

Zur Nachhaltigkeit von Gestaltungslösungen: Prototyping und Patterns

Manuel Burghardt, Tim Schneidermeier, Christian Wolff

Lehrstuhl für Medieninformatik, Universität Regensburg

Zusammenfassung

Zu den besonderen Kennzeichen von Design Thinking-Prozessen gehören einerseits frühzeitiges Umsetzen und Evaluieren von Design-Ideen in Form von Prototypen, und andererseits die Iteration von Design-Schritten im Falle eines negativen Testergebnisses. Das Produkt entsteht dabei letztendlich auf Basis eines evolutionären Designprozesses. In diesem Beitrag wird das Pattern-Konzept als Möglichkeit diskutiert, Designwissen während des gesamten Entwicklungsprozesses zu dokumentieren und nachhaltig für andere Designprozesse verfügbar zu machen.

1 Einleitung und Motivation

Der Design Thinking-Prozess lässt sich als allgemeiner Ansatz zur Lösung bestehender Probleme oder zur Entwicklung von Innovationen verstehen. Dabei spielt die Berücksichtigung des Nutzungs- und Nutzerkontexts bei der Ideengenerierung eine ebenso große Rolle wie die frühzeitige Evaluation von prototypisch umgesetzten Designideen. Damit offenbart der Design Thinking-Ansatz viele Gemeinsamkeiten mit benutzerzentrierten Entwicklungsansätzen wie sie z.B. durch die ISO-Norm 9241-210 „*Ergonomics of Human-System Interaction – Part 210: Human-Centered Design Process for Interactive Systems*“ (ISO 9241-210, 2010) beschrieben sind (Burghardt et al. 2011). Ein wesentliches Kennzeichen solcher Prozesse ist nicht nur das frühzeitige Umsetzen von Ideen durch Prototyping sowie das Evaluieren dieser Prototypen durch Benutzertests, sondern auch das Zurückspringen in frühere Entwicklungsstadien, im Falle der negativen Evaluation einer prototypisch umgesetzten Designidee. Eine wesentliche Herausforderung bei dieser Art der iterativen Produktentwicklung ist somit die *Dokumentation von Designentscheidungen* während des gesamten Prozesses, um bestimmte Entscheidungen auch später nachvollziehbar zu machen. In bestimmten Bereichen des *Software Engineering* ist eine solche Dokumentation im Sinne verbesserter *Traceability* bereits gängige Praxis (Turban et al. 2009). Die nachhaltige Dokumentation sowohl positiver als auch negativer Designideen verhindert, dass bestimmte Fehler wiederholt begangen werden,

und bietet für ähnliche Problemkontexte und künftige Entwicklungen standardisierte Lösungsansätze. Im Folgenden soll das aus der Architektur stammende Konzept von *Gestaltungsmustern* (Alexander 1977) als grundlegende Möglichkeit der Dokumentation von positiven und negativen Designentscheidungen während iterativer Entwicklungsprozesse diskutiert und anhand eines konkreten Beispiels aufgezeigt werden.

2 Design-Patterns

Das Konzept der Design-Patterns wurde erstmals von Alexander (1977; 1979) im Kontext der Architektur von Städten vorgestellt. Alexander präsentierte eine Sammlung von Patterns, die Design-Lösungen für typische Problemstellungen in der Domäne der architektonischen Städteplanung enthalten. Einige Jahre später wurde das Pattern-Konzept von Gamma et al. (1995) erfolgreich in den Bereich des *Software Engineering* übertragen. Seither wurden für die unterschiedlichsten Problemdomänen Design-Patterns entwickelt (vgl. PloP¹) unter anderem auch für den Bereich der *Human-Computer Interaction* (HCI) (Borchers, 2001). Generell werden Patterns dazu verwendet, implizites Designwissen („quality without a name“, Borchers 2001, 11) in einer expliziten, systematischen Struktur zu dokumentieren, und somit dieses Wissen nachhaltig verfügbar und wiederverwendbar zu machen. Patterns dienen somit als Container für positiv evaluierte Lösungen zu einem wiederkehrenden Problem und können allgemein als Wissensmanagementstrategie für die Softwareentwicklung betrachtet werden. Ein wesentliches Merkmal von Patterns ist deren einheitliche Struktur. Dadurch, dass Patterns allesamt immer den gleichen formalen Aufbau aufweisen, können sie einerseits gut miteinander kombiniert und gegeneinander abgewogen werden. Andererseits wird dadurch die Lesbarkeit und vor allem Verständlichkeit des beschriebenen Patterns garantiert. Borchers (2001) identifiziert insgesamt neun Charakteristika, die in allen Architektur-Patterns von Alexander zu finden sind:

1. Der Name des Patterns sollte die grundlegende Idee bereits zum Ausdruck bringen.
2. Bewertung der universellen Einsetzbarkeit, d.h. ein Wert von 0 bis 2 gibt Auskunft darüber wie breit das Pattern für unterschiedliche Problembereiche eingesetzt werden kann/konnte.
3. Beispielbild (Foto), das die praktische Anwendung des Patterns illustriert.
4. Kontext in dem das Pattern benutzt werden kann (Verhältnis zu übergeordneten Patterns; optional)
5. Kurzzusammenfassung des Problems (Konflikt) welches das Pattern löst.
6. Ausführliche Beschreibung des Problems, ggf. mit empirischen Daten sowie Diskussion bestehender Lösungsansätze.

¹ Konferenz Pattern Languages of Programs (<http://hillside.net/conferences/plop>)

7. Beschreibung des Lösungsansatzes, welcher durch das Pattern vorgegeben wird; der Lösungsansatz ist klar verständlich, aber gleichzeitig so generisch, dass er auf mehrere Szenarien angewendet werden kann.
8. Ein Diagramm, das den Lösungsansatz illustriert: "If you can't draw a diagram of it, it isn't a pattern." (Alexander 1979, 267)
9. Referenz zu untergeordneten Patterns (optional)

Ein weitergehender Ansatz zur formalen Standardisierung für Design-Patterns wurde in einem Workshop 2003 auf der internationalen Fachkonferenz ACM SIGCHI² in Form der XML-basierten Beschreibungssprache PLML³ erarbeitet (Fincher et al. 2003). Die aktuell umfangreichsten und bekanntesten Pattern-Sammlungen im Bereich der HCI (Tidwell 2011; Welie, van 2008) verwenden diesen Ansatz bzw. bauen darauf auf.

Im Folgenden soll diskutiert werden, ob und wie Patterns eine sinnvolle Ergänzung zum Prototyping während des Designprozesses, und damit des Designdenkens, darstellen können. Ein Praxisbeispiel illustriert die Vor- und Nachteile eines solchen Ansatzes.

3 Prototyping und Patterns im Designprozess

Abbildung 1 zeigt einen typischen Designprozess mit evolutionärem Prototyping, bei dem die unterschiedlichen Stadien und Erkenntnisse des Prozesses im jeweiligen Prototypen dokumentiert werden.



Abbildung 1: Typischer Designprozess mit evolutionärem Prototyping. Die Designerkenntnisse werden durch die unterschiedlichen Prototypen dokumentiert, und am Schluss in einem Endprodukt implementiert.

Ein Prototyp als Umsetzung und Sicherung von Designideen gleichermaßen, steht zwar für geringen Dokumentationsaufwand, bringt aber auch wesentliche negative Aspekte mit sich: dadurch, dass Prototypen für einen spezifischen Zweck, also für einen genau definierten Problemkontext, erstellt werden, sind sie nicht ohne weiteres auf andere Kontexte übertragbar. So läuft Designwissen, das in Form projektspezifischer Prototypen dokumentiert ist, Gefahr im Laufe der Zeit verloren zu gehen. Bereits erlangte Erkenntnisse müssen ggf. in Folgeprozessen neu erarbeitet werden. Ein weiterer Nachteil ergibt sich für die Kommunikation von Designideen an unterschiedliche *Stakeholder* – ein Prototyp stellt immer nur das

² <http://chi2012.acm.org/>

³ Pattern Language Markup Language (PLML)

Endergebnis, also die Lösung eines bestimmten Designproblems dar. Welches Problem damit gelöst wird, und warum eine Lösung zugunsten einer anderen vorgezogen wurde, kann aus dem Prototypen allein nicht ohne Weiteres nachvollzogen werden. Abbildung 2 zeigt einen Designprozess mit evolutionärem Prototyping, bei dem das Designwissen, das in einem spezifischen Prototypen implementiert ist, zusätzlich als Design-Pattern formuliert wird.

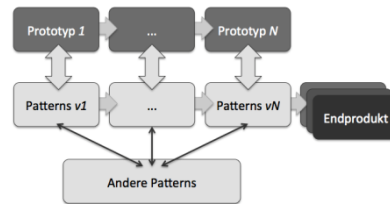


Abbildung 2: Designprozess, in dem evolutionäres Wissen zusätzlich in Form von Patterns festgehalten wird.

Die Patterns werden während des gesamten Designprozesses angepasst und iterativ verbessert. Das Produkt wird aus den (durch Prototypen) validierten Patterns zur Lösung eines bestimmten, wiederkehrenden Designproblems erstellt. Durch ihren generischen Charakter können Patterns leicht mit Patterns aus anderen Entwicklungsprozessen kombiniert werden, oder bei abweichenden Kontextbedingungen variiert und erweitert werden. So ist die erstmalige Erstellung eines Design-Patterns zwar zunächst mit zusätzlichem Aufwand verbunden, gleichzeitig aber die nachhaltige Dokumentation und damit Sicherung von Designwissen, insbesondere validierter Lösungen für konkrete Probleme, garantiert. Folglich können Patterns ihre Stärke vor allem im Kontext von *Design Reuse* (Sivaloganathan & Shahin 1999) oder auch einer langfristigen und nachhaltigen Produktlinienentwicklung (Böckle et al. 2004) ausspielen.

4 Praxisbeispiel

Im Rahmen eines Kooperationsprojekts⁴ aus Industrie und Forschung sollten generische Gestaltungslösungen für Regel- und Steuerungsgeräte im Umfeld von Haus- und Umwelttechnik entwickelt werden. Um das komplexe und weitläufige Feld zu erfassen wurden zunächst in einer ersten Analysephase Anforderungen und Kontextfaktoren erhoben. Die Entwicklung eines konkreten Produkts diente dazu, erste Designkonzepte zu erarbeiten, umzusetzen, zu evaluieren und iterativ zu verbessern (Böhm et al. 2011a; 2011b). Im Folgenden soll aufgezeigt werden, wie solche Konzepte zunächst als Prototypen umgesetzt, gute Lösungen in Form von Design-Patterns festgehalten und im Laufe des Projekts iterativ angepasst, erweitert und verbessert wurden.

⁴ Das Projekt wurde durch die Software-Offensive Bayern gefördert; FKZ IUK 0910-0003. Projektdauer: 12/2009 – 3/2012

4.1 Design-Patterns als Dokumentationswerkzeug im Entwicklungszyklus

Basierend auf den erhobenen Anforderungen für Zielgruppe, Funktionsumfang (*use cases*) und ergänzenden Kontextfaktoren (Hardware, Display etc.) wurden erste Prototypen in Form von Skizzen erstellt und iterativ verbessert. Die Ergebnisse wurden anschließend in einem klickbaren Prototypen umgesetzt. Beide Prototypen dienen zum einen der Dokumentation und Kommunikation des Designs, zum anderen konnte der interaktive Prototyp als Testartefakt für Usability-Studien verwendet werden. Als zusätzliches Dokumentationswerkzeug wurden Design-Patterns verwendet. Patterns bieten neben den oben beschriebenen Vorteilen aus ihrem Selbstverständnis heraus vor allem den Zugewinn, dass nicht nur der Ist-Zustand der Gestaltungslösungen festgehalten wird, sondern auch zusätzliche Informationen (z. B. Begründung für das Design) entsprechend der vorgegebenen Struktur festgeschrieben werden (müssen). Dies garantiert auch in Zukunft die Nachvollziehbarkeit der getroffenen Design-Entscheidungen. Für die Dokumentation während des Projekts wurde eine Pattern-Struktur erarbeitet, die auf den Ursprüngen von Christopher Alexander und den Ergebnissen der *PLML* (Fincher et al. 2003) basiert, und für den konkreten Projektkontext angepasst wurde (vgl. Abbildung 3). Das Schema gewährleistet die strukturierte Erfassung der design-relevanten Informationen.

Pattern-Schema

- # Name
- # Alias
- # Konfidenzlevel
- # Abbildung
- # Problem
- # Kontext
- # Voraussetzungen und Einschränkungen
- # Zusammenfassung
- # Lösung
- # Begründung
- # Beispiel
- # Verwandte Patterns

Abbildung 3: Verwendetes Pattern-Schema.

Eine ganz ähnliche Idee – Begründungen für Designentscheidungen festzuhalten und diese nachvollziehbar zu machen – wird mit (formlos) annotierten Prototypen verfolgt (z. B. *Axure*). Ohne eine vordefinierte Form neigen diese Annotationen jedoch oft dazu, Informationen nur spärlich und/oder unstrukturiert zu enthalten. Solche annotierte Prototypen werden häufig ausschließlich für die Kommunikation mit Stakeholdern benutzt, und nach Abschluss des konkreten Projekts wieder verworfen. Es existiert keine Versionierung oder andere Archivierungsansätze um das Wissen strukturiert für die Zukunft festzuhalten.

4.2 Konkretes Designproblem: *Gestaltung der Nutzereingabe*

Anhand eines konkreten Designproblems – der Gestaltung der Nutzereingabemodalität für ein Regelungsgerät – soll die Evolution eines Prototypen, die unterschiedlichen Lösungen und das Dokumentieren in Form von Design-Patterns kurz aufgezeigt werden. Abbildung 4 illustriert eine erste Gestaltungslösung in der Frühphase des Designprozesses.

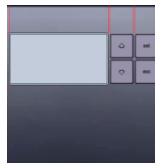


Abbildung 4: Erstes Design für die Nutzerinteraktion.

Nutzereingaben erfolgen mit Hilfe von zwei Pfeiltasten (auf und ab), einer *Set*- und einer *Escape*-Taste. Aus der prototypischen Abbildung ist zwar ersichtlich, welche Designentscheidung getroffen wurde, es lässt sich jedoch nicht nachvollziehen, warum diese Entscheidung getroffen wurde: warum werden Tasten verwendet, und nicht etwa ein berührungssensitiver Bildschirm, aus welchem Grund wurden die Labels *Set* und *Escape* gewählt, und wieso werden genau vier Tasten benötigt? Tatsächlich erwies sich das zunächst entworfene Design nach der Durchführung von Usability-Studien als nicht optimal für den speziellen Produkt- und Nutzungskontext. Auf Basis der Evaluationsergebnisse wurde das Design überarbeitet und erneut auf Benutzerfreundlichkeit getestet. Abbildung 5 zeigt die überarbeitete und zugleich finale Gestaltung der Nutzereingaben. Die Navigationstasten wurden durch einen klickbaren Drehencoder (*clickwheel*) ersetzt und die *Escape*-Taste neu angeordnet.

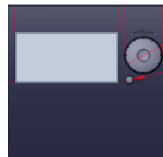
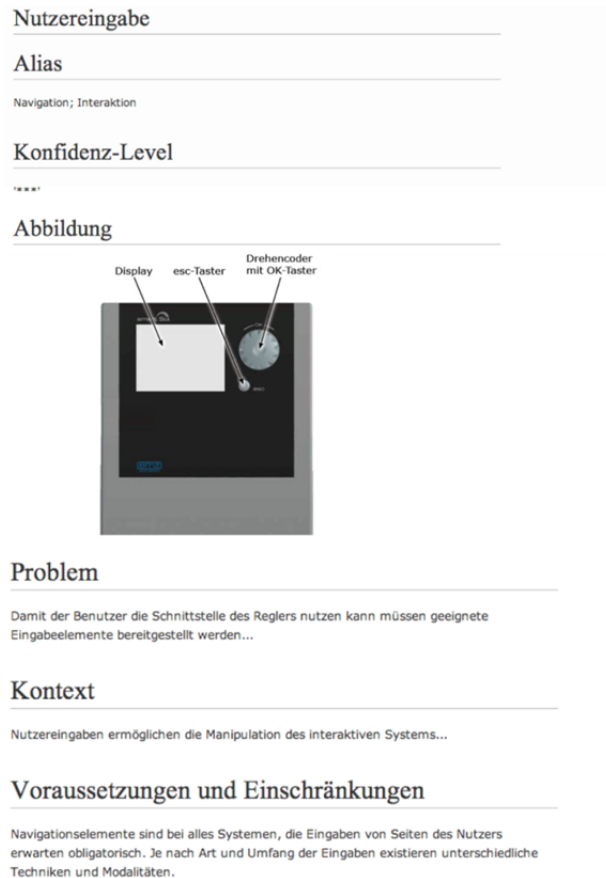


Abbildung 5: Überarbeitetes Design der Nutzerinteraktion.

Die durchgeführten Usability-Tests haben gezeigt, dass die Verwendung von Tasten in diesem speziellen Nutzungskontext oftmals problematisch ist. Dies schließt jedoch nicht aus, dass dieses Design in einem anderen Kontext (z. B. mit anderen Nutzergruppen) eine gute Lösung für die gleiche Problemstellung – der Gestaltung der Nutzerinteraktion – darstellt. So erscheint es sinnvoll auch diese Gestaltungslösung für eine mögliche spätere Wiederverwendung (*Design Reuse*) aufzubereiten und zu archivieren. Patterns erlauben die strukturierte Erfassung von Gestaltungslösungen, deren Begründungen und spezielle Anwendungskontexte (vgl. Abbildung 3). Zudem können Ergebnisse aus Usability-Studien, Verweise auf weitere Patterns oder Patternsammlungen sowie Literaturangaben in den Design-Patterns erfasst,

und so ein zusätzlicher Mehrwert geschaffen werden. Das Item *Voraussetzungen und Einschränkungen* definiert den Kontext in dem eine Gestaltungslösung angewendet werden kann bzw. in welchen Fällen eher davon abzuraten ist. Abbildung 6 stellt eine verkürzte Form des finalen Design-Patterns für die *Nutzereingabe* inklusive allen Attributen dar.



Zusammenfassung

Zur optimalen Bedienung des Gerätes stellt die Schnittstelle geeignete Funktionen und Hardware-Elemente (Buttons, LEDs, Display) für die Interaktion bereit.

Lösung

Für die Interaktion werden folgende Funktionen benötigt: Navigation in der Menüstruktur (auf und ab), Eingabe von Werten, die Möglichkeit einen Schritt rückgängig zu machen und die Möglichkeit eine Auswahl oder eine Eingabe zu bestätigen...

Begründung

Clickwheel: Drehbewegungen eignen sich für schnelles Navigieren und schnelles Ändern von Werten mit großem Wertebereich. Zudem eignet sich ein Drehencoder aus ergonomischer Sicht für die vertikale Betätigung (Wandmontage). Die Eignung dieser Eingabemodalität wurde durch Benutzertests validiert...

Beispiel



Verwandte Patterns

...

Abbildung 6: Design-Pattern für die „Nutzereingabe“.

4.3 Nachhaltige Entwicklung durch Wiederverwendung von Gestaltungslösungen

Alle während des Entwicklungs- und Designprozesses ausgearbeiteten Designlösungen werden strukturiert in Patterns aufbereitet, dokumentiert und so für die Wiederverwendbarkeit vorbereitet. Auf diese Weise können unterschiedliche Lösungsansätze für dasselbe Designproblem festgehalten und archiviert werden. Ein einfacher und effizienter Zugang für Entwickler und Designer zu den unterschiedlichen Stadien der Gestaltungsmuster ist für eine potenzielle Wiederverwendung wichtig. Im Rahmen des hier vorgestellten Projekts hat sich der Einsatz eines Wikis als adäquater Ansatz erwiesen. Im Wiki werden alle Patterns in Form einer Pattern-Bibliothek zentral verwaltet und über das Intranet des Unternehmens intern verfügbar gemacht. Mit Hilfe einer Versionierung lassen sich alle evolutionären Designlösungen im Entwicklungsprozess nachvollziehen. Die Gestaltungsentscheidungen jedes Stadiums eines Patterns können auf diese Weise für die Entwicklung weiterer Produkte wiederverwendet werden. Für jedes Designproblem liegen unterschiedliche (getestete) Lösungen vor, die sich je nach Kontext besser oder schlechter für weitere Entwicklungen eignen können. Konkret bedeutet dies, dass bei wiederholtem Auftreten eines Designproblems das Rad nicht jedes Mal neu erfunden werden muss. So können zeitliche und finanzielle Aufwände

reduziert und Produkte nachhaltig(er) entwickelt werden. Das Verwenden eines Wikis erlaubt zudem die vorliegenden Gestaltungsmuster stetig weiterzuentwickeln, neue Erkenntnisse (aus neuen Produktentwicklungen) einzuarbeiten und auf diese Weise dokumentiertes Designwissen kontinuierlich zu erweitern und zu verbessern. Grundsätzlich ist dabei auch die Möglichkeit gegeben, eigene Patternsammlungen mit den aus der Literatur bekannten Patternbibliotheken (insbesondere nach Tidwell und van Welie) zu vernetzen.

4.4 Prototyping und Design-Patterns: Aufwand und Nutzen

Gerade im Vergleich zu nicht oder nur informell annotierten Prototypen bedeutet die Dokumentation von Designwissen in Form von Patterns zunächst zusätzlichen Erstellungsaufwand, der sich aber bei komplexeren Fragestellungen (z. B. *Nutzerinteraktion* bei *Embedded Systems*) und auf Wiederverwendbarkeit ausgelegte Designlösungen (Produktlinienentwicklung) schnell bezahlt macht. Durch die vordefinierte Struktur von Patterns wird der Designer gezwungen, sich eingehender mit dem vorliegenden Problem zu beschäftigen. Wenn der Kern des Problems nicht in ein bis zwei Sätzen formuliert werden kann, wurde das Gesamtproblem, nach Alexanders (1977) Verständnis, noch nicht hinlänglich verstanden. Design-Patterns dienen neben der Dokumentation auch als Kommunikationsmedium, und sind aus ihrem Selbstverständnis heraus so formuliert, dass sie als *lingua franca* (Alexander 1977) für die am Entwicklungsprozess beteiligten Parteien (z. B. Entwickler, Designer, Projektmanager) fungieren und so die Verständigung erleichtern können. Eine Sammlung von Patterns wird idealerweise zu einer konsistenten Pattern-Sprache zusammengefasst, welche dann durch Referenzieren der Gestaltungsmuster aufeinander, und die Definition hierarchische Beziehungen zueinander, erweitert werden kann.

Im hier vorgestellten Projekt (Design-Lösungen für Regel- und Steuerungsgeräte in der Haustechnik) konnten bereits zwei weitere Produkte auf Basis der generischen Design-Patterns entwickelt werden. Neu hinzugewonnene Erkenntnisse wurden in die Patterns übernommen und stehen somit für künftige Entwicklungen zur Verfügung. Nicht zuletzt können auf diesem Wege Zukunftssicherheit und Nachhaltigkeit von Produkten gefördert werden.

5 Fazit und Ausblick

Prototypen zur frühzeitigen Erprobung von Designideen sind ein wichtiges Hilfsmittel in benutzerzentrierten Designprozessen. Dabei muss von Fall zu Fall entschieden werden, ob die formlose, unstrukturierte Dokumentation des Designwissens in Form von Prototypen ausreichend ist. Vor allem im Kontext komplexer Designprozesse und bei langfristig weiterzuentwickelnden Produktlinien kann es lohnenswert sein, auf eine nachhaltige Dokumentation des durch Prototypen erfolgreich evaluierten Designwissens zurückzugreifen. Patterns eignen sich durch ihre klare Struktur und verständliche Form zur Speicherung solchen Wissens in einem wiederverwendbaren Format. Der damit verbundene Aufwand muss jedoch stets mit Bezug auf den Projektkontext abgewogen werden. Für den Rahmen des hier beschriebenen Projekts zeigte sich die Verwendung von Patterns als adäquate und gute Lösung.

Design-Patterns stellen jedoch nur eine Möglichkeit dar Designwissen während des Prototyping-Prozesses zu dokumentieren. Welche Alternativen (z. B. annotierte, interaktive Prototypen), *best practices* und allgemeine Erfahrungen mit der Wiederverwendung von Designlösungen in der Designdenken-Community bestehen, soll im Workshop gemeinsam diskutiert werden.

Literaturverzeichnis

- Alexander, C., Ishikawa, S., Silverstein, M., & Jacobson, M. (1977). *A pattern language*. New York: Oxford University Press.
- Alexander, C. (1979). *The Timeless Way of Building* (Later prin.). Oxford University Press.
- Böckle, G., Knauber, P., Pohl, K., & Schmid, K. (Hrsg.) (2004). *Software-Produktlinien: Methoden, Einführung und Praxis*. dpunkt-Verlag.
- Böhm, P., Schneidermeier, T., & Wolff, C. (2011a). Customized Usability Engineering for a Solar Control Unit: Adapting Traditional Methods to Domain and Project Constraints. In A. Marcus (Ed.), *Design, User Experience, and Usability. Theory, Methods, Tools and Practice. Proceedings, Part II* (pp. 109–117). Springer.
- Böhm, P., Schneidermeier, T., & Wolff, C. (2011b). Smart Sol – Bringing User Experience to Facility Management : Designing the User Interaction of a Solar Control Unit. In A. Marcus (Ed.), *Design, User Experience, and Usability. Theory, Methods, Tools and Practice. Proceedings, Part II* (pp. 187–196). Springer.
- Borchers, J. (2001). *A pattern approach to interaction design*. Chichester [u.a.]: Wiley.
- Burghardt, M., Heckner, M., Kattenbeck, M., Schneidermeier, T., & Wolff, C. (2011). Design Thinking = Human-Centered Design? In M. Eibl (Ed.), *Workshopband Mensch & Computer 2011* (pp. 363–368).
- DIN EN ISO 9241-210. (2010). *Ergonomics of human-system interaction – Part 210: human-centred design process for interactive systems*. Beuth, Berlin.
- Fincher, S., Finlay, J., Greene, S., Jones, L., Matchen, P., Thomas, J., & Molina, P. J. (2003). Perspectives on HCI patterns: concepts and tools (introducing PLML). *CHI'03 extended abstracts on Human factors in computing systems* (pp. 1044–1045). ACM.
- Gamma, E. (1995). *Design patterns: elements of reusable object-oriented software*. Annals of Physics (Vol. 54). Addison-Wesley Professional.
- Sivaloganathan, S. and Shahin, T.M.M. (1999). Design Reuse: An Overview. In *Proceedings of the Institute of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 213(7), 641–654.
- Tidwell, J. (2011). *Designing Interfaces*. O'Reilly Media.
- Turban, B., Kucera, M., Tsakpinis, A., and Wolff, C. (2009). Bridging the Requirements to Design Traceability Gap. In R. E. D. Seepold and N. Martínez Madrid (Eds.), *Intelligent Technical Systems*. Berlin et al.: Springer, 275-288.
- Welie, M., v. (2008). *Patterns in Interaction Design*. www.welie.com. Zuletzt aufgerufen am 15. Juli 2012.

Kontaktinformationen

Tim Schneidermeier

Manuel Burghardt

Christian Wolff

Lehrstuhl für Medieninformatik

Universität Regensburg

93040 Regensburg

E-Mail: tim.schneidermeier@sprachlit.uni-regensburg.de

E-Mail: manuel.burghardt@sprachlit.uni-regensburg.de