

# Integration von Usability-Methoden in Vorgehensmodelle der Softwareentwicklung im Kontext betrieblicher Umweltinformationssysteme

Felix Hemke<sup>1</sup> und Volker Wohlgemuth<sup>2</sup>

**Abstract:** Der vorliegende Beitrag gibt einen Überblick über verschiedene Vorgehensmodelle zur Softwareentwicklung und stellt die jeweilige Einbindung der Benutzerperspektive in Form der Usability Evaluation dar. Eine Analyse der Vor- und Nachteile des Status Quo soll Verbesserungspotentiale in diesem Bereich aufzeigen, die für die Entwicklung betrieblicher Umweltinformationssysteme (BUIS) vorgeschlagen werden.

**Keywords:** Softwareentwicklungsmodelle, Usability, betriebliche Umweltinformationssysteme

## 1 Einleitung

Eine bedeutende Herausforderung in der Softwareentwicklung besteht darin, Design-Entscheidungen zu treffen, die die Benutzer in Form der Benutzungsschnittstelle direkt betreffen und für den Erfolg der Anwendung maßgeblich sind. Im schlechtesten Fall basieren die Design-Entscheidungen lediglich auf den eigenen Erwartungen und Annahmen der Entwickler sowie auf einer unzureichenden Einbindung der Benutzer selbst, was schnell zu einer problematischen Usability führen kann.

*„Software-Anwendungen oder Produkte weisen eine hohe Usability auf, wenn sie von den vorgesehenen Benutzern einfach erlernt und effizient verwendet werden können und diese damit ihre beabsichtigten Ziele und Aufgaben zufriedenstellend ausführen können.“ [Ric131]*

Es gibt eine Vielzahl von Methoden zur Einbindung von Benutzern in die Entwicklungsprozesse. Dazu zählen etwa die Modellierung von Use Cases, Usability-Tests, Contextual Inquiry und die Heuristische Evaluation [Sar11]. Gerade Unternehmen, die die ersten Versionen von bis heute existierender Software in den 80er oder 90er Jahren begonnen haben, sehen sich mit großen Herausforderungen konfrontiert. Sie tendieren heute, wo das Thema Usability aus der Softwareentwicklung nicht mehr wegzudenken ist, dazu, ein Bündel aus Methoden auf bestehende Softwareprodukte anzuwenden und somit die Benutzerfreundlichkeit in ein weitestgehend fertiges Produkt im Nachhinein zu integrieren. Häufig erfolgt

---

<sup>1</sup> HTW Berlin, Fachbereich 2: Ingenieurwissenschaften – Technik und Leben, Wilhelminenhofstraße 75A, 12459 Berlin, felix.hemke@htw-berlin.de

<sup>2</sup> HTW Berlin, Fachbereich 2: Ingenieurwissenschaften – Technik und Leben, Wilhelminenhofstraße 75A, 12459 Berlin, volker.wohlgemuth@htw-berlin.de

die Umsetzung in einmaligen Projekten. Was ist aber, wenn die Einbindung der Benutzerperspektive ergibt, dass das ganze Datenmodell einer Anwendung der Umsetzung im Sinne der Benutzer entgegensteht oder andere grundlegende Designentscheidungen revidiert werden müssen? Die Lösung liegt darin, die Benutzer in den generell fortlaufenden Entwicklungsprozess einzubinden [Möl10]. Usability Engineering stellt einen Prozess dar, der parallel zur klassischen Planungs- und Entwicklungsarbeit abläuft und die spätere Gebrauchstauglichkeit eines Systems sicherstellen soll [Ram16] (siehe Abb. 1).

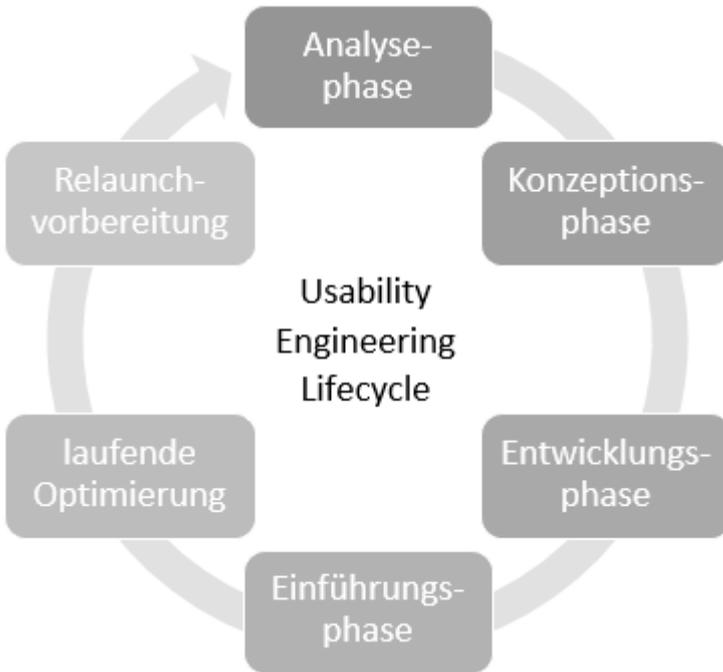


Abb. 1: Phasen des Usability Engineerings nach [Ram16]

Eine beispielhafte Methode des Usability-Engineerings ist die Erstellung von Personas. Dabei handelt es sich um typische Nutzermodelle, die Personen aus einer Zielgruppe charakterisieren, um deren Perspektive bei der Entwicklung zu berücksichtigen. Das generelle Ziel des Usability Engineerings ist die Einpassung eines Produktes in die Welt der Benutzer. In der Usability-Fachwelt gilt es deshalb als ein wichtiger Erfolgsfaktor, die Benutzer bereits in einem möglichst frühen Entwicklungsstadium einzubeziehen [Nie12]. Im besten Fall ist dies, noch bevor ein Prototyp oder ein Codefragment entwickelt wurde. Doch stellt sich die Frage, wenn das Usability-Engineering parallel zum sonstigen Entwicklungsprozess ablaufen soll, wie die Ergebnisse der Benutzereinbindung zuverlässig in die fortlaufenden Entwicklungsentscheidungen einfließen können. Eine nebenläufige Betrachtung des Usability-Engineering zum Softwareengineering birgt die

Gefahr, dass es zu Kommunikations-, Zuständigkeits- oder Terminplankonflikten oder anderen Abstimmungsschwierigkeiten kommt. Vielmehr muss eine integrierte Betrachtung von Methoden zur Steigerung der Usability innerhalb des Softwareentwicklungsprozesses erfolgen. Der vorliegende Beitrag soll deshalb untersuchen, wo die Einbindung des Usability-Engineerings bisher in bestehenden Softwareentwicklungsprozessen zu verorten ist und wie eine bessere Verschmelzung der bisher parallel ablaufenden Prozesse gelingen kann. Mit dem relativ neuen Ansatz der User Story Map wird außerdem ein Konzept vorgestellt, das dabei helfen kann, die Verbesserung der Usability im Entwicklungsprozess zu verankern. Zunächst wird auf die besondere Situation im Bereich betrieblicher Umweltinformationssysteme eingegangen. Anschließend wird vorgestellt, wo die Anwendung von Usability-Methoden in gängigen Softwareentwicklungsmodellen vorgesehen ist.

## 2 Bedeutung von Usability in betrieblichen Umweltinformationssystemen

Der Großteil der heute bestehenden Softwareanwendungen mit einem Fokus auf betrieblichem Umweltschutz existiert etwa seit Anfang bis Mitte der 1990er Jahre, bspw. das seit 1991 von SAP Modul Environment, Health and Safety (SAP EH&S) und die seit 1993 entwickelte Anwendung Umberto des Instituts für Umweltinformatik Hamburg (vgl. [Haa951], [Haa95], [Sch96]). Die als betriebliche Umweltinformationssysteme (BUIS) bezeichneten Softwareanwendungen erfassen Umweltauswirkungen von betrieblichen Prozessen, bilden sie ab und bewerten sie, um die Generierung, Planung und Steuerung von Umweltschutzmaßnahmen und das Umweltmanagement zu unterstützen (vgl. [Woh05]). Viele BUIS fallen damit in das eingangs beschriebene Problemfeld von Anwendungen, deren Version 1.0 vor der Etablierung von agilen Softwareentwicklungs- und Usability-Konzepten im Massenmarkt, entwickelt wurden.

Eine Studie zu Managementsystemen und dem Management natürlicher Ressourcen unter 350 Unternehmen fünf unterschiedlicher Branchen des VDI Zentrums Ressourceneffizienz [VDI17] hat ergeben, dass der Einsatz von Softwareanwendungen zur Unterstützung von Ressourcen- und Energieeffizienzmaßnahmen eher gering ist. So setzen derzeit 30,4% der befragten Unternehmen ein zertifiziertes und 20,5% ein nicht zertifiziertes Softwaresystem für das Umweltmanagement ein, wobei deren Einsatz gerade bei kleinen und mittleren Unternehmen mit 20 bis 249 Mitarbeitern noch geringer ausfällt. Wenngleich diese Anteile relativ hoch erscheinen, muss man einschränkend feststellen, dass die befragten Unternehmen eine relativ hohe Affinität zum Thema Umwelt- und Energiemanagement aufweisen und mit unter *nicht zertifizierten Softwaresystemen* bereits Excel-Dateien, die Ressourcen- oder Energiewerte erfassen, verstanden werden können. Relevante Daten zum Material- und Energieverbrauch, werden nur von einer schwachen Mehrheit der befragten Unternehmen (zwischen 51 und 64,8 Prozent je nach Kennzahl und Betriebsgröße) beobachtet und kontrolliert. Umfassende Instrumente wie Ökobilanzen oder CO<sub>2</sub>-Bilanzierungen kommen selten zum Einsatz. Die Anwendungen

kommen damit ihrem Vorhaben, der Verbesserung der Umweltauswirkungen der befragten Unternehmen, noch nicht im vom VDI erwünschten Maße nach. Vorgeschlagen werden in der Studie zuletzt u. a. Hilfestellungen und einfachere Lösungsansätze, gerade für KMU. Die Befragung der Studienteilnehmer zu den Gründen für die fehlende flächendeckende Ausbreitung von Umweltmanagementsystemen hat u.A. folgendes ergeben [VDI17]:

- Die Datenerfassung stellt einen zu hohen personellen Aufwand im betrieblichen Alltag dar,
- demgegenüber besteht ein schlechtes bzw. nicht kalkulierbares Kosten-Nutzen-Verhältnis,
- verbunden mit hohen Kosten zur Einführung eines Managementsystems.
- Außerdem bedeutet die oftmals zu geringe Qualifikation der Mitarbeiter eine Notwendigkeit zur Einbindung von externen Beratern

Bei der Reduzierung des Aufwands zur Nutzung der Softwareanwendungen spielt die Benutzerfreundlichkeit eine Schlüsselrolle. Eine gute Usability einer Software und eine damit einhergehende intuitive Bedienung durch die Benutzer verringern den Aufwand zum Erlernen bzw. Wiedererlernen und senken die Fehlerraten und den Supportaufwand (vgl. [Ric13]). Wenn die Softwareentwickler die Schwierigkeiten der unterschiedlichen Benutzergruppen im Vorfeld genau einschätzen können, könnten sie diese durch entsprechende Vorkehrungen reduzieren. So kann der Workflow der Anwendung auf die Arbeitsweise der Benutzergruppen angepasst werden, es können verbesserte Hilfemodule bereitgestellt werden und wiederkehrende Arbeitsschritte mittels Einrichtungsassistenten und Templates unterstützt werden.

In diesem Kontext wird die These aufgestellt, dass eine Verbesserung der Usability von BUIS, bedingt durch eine höhere Effizienz und Effektivität der Benutzung und eine höhere Zufriedenheit der Benutzer, eine verbesserte Datenlage und Auswertung nach sich ziehen und damit einen Beitrag zur Steigerung der Nachhaltigkeit im Unternehmen leistet.

### **3 Einordnung der Usability-Evaluation in bestehende Softwareentwicklungsmodelle**

#### **3.1 Wasserfallmodell**

Das erste Vorgehensmodell, das bis zur Entwicklung agiler Modelle weitläufig in der Software-Entwicklung eingesetzt wurde, ist das Wasserfallmodell [Kuh12]. Es sieht eine Entwicklung in fünf bzw. sieben chronologisch ablaufenden Phasen vor. Je nach Variante werden die Anforderungen an die Software in der ersten (Anforderungsanalyse) bzw. der

zweiten Phase (Softwareanforderungsanalyse) durch die Auftraggeber und andere bedeutende Stakeholder definiert. Das Ergebnis wird in der Regel in einem Lastenheft dokumentiert. Aus Benutzerperspektive hat dieses Vorgehen entscheidende Nachteile:

- Die Auftraggeber/Stakeholder sind nicht gleichbedeutend mit den Endanwendern und können die Bedürfnisse dieser oftmals nur durch Annahmen abschätzen.
- Die Beteiligung von Benutzern ist hier allenfalls in der Definitionsphase vorgesehen.
- Es handelt sich um ein chronologisches (nicht iteratives) Vorgehensmodell, so dass es keine Möglichkeiten zur Korrektur gibt, wenn die Anwender nach der Umsetzung der Anforderungen feststellen, dass das Ergebnis nicht den eigenen Erwartungen entspricht.

### 3.2 V-Modell

Basierend auf dem Wasserfallmodell wurde anschließend das V-Modell entwickelt [Kuh15], welches zu jeder Entwicklungsphase eine Teststufe vorsieht und damit eine einmalige nachträgliche Evaluation der Anforderungen ermöglicht, ansonsten jedoch ähnliche Schwächen bezüglich der Integration der Benutzeranforderungen mittels Usability-Methoden aufweist. Insgesamt lässt sich für die beide Modelle feststellen, dass sie zwar relativ leicht anzuwenden sind. Jedoch, sehen die Endanwender das Produkt erst ganz am Ende und stellen dann häufig fest, dass zwischen der Anforderungsanalyse und der Realisierung des Endproduktes Entscheidungen getroffen wurden, die nicht in ihrem Sinne sind. Aufgrund des linearen Ablaufs und der fehlenden Rückkopplungen sind beide Modelle eher ungeeignet, um sie für eine prozesshafte Integration der Interessen der Benutzer in den Entwicklungsprozess einzusetzen. Ein iteratives Design-Vorgehen ist als erfolgsversprechender anzusehen [Nie12].

### 3.3 Agile Prinzipien

Als Alternativen, die ein iteratives Vorgehen beinhalten, haben agile Entwicklungsprinzipien im letzten Jahrzehnt an Bedeutung gewonnen, wobei das Scrum-Modell die größte Bedeutung zur Umsetzung der agilen Prinzipien eingenommen hat [Dau15]. Die gemeinsame Nutzung von Usability-Methoden und agilen Entwurfsmodellen gilt große Herausforderung, für die es bisher noch keine Universallösung gibt [Lor17]. Aus Benutzerperspektive ist im Scrum-Modell als problematisch anzusehen, dass die Rollen in einem klassischen Scrum-Team keine feste Integration von Benutzern vorsehen (vgl. [Sch07]):

- Der Product-Owner vertritt die fachliche Auftraggeberseite für die Anwendung.
- Der Scrum-Master sorgt dafür, dass der Scrum-Prozess korrekt umgesetzt wird.

- Das Scrum-Team besteht aus Entwicklern, Architekten, Testern, technischen Redakteuren.
- Die Vertreter der Rollen werden dabei in sog. „Pigs and Chickens“ (auf Deutsch „Schweine und Hühner“, vgl. [Sch07]) eingeteilt, wobei die Pigs selbst an der Umsetzung der Arbeiten beteiligt sind und größere Entscheidungsbefugnisse besitzen, wohingegen Chickens relativ unbeteiligte Personen sind, die zwar involviert werden, aber letztlich keine bedeutende Verantwortung übernehmen.

Diese Übersicht macht deutlich, dass zwar die Kundenseite prinzipiell durch den Product-Owner repräsentiert wird, was aber nicht gleichbedeutend mit den Endanwendern ist. Außerdem macht die Einteilung in Pigs and Chickens deutlich, dass im klassischen Scrum-Modell vor Allem jene Personen ein Entscheidungsrecht haben, die an der Umsetzung der Entwicklungsarbeiten maßgeblich beteiligt sind. Eine Möglichkeit besteht darin, das Scrum-Team um Benutzer und Usability-Experten zu ergänzen oder dem Product-Owner einen Usability Engineer als Coach zur Seite zu stellen [Mem09].

Die Aufgaben werden in sog. Sprints bearbeitet, wobei zu jedem Sprint neben der Entwicklung und Dokumentation das iterative Testen gehört. Dadurch wird der Aufwand für notwendige Anpassungen kleiner als wenn die Tests erst am Ende der Entwicklung erfolgen. Nach [Lor16] sollten neben den Funktions-, Integrations- und Systemtests iterative Usability-Tests in möglichst jedem Sprint eingesetzt werden.

### **3.4 Einbindung von Usability-Methoden in Scrum**

Das Scrum Vorgehensmodell beinhaltet keine vom Umfang her mit dem Wasserfallmodell vergleichbare Analysephase. Die Anforderungen werden vielmehr im Laufe der Entwicklung Schritt für Schritt spezifiziert und im Product Backlog dokumentiert. Besondere Relevanz zur besseren Verbindung von Usability-Methoden mit agilen Entwicklungskonzepten hat der Ansatz, Prototypen und Klick-Dummys in der Konzeptionsphase zu nutzen, um im Scrum-Team noch vor Beginn der Entwicklung zu ermitteln, ob die Konzeptideen Anforderungen der Benutzer tatsächlich erfüllen [Str12].

Die Erstellung der Product Backlogs eignet sich darüber hinaus gut zur Einbindung von Usability Methoden, wie der User Story Map von Jeff Patton [Pat08] (siehe Abb. 2).

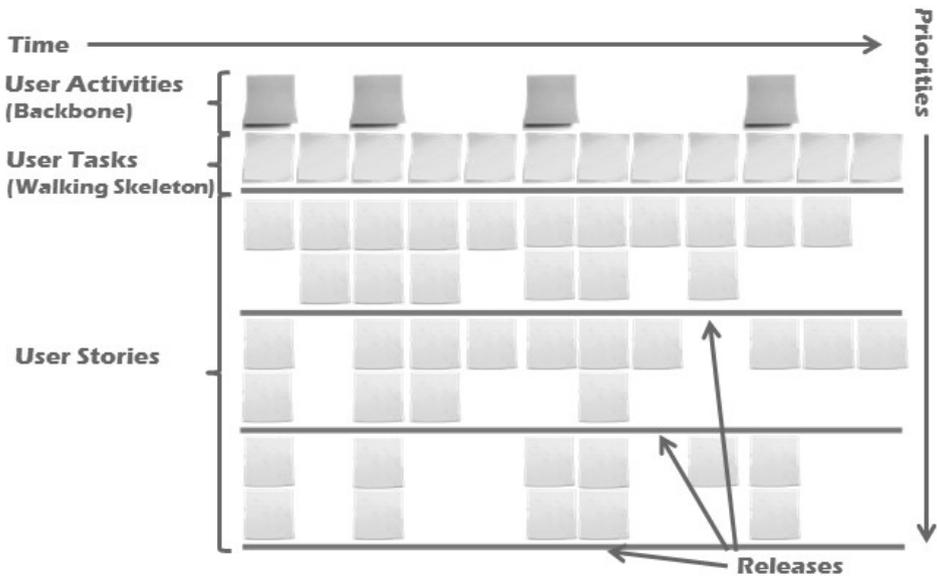


Abb. 2: User Story Maps nach Patton (Quelle: Steve Rogalsky)

User Story Maps stellen eine Entwicklungstechnik dar, mit der gemeinsam im Team User Storys erstellt werden können, die die Anforderungen an die Anwendung in der Sprache der Benutzer beinhalten. Das Ergebnis besteht aus horizontal angeordneten:

- Activities/Epics, die die wichtigsten übergeordneten Ziele der Benutzer im Workflow mit der Anwendung darstellen.
- Tasks, die die Activities in kleinere, überschaubarere Einheiten aufteilen. Ein Task stellt aus Benutzersicht eine Tätigkeit da, um das Ziel, das mit der Anwendung verfolgt wird, zu erreichen.
- User Stories stellen in Alltagssprache formulierte kurze Software-Anforderungen dar. Sie bilden die Activities in granularer Form ab.

User Story Maps haben zum Zweck, die Anforderungen der Benutzer möglichst fein durchzuplanen, ohne dabei von den übergeordneten Zielen der Anwendung abzuweichen. Außerdem sind sie gut geeignet, um sie mit fachlich sehr heterogenen Teams zu erstellen. Dadurch lassen sich die Ideen von Entwicklern, Testern, Anwendern, Designern und Controllern gleichermaßen berücksichtigen.

## 4 Fazit

Eine integrierte Betrachtung von Usability-Methoden in den Softwareentwicklungsprozess stellt sicher, dass die Anwendung die Anforderungen der Benutzer erfüllt und durch diese effizient und effektiv genutzt werden kann. Außerdem macht das Arbeiten mit einer intuitiv am Nutzer orientierten Anwendung mehr Spaß. Gerade bei BUIS hängt der weitere Erfolg bei der Ausbreitung im betrieblichen Alltag maßgeblich davon ab, dass der Aufwand zur Einarbeitung und Nutzung möglichst verringert wird. Die größte Herausforderung in diesem Bereich besteht darin, in langjährig bestehenden Entwicklungsstrukturen ein neues Paradigma zu berücksichtigen, in dem die Benutzer bereits von Anfang an in den Entwicklungsprozess integriert werden. Konsequenterweise bedeutet das, dass einige Anwendungen aus diesem Bereich von Grund auf neu entwickelt werden müssten, da technische Limitierungen ein benutzerfreundliches Redesign verhindern können. Eine Einbindung der Benutzerperspektive hat das Potential, dass die Anwendungen ihrem eigentlichen Zweck, der Reduzierung von negativen betrieblichen Umweltauswirkungen, besser gerecht werden. Beispielsweise könnte die fehlende Datenbasis zur Umsetzung von Umweltmanagemententscheidungen dadurch verbessert werden, dass mobile Datenerfassungswerkzeuge entwickelt werden, die einen spielerischen Ansatz mitbringen und so ein gesteigertes Interesse bei den Benutzern wecken, umweltrelevante Daten zu erfassen. Die Einbindung von Usability-Methoden ist zwar prinzipiell in jedem Entwicklungsmodell möglich, jedoch eignen sich agile Entwicklungsmethoden wie Scrum besonders gut, da sie iterierbar sind und die Vorgaben keine strikten Gesetzmäßigkeiten darstellen, sondern sie bewusst sehr flexibel für Änderungen sind.

## Literaturverzeichnis

- [Dau15] Daut P. (2015): computerwoche. <https://www.computerwoche.de/a/scrum-und-co-auch-agile-methoden-lassen-sich-skalieren,3219768>, zuletzt geprüft 2017.
- [Haa95] Haasis H.-D.; Hilty L.M.; Kürzl H. et al. (1995): Betriebliche Umweltinformationssysteme, Projekte und Perspektiven. Umweltinformatik aktuell, Bd. 5. Marburg: Metropolis-Verlag.
- [Haa951] Haasis H.-D.; Hilty L.M.; Hunscheid J. et al. (1995): Umweltinformationssysteme in der Produktion. Umweltinformatik aktuell, Bd. 6. Marburg: Metropolis-Verlag.
- [Kuh12] Kuhrmann M. (2012): Enzyklopädie der Wirtschaftsinformatik - Wasserfallmodell. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Wasserfallmodell>, zuletzt geprüft am 12.05.2017.
- [Kuh15] Kuhrmann M. (2015): Enzyklopädie der Wirtschaftsinformatik - V-Modell XT. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/V-Modell-XT>, zuletzt geprüft am 12.05.2017.
- [Lor16] Loranger H. (2016): Top 10 Tips for UX Success From Agile Practitioners. <https://www.nngroup.com/articles/ux-success-agile/>, zuletzt geprüft 2017.

- [Lor17] Loranger H. und Laubheimer P. (2017): The State of UX Agile Development. <https://www.nngroup.com/articles/state-ux-agile-development/>, zuletzt geprüft 2017.
- [Mem09] Memmel T. (2009): Agile + Usability: Produkte mit hoher Bedienqualität agil entwickeln. <http://www.usabilityblog.de/2009/08/agile-usability-produkte-mit-hoher-bedi-en-qualitat-agil-entwickeln-teil-2/>, zuletzt geprüft 2017.
- [Möl10] Möller S. (2010): Quality Engineering. Berlin: Springer. S. 57.
- [Nie12] Nielsen J. (2012): Usability 101: Introduction to Usability. <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>, zuletzt ge-  
prüft 2017.
- [Pat08] Patton J. (2008): The New User Story Backlog is a Map. <http://jpattonassocia-tes.com/the-new-backlog/>, zuletzt geprüft 2017.
- [Ram16] Rampl H. (2016): Handbuch Usability. <http://www.handbuch-usability.de/usability-en-gineering.html>, zuletzt geprüft 2017.
- [Ric13] Richter M. und Flückiger M. (2013): Usability Engineering kompakt. Berlin Heidelberg: Springer Vieweg. S. 4.
- [Sar11] Sarodnick F. und Brau H. (2011): Methoden der Usability Evaluation. Frankfurt, Mün-  
chen: Verlag Hans Huber. S. 119-203.
- [Sch07] Schwaber K. (2007): Agiles Projektmanagement mit Scrum. München: C.H.Beck.
- [Sch96] Scheer A.-W.; Haasis H.-D.; Hilty L.M. et al. (1996): Computergestützte Stoffstromma-  
nagement-Systeme. Umweltinformatik aktuell, Bd. 8. Marburg: Metropolis-Verlag.
- [Str12] Struckmeier A. (2012): Iterativ testen – Warum eigentlich? <http://www.usabili-tyblog.de/2012/09/iterativ-testen-warum-eigentlich/>, zuletzt geprüft 2017.
- [VDI17] Ressourceneffizienz V.Z. (2017): Ressource Deutschland. [http://www.ressource-deutschland.de/fileadmin/user\\_upload/downloads/studien/VDI\\_ZRE\\_Studie\\_Manage-mentsysteme\\_und\\_das\\_Management\\_natuerlicher\\_Ressourcen.pdf](http://www.ressource-deutschland.de/fileadmin/user_upload/downloads/studien/VDI_ZRE_Studie_Manage-mentsysteme_und_das_Management_natuerlicher_Ressourcen.pdf), zuletzt geprüft 2017.
- [Woh05] Wohlgemuth V. (2005): Komponentenbasierte Unterstützung von Methoden der Mod-  
ellbildung und Simulation im Einsatzkontext des betrieblichen Umweltschutzes. Ham-  
burg: Shaker Verlag.