinien wurden von uns – ebenso wie die Prinzipien – so konzipiert, dass sie als Arbeitsgrundlage für Softwareingenieure

und -entwickler dienen können, zugleich aber auch als Kommunikationsbasis im Entwicklungsteam. Unser besonderes Augenmerk bei der Ausarbeitung dieser Werkzeuge lag auf der Dialog- und Interaktionsgestaltung betrieblicher Anwendungssysteme. Prinzipiell sind die Richtlinien und Prinzipien aber auch auf andere Anwendungsdomänen übertragbar. Sie stehen heute in Form einer konsolidierten, frei zugänglichen Sammlung auf der USecureD-Plattform zur Verfügung. Hierdurch haben interessierte Entwickler und Unternehmen die Möglichkeit, je nach Qualitätszielen oder technischen Aspekten, die bei ihnen im Vordergrund stehen, eine Auswahl aus den angebotenen Werkzeugen zu treffen und sie bei Bedarf individuell anzupassen.

Links & Literatur

[1] Whitten, Alma; Tygar, J. D.: „Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0“, in: Proceedings of the 8th Conference on USENIX Security Symposium, USENIX Association, Berkeley, 1999

[2] Garfinkel, Simson L.: „Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable“, Dissertation, Massachusetts Institute of Technology, Cambridge, 2005

[3] Yee, Ka-Ping: „User Interaction Design for Secure Systems“, in: ICICS ‘02 Proceedings of the 4th International Conference on Information and Communications Security, S. 278–290, Springer, London, 2002

[4] Furnell, Steven; Katsabas, Dimitris; Dowland, Paul; Reid, Fraser: „A Practical Usability Evaluation of Security Features in End-User Applications“, in: New Approaches for Security, Privacy and Trust in Complex Environments: Proceedings of the IFIP TC-

11 22nd International Information Security Conference, S. 205–216, Springer US, New York, 2007

[5] https://cups.cs.cmu.edu/soups/2007/workshop/Design_Guidelines.p

[6] Hof, Hans-Joachim: „User-Centric IT Security: How to Design Usable Security Mechanisms“, in: CENTRIC 2012 – The Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, S. 7–12, IAIRA, Wilmington, 2012

[7] Birolini, Alessandro: „Zuverlässigkeit von Geräten und Systemen“, Springer, Berlin, 1997

[8] http://www.usabilityblog.de/2009/08/usability-guidelines-teil-1-definition-abgrenzung/

[9] http://webstyleguide.com/wsg3/

[10] https://www.nngroup.com/articles/sixty-guidelines-from-1986-revisited/

[11] Nurse, Jason R. C.; Creese, Sadie; Goldsmith, Michael; Lamberts, Koen: „Guidelines for Usable Cybersecurity: Past and Present“, in: 2011 Third International Workshop on Cyberspace Safety and Security, S. 21–26, IEEE, Piscataway, 2011

[12] http://hcibib.org/sam/

4 Metriken, Evaluationswerkzeuge und Testplattform

Auf welche Weise kann man überprüfen, ob bei einer Anwendungssoftware das Qualitätsmerkmal Usable Security erfolgreich umgesetzt wurde, sprich: ob die Sicherheitsfunktionen dieser Software benutzerfreundlich gestaltet sind? Dieser Fragestellung geht das vierte Kapitel nach. Er schlägt ein Set von Metriken und einfachen Werkzeugen vor, mit denen auch kleinere Hersteller die Qualität ihrer Produkte evaluieren können – ohne tiefere Spezial- und Expertenkenntnisse und in der Regel ohne Laborumgebung.

Softwaresysteme und interaktive Produkte müssen so gestaltet sein, dass sie den Benutzer bei seinen sicherheits- und datenschutzrelevanten Zielen und Vorhaben möglichst gut unterstützen. Insbesondere müssen diese Produkte mit Sicherheitselementen ausgestattet sein, die für möglichst viele Nutzergruppen verständlich und benutzbar sind. Damit das gelingt, bedarf es eines interdisziplinären Ansatzes, der Methoden und Werkzeuge des Security Engineerings mit Modellen der Psychologie und Erkenntnissen aus der Mensch-Maschine-Interaktion sowie der Designforschung zusammenführt.

Evaluierung der Benutzerfreundlichkeit von Sicherheitsfunktionen

Im USecureD-Projekt wurde daher ein Evaluierungskonzept entwickelt, das Softwarearchitekten und -entwickler in die Lage versetzt, im eigenen Kontext selbstständig Evaluierungen der implementierten Konzepte und Softwareanwendungen durchzuführen (vgl. Tabelle 4.1). Ziel war es, eine

Evaluierungsmethodik zu schaffen, die aussagekräftige Resultate in Bezug auf das Qualitätsmerkmal Usable Security liefert, aber trotzdem möglichst „leichtgewichtig“ ist. Leichtgewichtig bedeutet nach unserem Verständnis, dass der Methodeneinsatz je nach Evaluierungsfall an die Erfordernisse des konkreten Projekts angepasst werden kann, um so einen optimalen Einsatz der personellen und zeitlichen Ressourcen zu ermöglichen. Im Vordergrund stehen sollten effiziente und praxistaugliche Werkzeuge, die in unterschiedlichen Entwicklungsstadien einer Anwendungssoftware eine relativ einfache Bewertung des Qualitätsmerkmals Usable Security ermöglichen.

Analytische Evaluation	Empirische Evaluation
Heuristische Evaluation	Metriken
Cognitive Walkthrough	Fragebögen: AttrakDiff, INTUI, PANAS, Bedürfnisskalen
Konsistenz- und Patternprüfung	

Tabelle 4.1: Das Usable-Security-Evaluierungskonzept im Überblick

Bei der Evaluierung eines Prototyps bzw. einer Software kann unterschieden werden zwischen einer analytischen Evaluation und einer empirischen Evaluation. Bei der analytischen Evaluation erfolgt die Bewertung in der Regel durch einen oder mehrere interne Mitarbeiter des Herstellers (bzw. Experten), bei einer empirischen Evaluation dagegen durch mehrere Endnutzer oder externe Probanden. Für beide Anwendungsbereiche erläutern wir die Rahmenbedingungen und stellen entsprechende Methoden und Werkzeuge vor.

Analytische Evaluation

Mit analytischen Evaluationsmethoden kann man sich ohne allzu großen Aufwand eine schnelle Übersicht über die aktuelle Qualität eines Prototyps bzw. eines Produkts verschaffen. Die

Projektphase bzw. Reife des Prototyps ist hierbei nicht relevant. Oft ist es sogar sinnvoll, die durchgeführte Analyse bei späteren Entwicklungsiterationen zu wiederholen. Die analytische Evaluation kann durch einen geschulten internen Mitarbeiter oder durch einen externen Experten durchgeführt werden. Wichtig ist, dass die Person nicht direkt in die Entwicklung des zu bewertenden Produkts involviert ist. Dadurch ist z. B. eine objektive Distanz zu getroffenen Designentscheidungen gewährleistet.

Heuristiken

Ziel der heuristischen Evaluation [1] ist es, generelle Probleme und Sachverhalte aufzudecken, die entweder bei der Entwicklung nicht bedacht oder mit Schwachstellen umgesetzt wurden. Da ein einzelner Tester leicht etwas übersieht, ist es von großem Vorteil, zwei bis drei Mitarbeiter für diese Aufgabe einzuteilen – so lässt sich erfahrungsgemäß die überwiegende Mehrzahl der Fehler finden. Deutlich mehr als drei Mitarbeiter bringen bei der Fehlerfindung dagegen kaum mehr einen Mehrwert [1].

Die bekannten Usabilityheuristiken von Jakob Nielsen [2] sind relativ einfach zu erlernen und anzuwenden. Es sind allgemein anerkannte Prinzipien des Usability-Engineerings wie z. B. „Sichtbarkeit des Systemstatus“, „Übereinstimmung zwischen System und realer Welt“ und „Benutzerkontrolle und -freiheit“. Diese zehn Heuristiken können z. B. in Form von Checklisten eingesetzt werden. Sie bieten einen guten Ansatz, um Benutzeroberflächen auf ihre allgemeine Usability hin zu untersuchen. Die Heuristiken für IT-Securitymanagementtools [3] sind von Nielsens Heuristiken abgeleitet. Sie gehen sowohl auf die Sicherheitsmechanismen eines Systems ein (z. B. anpassbare Darstellung von Informationen) als auch auf die Besonderheiten,

die durch kooperatives Arbeiten in Mehrbenutzersystemen entstehen (z. B. Teilen von Wissen; Planung und Aufteilung der Arbeit zwischen Nutzern). Dadurch sind sie sehr gut für die heuristische Evaluierung im Usable-Security-Kontext geeignet.

Cognitive Walkthrough

Beim Cognitive Walkthrough [4] versetzt sich ein Experte in die Rolle eines Nutzers und versucht, eine vorab definierte Liste von Aufgaben abzuarbeiten. Im Usable-Security-Kontext sollte diese Aufgabenliste sicherheitskritische Schritte beinhalten, damit der Experte beispielsweise erkennen kann, wie gut die Anwendung den zukünftigen Nutzern dabei hilft, Probleme im Umgang mit Sicherheitsfunktionen zu vermeiden bzw. zu lösen. Auch beim Cognitive Walkthrough empfiehlt es sich, dass er von mehreren Mitarbeitern durchgeführt wird, die nicht in die Entwicklung involviert sind. Eine gute Hilfestellung, um sich besser in die Rolle eines potenziellen Nutzers zu versetzen, bieten Nutzerprofile oder sogenannte Personas [5]. Personas sind fiktive Personen, die typische Anwender einer Zielgruppe repräsentieren. Durch das Nachvollziehen der kognitiven Prozesse dieser Nutzer fallen schnell Usability Issues an, die dokumentiert und priorisiert werden können.

Konsistenz- und Patternprüfung

Eine konsistente Verwendung von Layoutelementen, Anwendungslogik und Strukturen ist unentbehrlich für ein gutes Usabilityerlebnis; das gilt selbstverständlich auch für die Sicherheitsfunktionen eines Systems. Mittels einer Checkliste kann man schnell überprüfen, ob eine Anwendung diese Eigenschaften durchgängig erfüllt. Im Usable-Security-Kontext hilft es zudem, die oben genannten Entwurfs- und Gestaltungswerkzeuge zur Prüfung heranzuzuziehen, z. B. können die Usable-Security-Patterns checklistenartig auf ihre

Umsetzung hin überprüft werden. Werden diese Patterns nicht korrekt, nicht konsistent oder überhaupt nicht umgesetzt, bietet das bereits einen guten Hinweis auf mögliche Schwachstellen.

Empirische Evaluation

Als empirische Evaluation wird im Usability-Kontext eine Nutzerstudie bezeichnet, bei der künftige bzw. reale Benutzer mit einem Produkt konfrontiert und während der Interaktion mit diesem Produkt beobachtet werden [6]. Die Beobachtung der Nutzer ermöglicht hierbei Rückschlüsse auf die Stärken und Schwächen des untersuchten Produkts. Generell ist diese Art der Analyse neben lauffähigen Produkten auch bei jeder Prototypstufe (Low Fidelity oder High Fidelity) durchführbar. Für die Durchführung der empirischen Evaluation wird mindestens ein Versuchsleiter benötigt, zudem eine ausreichend große Gruppe von Probanden.

Abhängig vom Ort der Durchführung können zwei Varianten der empirischen Evaluation unterschieden werden: Bei einer Laborstudie testen die Probanden das Produkt innerhalb eines Usabilitylabors, bei einer Feldstudie in der natürlichen Umgebung bzw. unter natürlichen Bedingungen. Eine Feldstudie führt daher oftmals zu realistischeren Ergebnissen, jedoch sind hierbei bestimmte Messungen schwieriger zu erbringen als im Laborversuch, ebenso das Beobachten der Nutzer und das Erfassen von Notizen.

In der Studie bearbeiten die Probanden, die idealerweise das Profil eines typischen Nutzers haben sollten, verschiedene Testaufgaben. Diese Aufgaben sollten ein breites Spektrum an bereits implementierten Funktionen und möglichst alle Sicherheitsmechanismen abdecken. Es sollten Testszenarien konstruiert werden, bei denen der Nutzer auf Gefahren und Sicherheitsrisiken stößt, die abzuwenden sind [7].

Wie der Nutzer an ein solches Sicherheitsrisiko herangeführt werden kann, ist von Fall zu Fall verschieden. Werden in der Studie beispielsweise E-Mail-Systeme getestet, so kann der Nutzer im Test eine Phishing-Mail erhalten bzw. Malware-Attrappen als Anhang zugeschickt bekommen.

Metriken für Usable Security

Metriken dienen im Bereich der Softwarequalität als Instrument, um verschiedene Eigenschaften einer Software bzw. ihrer Entwicklung oder Anwendung zu quantifizieren [8]. Sie ermöglichen es, eine objektive Einschätzung der Qualität einer Software vorzunehmen oder mehrere Softwareprodukte hinsichtlich ihrer Qualität miteinander zu vergleichen. Klassische Usabilitymetriken zielen vor allem auf die Kriterien Effektivität (Erfolgsrate bzw. Fehlerrate bei den Testaufgaben), Effizienz (aufgewendete Zeit) und Zufriedenheit ab. Für den Bereich Usable Security wurde daher von Kainda, Flechais und Roscoe [7] das in **Abbildung 4.1** gezeigte Set von Qualitätsfaktoren vorgeschlagen, das die Bereiche Usability und Security stärker integriert. Zur Überprüfung dieser Qualitätsfaktoren empfehlen sie Metriken, mit denen die Reaktionen von Probanden erfasst werden können. Im Rahmen einer empirischen Evaluation kann der Versuchsleiter anhand dieser Metriken beispielsweise die Bearbeitung von Testaufgaben durch einen Nutzer gezielt beobachten und bewerten.

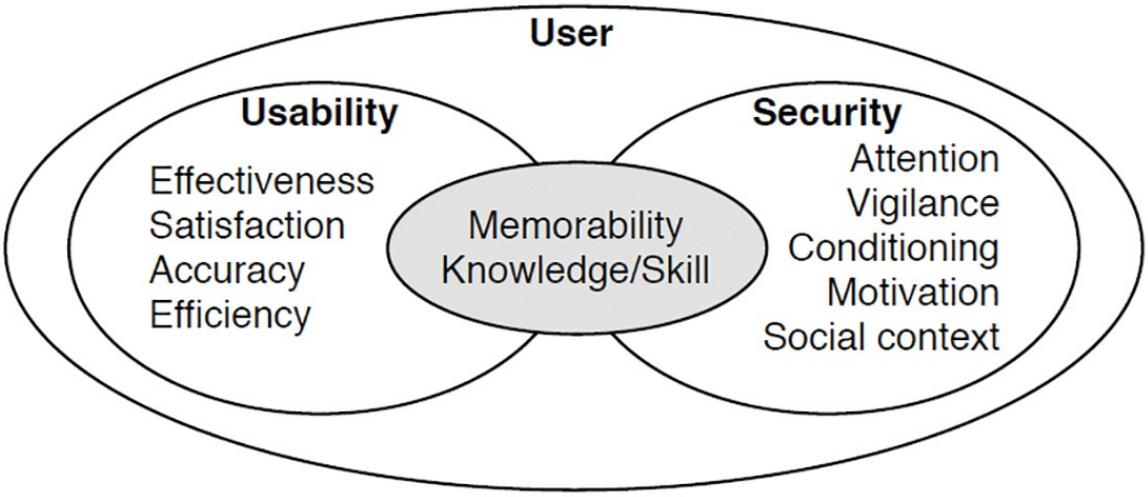


Abbildung 4.1: Security Usability Threat Model nach Kainda, Flechais und Roscoe [7]

Um solche Beobachtungen möglichst strukturiert zu erfassen bzw. zu klassifizieren, wurde im Zusammenhang von Programmierschnittstellen die Verwendung sogenannter Usability Tokens [9] vorgeschlagen und getestet. Anhand dieser Tokens kann zwischen „surprise“ (Nutzer ist überrascht von einem Element, das nicht erwartungskonform ist), „choice“ (Nutzer wird mit einer Auswahl zwischen mehreren Elementen konfrontiert), „missed“ (Nutzer vermisst ein Element), „incorrect“ (Nutzer wendet ein Element falsch an) und „unexpected“ (Nutzer wendet ein Element in unerwarteter Weise an) unterschieden werden. Neben der Verwendung der von Kainda et al. vorgeschlagenen Metriken und solcher Tokens bietet es sich zudem an, Zeitmessungen durchzuführen (das kann Aufschluss über die Effizienz des Probanden bei der Aufgabenlösung geben) und in strukturierter Weise die Vollständigkeit und Genauigkeit der Aufgabenlösung festzuhalten.

Geeignete Fragebögen

Es gibt einige weitere Fragestellungen, die für das Thema Usable Security relevant sind, aber durch die beschriebenen Metriken nicht abgedeckt werden, etwa: Wie schätzt der Nutzer die

Produktqualität ein? Empfindet er die Nutzung des Produkts als intuitiv? Wie fühlt er sich während der Nutzung des Produkts? Sind bestimmte psychologische Grundbedürfnisse, die beim Nutzungserlebnis eine Rolle spielen, erfüllt? Mit einer Auswahl leicht anwendbarer Fragebögen können solche Informationen im Rahmen einer empirischen Evaluation erhoben werden:

- AttrakDiff [10] ist ein Instrument zur Erfassung des wahrgenommenen Produktcharakters und zur globalen Bewertung interaktiver Produkte. Dieser Fragebogen lässt auf die pragmatische und auf die hedonische Qualität des untersuchten Produkts schließen. Wichtig ist das beispielsweise bei der Bearbeitung von Warnhinweisen und dem Abwenden von Gefahren durch Interaktionen des Nutzers, da vor allem unerfahrene Nutzer an sicherheitsrelevante Probleme herangeführt werden müssen.
- INTUI [11] ist ein Instrument zur Erfassung verschiedener Komponenten intuitiver Nutzung. Relevant für Usable Security ist z. B. die Verbalisierungsfähigkeit, da gezeigt werden kann, ob der Nutzer bestimmte Abläufe oder Vorgehensweisen verstanden hat und bei der Bearbeitung der Testaufgaben bewusst oder unbewusst vorhandenes Vorwissen angewendet hat.
- PANAS (Positive And Negative Affect Schedule) [12] ist ein Instrument zur Erfassung der Affektlage des Probanden. Dadurch, dass der Fragebogen die Affektlage in lediglich zwei Dimensionen erfasst (positiv/negativ), können deutliche Erkenntnisse über den Prototyp bzw. über das generelle Befinden des Probanden während der Nutzung erlangt werden, die sonst eventuell unbemerkt blieben.
- Die Bedürfnisskalen [13] sind ein Instrument zur Erfassung des Ausmaßes, in dem psychologische Grundbedürfnisse der

Probanden erfüllt sind. Relevant für den Bereich Usable Security sind insbesondere die Skalen Autonomie, Kompetenz und Sicherheit.

Tabelle 4.2 zeigt, welche Qualitätsmerkmale bzw. -kriterien in einer empirischen Evaluation durch welche der vorgeschlagenen Metriken bzw. durch welches Messverfahren (z. B. Fragebogen, Beobachtung oder Zeitmessung) abgedeckt werden können.

Merkmal	Kriterium	Metrik/Indikator	Messverfahren/Instru
Usability	Effektivität – Vollständigkeit	Erfolgreiche Bearbeitung	Beobachtung/Erfassung durch Versuchsleiter, dreistufige Effektivitätsskala [14]
Usability	Effektivität – Genauigkeit	Erfolgreiche Bearbeitung	Beobachtung/Erfassung durch Versuchsleiter, vierstufige Effektivitätsskala [14]
Usability	Effizienz – Erfüllungszeit	Benötigte Zeit pro Aufgabe	Zeitmessung durch Versuchsleit
Usability	Effizienz – Mentale Beanspruchung	Mühelosigkeit	INTUI [12]
Usability	Zufriedenheit	Emotionen des Probanden, Metriken für Merkmale User Experience (s. u.)	Beobachtung durch Versuchsleit Messverfahren/Instrumente für Merkmale User Experience (s. u.
Security	Aufmerksamkeit	Fehler durch fehlende Aufmerksamkeit	Beobachtung, Befragung durch Versuchsleiter
Security	Wachsamkeit	Fehler durch fehlende Wachsamkeit	Beobachtung, Befragung durch Versuchsleiter
Security	Konditionierung	Fehler durch Konditionierung	Beobachtung, Befragung durch Versuchsleiter
Security	Motivation	Vorteile durch Benutzung von Sicherheitsfunktionen, (Stör-)Anfälligkeit, Barrieren, Schwierigkeiten des Nutzers	Beobachtung, Befragung durch Versuchsleiter

Security	Einprägsamkeit	Erinnerung des Probanden	Beobachtung, Befragung durch Versuchsleiter
Security	Vorwissen/Erfahrung	Erfolgreiche Bearbeitung, Übereinstimmung zwischen Produkt und mentalem Modell des Nutzers	Beobachtung, Befragung durch Versuchsleiter
User Experience	Pragmatische Qualität	Vier Fragebogen-Items	AttrakDiff mini [10]
User Experience	Hedonische Qualität – Identität	Zwei Fragebogen-Items	AttrakDiff mini
User Experience	Hedonische Qualität – Stimulation	Zwei Fragebogen-Items	AttrakDiff mini
User Experience	Attraktivität	Zwei Fragebogen-Items	AttrakDiff mini
User Experience	Positiver/negativer Affekt	10 Fragebogen-Items (fünf positiver Affekt, fünf negativer Affekt)	PANAS (Kurzversion) [13]
User Experience	Erfüllung der psychologischen Grundbedürfnisse nach „Autonomie“, „Kompetenz“ und „Sicherheit“	Neun Fragebogen-Items (je drei pro Bedürfnis)	Bedürfnisskalen [14]
Intuitive Nutzung	Mühelosigkeit	Fünf Fragebogen-Items	INTUI
Intuitive Nutzung	Bauchgefühl	Vier Fragebogen-Items	INTUI
Intuitive Nutzung	Magisches Erleben	Vier Fragebogen-Items	INTUI
Intuitive Nutzung	Verbalisierungsfähigkeit	Drei Fragebogen-Items	INTUI
Intuitive Nutzung	Globales Intuitivitätsurteil	Ein Fragebogen-Item	INTUI

Tabelle 4.2: Übersicht: Empirische Evaluation im Bereich Usable Security

Neben den beschriebenen empirischen Methoden, die auch von kleineren Herstellern mit relativ einfachen Mitteln durchgeführt werden können, bietet das Crowd Testing umfassende Möglichkeiten, Neuentwicklungen hinsichtlich des Qualitätsmerkmals Usable Security zu testen. Da die Tester ihre eigenen Geräte einsetzen, kann insbesondere für eine Vielzahl von Endgeräten getestet werden. Das geschieht in gewohnter, natürlicher Umgebung der Tester, sodass die Nutzertests äußerst realitätsnah sind und eine hohe Aussagekraft haben.

Weitere Informationen zum beschriebenen Evaluierungskonzept und zu allen vorgestellten Methoden und Werkzeugen werden auf der USecureD-Website bereitgestellt. Diese umfassen z. B. Ausführungen zum theoretischen Hintergrund der Evaluierungswerkzeuge und zu weiterführender Literatur (mit Vorlagen der empfohlenen Fragebögen und Hinweisen zur Auswertung und Interpretation der erhobenen Daten).

Links & Literatur

[1] https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

[2] https://www.nngroup.com/articles/ten-usability-heuristics/

[3] Jaferian, Pooya: „Heuristics for Evaluating IT Security Management Tools“, in: Symposium on Usable Privacy and Security (SOUPS), Pittsburgh, 2011

[4] http://www.usabilitybok.org/cognitive-walkthrough

[5] http://www.usability-in-germany.de/definition/personas

[6] Schweibenz, Werner; Thissen, Frank: „Qualität im Web: Benutzerfreundliche Webseiten durch Usability Evaluation“,

Springer, Berlin, 2002

[7] Kainda, Ronald; Flechais, Ivan; Roscoe, Andrew William: „Security and Usability: Analysis and Evaluation“, Oxford University Computing Laboratory, 2010

[8] DIN ISO/IEC 25000: Software-Engineering – Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) – Leitfaden für SQuaRE, 2013

[9] Piccioni, Marco; Furia, Carlo A.; Meyer, Bertrand: „An Empirical Study of API Usability“, in: ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, IEEE, 2013

[10] Hassenzahl, Marc; Burmester, Michael; Koller, Franz: „AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität“, in: Mensch & Computer 2003: Interaktion in Bewegung, S. 187–196, B. G. Teubner, Stuttgart, 2003

[11] Ullrich, Daniel; Diefenbach, Sarah: „INTUI. Exploring the Facets of Intuitive Interaction“, in: Mensch & Computer 2010, S. 251–260, Oldenbourg, München, 2010

[12] Watson, David; Clark, Lee Anna; Tellegen, Auke: „Development and validation of brief measures of Positive and Negative Affect: The PANAS scales“, in: Journal of Personality and Social Psychology, 1988

[13] Sheldon, Kennon M.; Elliot, Andrew J.; Kim, Youngmee; Kasser, Tim: „What is satisfying about satisfying events? Testing 10 candidate psychological needs“, in: Journal of Personality and Social Psychology 80, 2001

[14] http://bit.ly/2nDZsDz

5 Entscheidungshilfen für Anwender

Im letzten Kapitel präsentieren wir zum einen Entscheidungshilfen, die Anwenderunternehmen bei einer einfachen und zielgerichteten Evaluation von Softwareprodukten mit diesem Qualitätsmerkmal unterstützen. Zum anderen stellen wir die Funktionen und Features der USecureD-Plattform vor.

Produktauswahl und -bewertung mit Checklisten

Um Anwenderunternehmen bei einer objektiven Auswahl bedarfsgerechter Softwareprodukte zu unterstützen, haben wir eine Reihe von Checklisten für den Bereich Usable Security erstellt. Neben der Produktauswahl sind diese Checklisten auch sehr gut dazu geeignet, entwickelte Produkte hinsichtlich des Qualitätskriteriums Usable Security zu bewerten. Ein Entwicklerteam kann die Checklisten also beispielsweise nutzen, um einen Prototyp oder ein fertiges Softwaresystem entweder global oder hinsichtlich bestimmter Usability- oder Securityteilaspekte zu evaluieren. Werden die Checklisten für die Softwareauswahl verwendet, so erfasst der Nutzer pro Checklistenpunkt, wie wichtig dieses Kriterium für ihn ist, und notiert bei Bedarf eine ergänzende oder erklärende Bemerkung. Die so erstellte Anforderungsspezifikation kann er anschließend zum Beispiel einem Lieferanten zukommen lassen. Werden die Checklisten für die Evaluation genutzt, so erfasst der Nutzer, ob die einzelnen Prüfpunkte erfüllt oder nicht erfüllt sind, auch hier optional mit einer Anmerkung zur Bewertung.

Checklisten sind ein probates Mittel, um sich wiederholende Abläufe zu strukturieren, zu kontrollieren und zu dokumentieren. Hierzu enthalten sie in korrekter Reihenfolge die erforderlichen Handlungsanweisungen oder Abfrageparameter.

In diesem Fall sind dies Parameter, die die Gebrauchstauglichkeit und Benutzerfreundlichkeit der Sicherheitsfunktionen einer Software betreffen. Grundlage für die Erstellung unserer Checklisten bildeten die im USecureD-Vorhaben entwickelten Entwicklungsrichtlinien und Patterns. Bei den Beschreibungen sämtlicher Richtlinien und Patterns haben wir zunächst eingehend untersucht, welche Aspekte als Prüfpunkte für Checklisten geeignet sind. Im Anschluss an die Identifikation der Prüfpunkte haben wir diese ausformuliert und Bereiche gebildet, um die gesammelten Punkte inhaltlich zu gruppieren. Auf diese Weise entstanden insgesamt vierzehn Checklisten für

- Benutzeroberfläche und Layout
- Terminologie
- Aufgabenorientierung und mentale Belastung
- Steuerbarkeit
- Erlernbarkeit und Hilfe
- Fehlerprävention und Fehlerbehandlung
- Systemhinweise und -feedback
- Sicherheitswarnungen
- E-Mails und Datenübertragung
- Verschlüsselung und Signatur
- Zugangs- und Zugriffskontrolle
- Datenschutz und Datenlöschung
- Formulare
- Systementwicklung

Zuletzt haben wir innerhalb dieser Checklisten jeweils eine logische Reihenfolge für die einzelnen Prüfpunkte festgelegt, z. B.

vom Allgemeinen zum Besonderen oder entsprechend der Abfolge, in der die Prüfpunkte inhaltlich aufeinander aufbauen. Das Beispiel im Kasten „Checkliste: Zugangs- und Zugriffskontrolle“ verdeutlicht diesen Aufbau.

Checkliste: Zugangs- und Zugriffskontrolle

- Die vom System verwendeten Verfahren zur Benutzeridentifikation sind so einfach wie möglich (in Einklang mit dem angestrebten Schutz).
- Mit dem System können Identifikationsmethoden, die in der Organisation bereits existieren, weiter genutzt werden.
- Das System verwendet für die Zugangs- und Zugriffskontrolle technische Lösungen, die die mentale Belastung des Nutzers minimieren; z. B. erlaubt es dem Nutzer, seine Passwörter sicher zu speichern.
- In dem System werden nur dann Passwörter verwendet, wenn dies wirklich notwendig ist. Das System erlaubt dem Nutzer, sein Passwort schnell und einfach zu ändern und ein eigenes Passwort zu wählen.
- Dem Nutzer steht eine einfache Möglichkeit zur Verfügung, mit der er Zugangsdaten zurücksetzen oder wiederherstellen kann.
- Der Nutzer wird nur dann aufgefordert, sein Passwort zu ändern, wenn es einen Hinweis oder einen Verdacht auf eine Gefährdung gibt.
- Die Autorisierung des Nutzers erfolgt mit dem (ersten) Log-in; anschließend ist keine weitere Authentifizierung notwendig, wenn der Nutzer bestimmte Daten anfordert.
- Bei einer bestimmten Anzahl oder Rate von Fehlversuchen sperrt das System automatisch das betroffene Benutzerkonto.

Das System nutzt beim Ändern von Zugangsdaten eine

vordefinierte E-Mail-Adresse, um den Nutzer zu identifizieren und zu autorisieren.

Bei der Anwendung der Checklisten ist zu berücksichtigen, dass nicht alle Prüfpunkte für jeden Anwendungskontext gleichermaßen relevant sind. Vor dem Abarbeiten einer Checkliste sollte daher stets geprüft werden, welche Kriterien in dem konkreten Fall sinnvoll bzw. geeignet sind.

Demonstrator mit Musterlösungen

Um Anwender stärker für das Thema Usable Security zu sensibilisieren, haben wir außerdem einen webbasierten Demonstrator [1] entwickelt, der die in diesem shortcut vorgestellten Methoden und Konzepte visualisiert und begreifbar macht. Das Frontend des Demonstrators (**Abb. 5.1**) setzt auf dem responsiven CSS-Framework Bootstrap [2] auf und arbeitet mit jQuery [3] als JavaScript-Bibliothek. Neben Plug-ins wie Bootstrap-Datepicker [4] und progressStep [5] wurden bei der Implementierung diverse Bibliotheken für Graphics und Cliparts genutzt, z. B. GLYPHICONS [6], Flaticon [7] und Publicdomainvectors.org [8].

Abbildung 5.1: Startseite des webbasierten USecureD-Demonstrators

Nach dem Starten des Demonstrators wird der Benutzer Schritt für Schritt durch mehrere sicherheitsrelevante Anwendungsfälle (vgl. [9]) geführt. Anhand von Infotexten wird ihm hierbei die Bedienung des Demonstrators erläutert, insbesondere werden ihm die zugrunde liegenden Usable-Security-Konzepte erklärt. Beim Durchspielen der Anwendungsfälle erhält der Benutzer jeweils Informationen über die möglichen Folgen und die Tragweite seiner Entscheidungen, z. B. wenn er die E-Mail-Verschlüsselung und -Signatur im Demonstrator aktiviert oder deaktiviert. Bei Bedarf kann er weiterführende Informationen zu den einzelnen USecureD-Werkzeugen abrufen, die in einem Anwendungsfall bzw. Bedienschritt angewendet wurden; diese sind jeweils in den Infotexten verlinkt.

Werkzeugsammlungen für Entwickler

Gebündelt stehen all diese Werkzeuge auf der USecureD-

Plattform zur Verfügung. Wir unterscheiden folgende Arten von Werkzeugen:

- Prinzipien für Usable Security, also allgemeine Grundsätze, denen die Entwicklung bzw. Implementierung folgen sollte.
- Richtlinien für Usable Security, die beschreiben, wie diese Prinzipien konkret umgesetzt werden können.
- Patterns für Usable Security, sprich: wiederverwendbare, bewährte Musterlösungen für typische wiederkehrende Probleme, die während der Systementwicklung auftreten können.

Durch bestehende Verknüpfungen der Werkzeuge untereinander, die auch auf der Plattform visualisiert wurden (Kapitel 2 und 3), können unmittelbar weitere Prinzipien, Richtlinien und Patterns gefunden werden, die für die aktuelle Problemstellung geeignet sind. Softwareentwickler, die in der direkten Verantwortung stehen, Sicherheitsfeatures zu implementieren, können auf diese Weise gleich mehrere verschiedenartige Entwicklungswerkzeuge identifizieren, die Lösungen für die praktische Umsetzung bieten und einfach angewendet werden können.

Zu sämtlichen Werkzeugen sind weiterführende Informationen mitsamt den meist wissenschaftlichen Quellen angegeben. Auf diese Weise schlägt die USecureD-Plattform eine Brücke (Kasten: „USecureD-Plattform“) zwischen teils theoretischen Erkenntnissen aus der Wissenschaft und Forschung und der Anwendung dieses Wissens in der Praxis. Zudem bietet die USecureD-Plattform der Community einen Platz für den Erfahrungsaustausch sowie für Diskussionen und Anregungen zur Verbesserung der Werkzeuge bzw. der Plattform selbst. Dafür steht einerseits eine anonyme Feedbackfunktion für

Anmerkungen und Fehlerberichte zur Verfügung. Andererseits gibt es für angemeldete Benutzer auch die Möglichkeit, über die Werkzeuge im Kommentarbereich zu diskutieren und eigene Erfahrungen oder Optimierungsvorschläge mitzuteilen.

USecureD-Plattform

Die USecureD-Plattform (**Abb. 5.2**) bietet kostenfreie Entwicklungswerkzeuge zur Verbesserung der Usable-Security-Merkmale von Softwareprodukten. Machen Sie gerne von den Werkzeugen Gebrauch und nutzen Sie die Feedback- und Kommentarfunktion der Plattform für Anregungen oder zur Diskussion! Auf diese Weise können Sie unmittelbar zur Verbesserung der Werkzeuge und zu einer lebhaften Community rund um das Thema gebrauchstaugliche Informationssicherheit beitragen.

Mehr Informationen: *https://das.th-koeln.de/usecured*

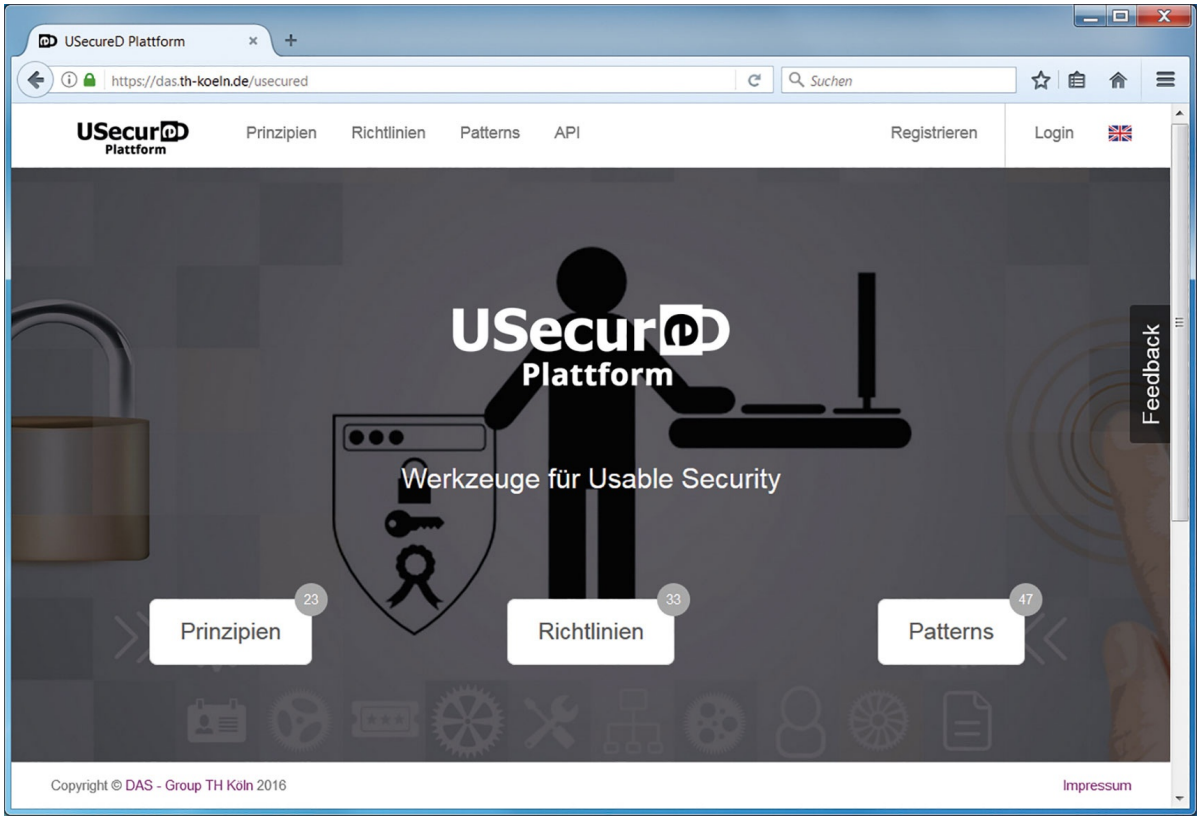


Abbildung 5.2: Die USecureD-Plattform

API zum Abruf der Werkzeuge

Aufgrund der Anregung von Plattformbenutzern wurde die Funktionalität der USecureD-Plattform um ein API für den Abruf der Werkzeuge erweitert. Ein programmatischer Zugriff auf die Werkzeuge über ein REST-basierendes API ermöglicht es Unternehmen, die Werkzeuge in eigene Prozesse zu integrieren oder für Evaluationen im eigenen Unternehmenskontext heranzuziehen.

Als erster Schritt wurde das Datenbeschreibungsformat JSON (JavaScript Object Notation, [10]) als zusätzliche Repräsentation der Werkzeuge – neben HTML – festgelegt. Mit JSON können Daten leichtgewichtig beschrieben werden, wodurch dieses textbasierte Datenformat insgesamt eine weite Verbreitung gefunden hat. Die JSON-Repräsentation enthält alle Informationen und Verknüpfungen, die auch in den Beschreibungsvorlagen der Werkzeuge und auf der Plattform in HTML zu sehen sind. Darüber hinaus werden Listen aller verfügbaren Prinzipien, Richtlinien und Patterns generiert (ebenfalls im JSON-Format), in denen Namen und Verlinkungen zu den einzelnen Werkzeugen enthalten sind. Wie für REST-basierende Web-APIs üblich, wird über den Accept-Header [11] bestimmt, welche Repräsentation eines Werkzeugs zurückgeliefert wird (z. B. *Accept: application/json* für JSON). Da alle Werkzeuge in Englisch und Deutsch verfügbar sind, kann über den Accept-Language-Header zudem die Sprache des angeforderten Werkzeugs eingestellt werden.

Nutzer der Plattform können eine Servicebeschreibung [12] einsehen, die mithilfe des API-Beschreibungsframeworks Swagger [13] erstellt wurde. Diese Beschreibung kann in Programme wie Postman [14] oder den Swagger Editor [15] importiert werden, wodurch Entwickler einen schnellen

Überblick über die Funktionalität des API erhalten und bereits mithilfe der Programmoberfläche erste API-Aufrufe tätigen können. Außerdem ermöglichen diese Programme die automatisierte Erstellung von Programmcode zum Abruf der Werkzeuge in verschiedenen Programmiersprachen. Somit bekommen Unternehmen die Möglichkeit, die im USecureD-Projekt erarbeiteten Werkzeuge in eigene Entwicklungs- oder Evaluierungsprozesse zu integrieren und z. B. für Qualitätserhebungen oder Assessments zu verwenden.

Ausblick

Zusätzlich zu den bestehenden Features der Plattform wird aktuell an weiteren Funktionalitäten gearbeitet, die in künftigen Ausbaustufen der Plattform zur Verfügung stehen sollen. Neben Tools für die analytische und empirische Evaluation, anschaulichen Fallbeispielen und praktischen Auswahlhilfen sind verschiedene Onlineservices für IT-Hersteller und Anwenderfirmen geplant. Ziel ist es, insbesondere den Entwicklern von Sicherheitsfunktionen und -komponenten eine systematische Unterstützung bei der Erreichung des Qualitätsmerkmals Usable Security anbieten zu können, und zwar während sämtlicher Phasen eines Softwareentwicklungsprojekts.

Auch die Forschungsarbeiten, die der Entwicklung der Plattform zugrunde liegen, sollen in wissenschaftlichen Anschlussarbeiten weitergeführt werden. Dies geschieht beispielsweise im Rahmen des Arbeitskreises „Usable Security & Privacy“ der German UPA [16]. Zudem streben die Forschungspartner thematisch passende Folgeprojekte an.

Links & Literatur

[1] https://www.usecured.de/Demonstrator

[2] http://getbootstrap.com/

[3] https://jquery.com/

[4] https://bootstrap-datepicker.readthedocs.io/en/latest/index.html

[5] https://github.com/mateagar/progressStep

[6] http://glyphicons.com/

[7] http://www.flaticon.com/

[8] https://publicdomainvectors.org/

[9] https://www.usecured.de/ergebnisse/

[10] http://www.json.org/json-de.html

[11] https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

[12] https://das.th-koeln.de/usecured/assets/api/swagger.yaml (oder .json)

[13] http://swagger.io/

[14] https://www.getpostman.com/

[15] http://editor2.swagger.io

[16] http://germanupa.de/aktivitaeten/arbeitskreise/usable-security-privacy/

Die Autoren

Hartmut Schmitt ist Koordinator für Forschungsprojekte bei der HK Business Solutions GmbH. Er ist Projektleiter des Projekts USecureD – Usable Security by Design.

Peter Nehren ist wissenschaftlicher Mitarbeiter an der Technischen Hochschule Köln und arbeitet im Bereich Usable Security.

Luigi Lo Iacono ist Professor an der Fakultät für Informations-, Medien- und Elektrotechnik an der Technischen Hochschule Köln. Forschungsgebiet: Software und Security Engineering.

Peter Leo Gorski ist wissenschaftlicher Mitarbeiter an der Technischen Hochschule Köln und arbeitet im Bereich Usable Security und Service Security.