

Agile UX – Wege zur agilen nutzerzentrierten Entwicklung

Hartmut Obendorf
C1 WPS
Vogt-Kölln-Str. 30
22527 Hamburg
hartmut@obendorf.de

Rainer Gibbert
XING AG
Gänsemarkt 43
20354 Hamburg
rainer.gibbert@xing.com

Inken Petersen
XING AG
Gänsemarkt 43
20354 Hamburg
inken.petersen@xing.com

Thomas Memmel
Zühlke Engineering AG
Wiesenstrasse 10a
CH-8952 Schlieren (Zürich)
thomas.memmel@zuehlke.com

Abstract

Methoden und Techniken der agilen Softwareentwicklung sind inzwischen so stark verbreitet, dass sie aus der Praxis nicht mehr wegzudenken sind. Obwohl es bereits erste Erfahrungen und Vorgehensmodelle gibt, haben Usability Engineering und User Experience (UX) Design in agilen Projekten aber weiterhin einen schweren Stand.

Obwohl sie selbst oft iterativ angewendet werden, werden sie vielfach als zu schwergewichtig empfunden. Der Anspruch von Usability Experten, möglichst früh im Prozess ein stimmiges und vollständiges Bild des User Interface zu erarbeiten, passt auf den ersten Blick nicht zum Charakter agiler Softwareentwicklungsprojekte.

Die Autoren stellen in diesem Artikel und in ihrem Tutorial diese Schwierigkeiten vor, zeigen Ansätze zu ihrer Lösung auf und skizzieren die zu bearbeitende Themenfelder.

Keywords

Agile UX, Agile Methoden, Usability Engineering, User-Centered Design, Scrum

1.0 Einleitung

Für agile Entwicklungsmethoden stehen explizit die am Prozess beteiligten Personen sowie die Software selbst im Vordergrund. Durch den propagierten informellen Charakter des Ansatzes und den hohen Stellenwert von Kommunikation entsteht in der Wahrnehmung der Durchführenden ein deutlicher Unterschied zu klassischen Analyse- und Prozessmodellen; agile Methoden werden als vergleichsweise leichtgewichtig empfunden.

Agile Methoden haben damit grundlegende Veränderungen in der Wahrnehmung von Softwareentwicklung ausgelöst und die Eignung streng sequentieller Arbeitsschritte für die Entwicklung von Software in Frage gestellt. Design und Umsetzung von Software werden zunehmend als iterativ inkrementelle Prozesse verstanden. Eine Zuwendung zu agilen Methoden ist daher meist mit der Erkenntnis verbunden, dass interaktive Softwareprodukte selten vollständig im Voraus spezifiziert werden können und dass unterschiedliche Aspekte einer Software einen unterschiedlich großen Nutzen für die Anwender haben. Die Idee von evolutionärem Prototyping, d.h.

des Einsatzes und der schrittweisen Erweiterung der zu entwickelnden Software, findet sich daher in den meisten agilen Entwicklungsmethoden wieder.

Drei Aspekte sind den wichtigsten agilen Methoden, darunter in erster Linie XP und Scrum, gemein:

- Der Wunsch, Projekte anders zu organisieren als mit traditionellen Entwicklungsmethoden (wie z.B. Wasserfall-Modell, Rational Unified Process)
- Der Rückbezug auf die gemeinsame Absichtserklärung des agilen Manifests und damit eine explizite Wertschätzung des informellen Austausches sowie eine Verlagerung von der Ausführung von Vorgaben zur gemeinsamen Festlegung von Möglichkeiten
- Der Versuch, die Schwierigkeiten des traditionellen Projektmanagements (z.B. Zeitplanung zur Fertigstellung einer Menge von Funktionen) durch Time-Boxing (d.h., welche Funktionalität kann in einer gegebenen Zeit fertig gestellt werden) zu ersetzen.

2.0 Dissonanzen: Agile Softwareentwicklung und UX

Ebenso wie Usability-Engineering und UX Design zielt agile Softwareentwicklung darauf ab, ein Produkt zu entwickeln, das möglichst gut auf die Bedürfnisse der Kunden bzw. Anwender abgestimmt ist (vgl. z.B. Larman 2003). Diese Ähnlichkeit der Ziele legt die Kombination von Usability-Methoden mit agilen Prozessen nahe.

Obwohl sich in der jeweils anderen Methode jedoch oft eine ideale Ergänzung findet, stehen einem gemeinsamen Einsatz dennoch Hindernisse im Weg: zuvorderst die Betonung eines strikt iterativen Vorgehens bei XP und Scrum, bis hin zum expliziten Ausschluss eines „Big-Upfront-Design“. Zwar sind auch viele User-Centered Design-Methoden iterativ angelegt, sie gehen aber oft davon aus, dass sich diese Iterativität nur auf die Konzept- und Designentwicklung bezieht – ist diese abgeschlossen, werden „fertige“ Vorgaben an die Softwareentwicklung übergeben. Aus Sicht agiler Prozesse ist darin genau das unerwünschte Vordefinieren eines vollständigen Designs zu verstehen, das keine Änderungen mehr zulässt (Big Upfront Design). Umgekehrt bestehen

seitens Usability Experten große Bedenken, dass die Konsistenz und das Benutzerempfinden ein Produkt "aus einem Guss" zu bedienen, ohne eine gewisse Vollständigkeit des User Interface-Entwurfs kaum erreichbar sind.

3.0 Harmoniebildung: Wege zu einer Vereinbarkeit von UX und Agile

Agile Methoden haben Defizite in der Einbeziehung von Nutzern und der Analyse und Spezifikation des Nutzungskontextes, die von User-Centered Design Methoden geschlossen werden können; so funktioniert etwas die Einbindung eines Nutzers in das Entwicklungsteam (customer on site) selten wie erhofft und es existieren keine Mechanismen, um etwa Konsistenz in der Interaktion abzusichern. Ebenso kann umgekehrt die iterative Entwicklung helfen, Fehler im Design frühzeitig zu erkennen und zu beheben (vgl. Obendorf & Finck 2007; Obendorf & Finck 2008).

Trotz der angesprochenen Herausforderungen, die in konkreten Projekten auftreten können, ist daher ein verstärkter Einsatz von Kombinationen von User-Centered Design und agiler Softwareentwicklung zu beobachten. Wir beschränken uns hier darauf, verschiedene Möglichkeiten anzudeuten, wie die Synchronisation zwischen UX und Softwareentwicklung stattfinden kann:

3.1 Verzahnung UX und Agile: Sprint Zero und Planung für Sprint 1+n

Eine der ersten weit verbreiteten Möglichkeiten, User-Centered Design Methoden mit agilen Techniken – zuvor erst Scrum – zu kombinieren, ist die parallele Durchführung von Sprints für UX und Softwareentwicklung (vgl. Baker et al. 2004, Patton 2008). In einem vorgelagerten Sprint Zero (Iteration 0) werden dabei von UX-Seite aus Vorgaben für die Gestaltung der Anwendung erarbeitet.

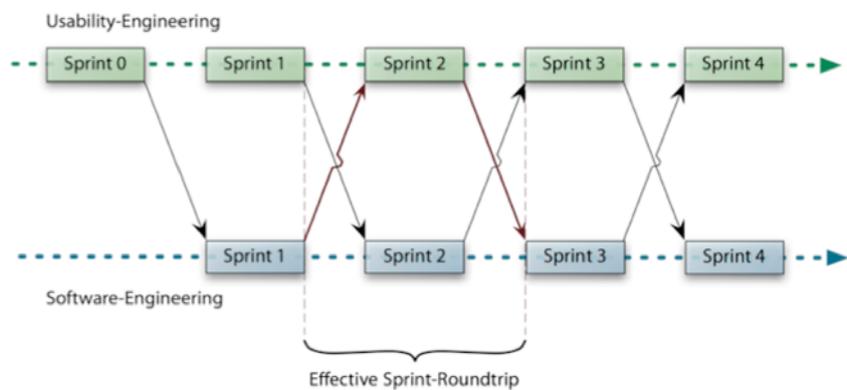


Abb. 1: Verzahnung von UX und Agiler Softwareentwicklung

Während die Entwickler ausgewählte Bestandteile der Software implementieren (Iteration n), arbeitet das UX-Team für folgende Iterationen voraus. Auf diese Weise können etwa wichtige Benutzerstudien für die übernächste Iteration (n+2) durchgeführt werden. Für die nächste Iteration (n+1) können von den UX-Experten bereits konkrete User Interface Elemente gestaltet und zum Beispiel interaktive Prototypen entwickelt werden. Diese dienen den Entwicklern in der nächsten Iteration als interaktive Spezifikation. So arbeiten beide Disziplinen in getrennten, aber kooperierenden Teams mit klaren Verantwortlichkeiten. Definierte Abstimmungspunkte können vermeiden, dass es durch die Integration des UX-Teams zu Wartezeiten bei der Entwicklung kommt.

Selbstverständlich geben die UX-Experten den Entwicklern nicht nur neue Konzepte und Designs vor, sondern sie erhalten auch programmierte Resultate (z.B. aus Iteration n-1) für Usability Tests zurück (Jocham & Memmel 2009). Eine Usability Testmethode, die gut zu einem agilen Vorgehen passt, ist zum Beispiel RITE (Rapid Iterative Testing and Evaluation). RITE (Medlock et al. 2000) sieht vor, dass gefundene Usability Probleme vor Durchführung des nächsten Tests behoben werden. Im Gegensatz zu

formaleren Testmethoden testen also nicht alle eingeladenen Probanden exakt die gleiche Software, sondern diese wird von Test zu Test bereits verbessert. RITE bietet sich an, wenn Usability Experten in einem agilen Umfeld eng mit den Entwicklern zusammenarbeiten und die Ressourcen für Usability Tests eingeschränkt sind.

3.2 Integration UX und Agile: Verteilte Rollen in einem Team

Eine andere Möglichkeit der Kombination von Usability mit agilen Methoden sieht vor, UX-Experten direkt mit den Entwicklern in einem Team zu integrieren. Dabei werden zunächst eine Produktvision und technische Prototypen gemeinsam entwickelt. Schließlich wird iterativ und inkrementell mit Hilfe von Low-Fidelity Prototypen die Spezifikation des Produktes erstellt und mit Benutzern evaluiert. Dann erfolgt in enger Zusammenarbeit mit den UX-Experten die Implementierung.

Durch die enge Integration des UX-Experten in das Scrum Team kann dieser jederzeit an allen bearbeiteten Features oder User Stories mitarbeiten – ohne die Notwendigkeit für eine ausführliche Dokumentation. User Stories werden dafür mit User Interface Spezifikationen und Prototyp-Beschreibungen kombiniert. Dadurch sind die User Stories nicht mehr nur ein „Versprechen für

Kommunikation“, sondern auch eine inkrementell wachsende, beispielhafte Spezifikation des Produktes (Obendorf & Finck 2007; Obendorf & Finck 2008).

3.3 Entkopplung: Product Discovery und Product Execution

Eine dritte Möglichkeit besteht in einer Aufteilung der Produkt- oder Software-Entwicklung in zwei Phasen: eine Phase der Product Discovery und einer der Product Execution (Cagan, 2008).

In der Product Discovery wird durch eine ebenfalls iterative Vorgehensweise versucht, ein ideales minimales Produkt zu identifizieren. Beteiligt in der Discovery Phase sind dabei ein Dreier-Team aus Product Owner, UX-Experte und Entwickler. Das Produkt wird hierbei allerdings nicht langwierig dokumentiert und spezifiziert, sondern in Form eines High-Fidelity-Prototypen umgesetzt, der einerseits der Kommunikation im Team und mit Stakeholdern dient, andererseits in kurzen Usability Tests evaluiert und iterativ weiterentwickelt wird. Ziel der

Product Discovery ist es insgesamt, die Nutzer und deren Bedürfnisse kennenzulernen und ein Produkt zu identifizieren, welches nutzbringend (valuable), benutzbar (usable) und technische machbar (feasible) ist – und dabei minimalen Aufwand verursacht und möglichst wenig Funktionalität umfasst.

Wurde dieses minimal nutzbringende Produkt identifiziert, wird der Prototyp in einen Product Backlog überführt und anschließend in die Product Execution übergeben, wo er mit agilen Methoden wie Scrum umgesetzt wird. Wichtig hierbei ist jedoch, dass am Konzept bzw. Prototypen nichts mehr verändert wird, sodass die Implementierung ungehindert von statten gehen kann.

Während das Produkt in der Product Execution umgesetzt wird, kann sich das Team aus Product Owner, UX-Experte und Entwickler wieder in die nächste Iteration der Product Discovery begeben, in der neue Ideen zur Weiterentwicklung des Produkts entwickelt werden können.

3.4 Welche Kombination für welchen Zweck?

Über den richtigen Modus der Zusammenarbeit im Sinne von Agile UX bestimmen verschiedene Rahmenbedingungen, wie zum Beispiel:

- Personalressourcen: in welchem Ausmaß stehen UX-Experten für das Projekt zur Verfügung – ist eine kontinuierliche Begleitung oder sogar eine Mitarbeit im Team möglich? Können umgekehrt die Entwickler bei der Sprint-Planung Aufwände für UX Spikes oder Anpassungen aufgrund von Testergebnissen reservieren?
- Teamzusammensetzung: wie fügen sich Mitarbeiter aus dem eigenen Unternehmen mit solchen von hinzugezogenem Dienstleistern zu einem Team zusammen? Sind Anwender als On-Site User Teil des Teams, damit die Benutzerbenutzerperspektive im Sinne von UX einfließen kann?

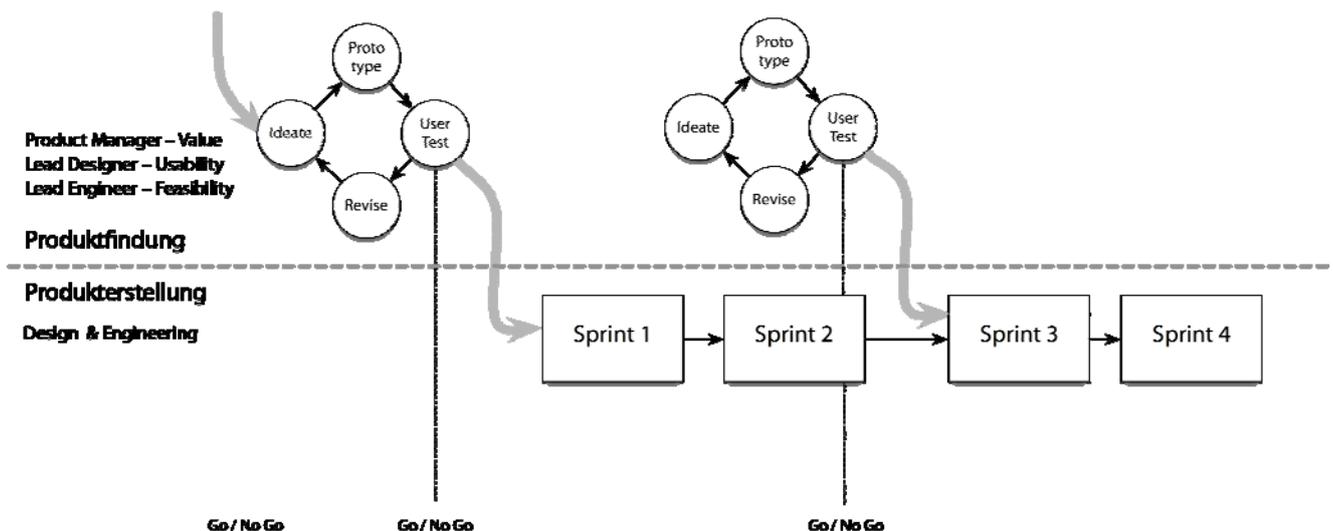


Abb. 2: Entkopplung von Entwicklung und Design durch die Einführung einer Phase für Product Discovery

- Kollokation: die Möglichkeit, ohne Dokumentenaustausch zusammenzuarbeiten, besteht in erster Linie dann, wenn die Teammitglieder am gleichen Ort arbeiten und dabei über gemeinsame Räumlichkeiten verfügen.

4.0 Tutorial im UPA-Track 2010

Auf Basis von (Zimmermann et al. 2009) setzen wir die Auseinandersetzung von Usability Experten mit agilen Methoden durch ein wie folgt aufgebautes Tutorial fort:

1. Einführung in die agile Softwareentwicklung: Nachzeichnung agiler Grundprinzipien, Vorstellung des Agile Manifesto und historische Betrachtung. Besprechung wichtiger Methoden und Begriffe. Darstellung der Chancen und Schwierigkeiten einer Kombination von UX und agiler Softwareentwicklung.
2. Projektbeispiele: Präsentation und Diskussion zu verschiedenen Formen von Agile UX Projekten und der entsprechenden Rollenverteilung im Team. Vorstellung von abgeleiteten Best Practices. Erleben von Agile UX: Übungen, zum Beispiel zu den Auswirkungen der Prinzipien agiler Entwicklung auf das Projekt, und zur Definition inkrementeller Anforderungen

3. Open Spaces: Offene Diskussion von Themen, für die bei den Teilnehmenden Informations- und Klärungsbedarf besteht

5.0 Arbeitskreis Agile

Der Arbeitskreis (AK) Agile sucht den Austausch von UX-Experten und Softwareentwicklern. Nur mit der Mitwirkung erfahrener Experten beider Disziplinen und durch einen intensiven Austausch über gemachte Erfahrungen kann das Potential von Agile UX erörtert werden. Der AK Agile soll nicht nur Raum für einen regen Austausch bieten, sondern die Ergebnisse kanalisieren und veröffentlichen. Ein erster Schritt zur Mitwirkung im AK Agile ist die Anmeldung in unserer XING Gruppe (www.xing.com/net/agileusability). Wir freuen uns über die Beiträge vieler Experten aus Forschung und Praxis.

6.0 Literatur

Baker, L.; Beyer, H.; Holtzblatt, K. (2004): An Agile Customer-Centered Method: Rapid Contextual Design. In: XP Agile Universe 2004 Proceedings.

Cagan, M. (2008): Inspired: How To Create Products Customers Love. SVPG Press

Jocham, R.; Memmel, T. (2009): Agile User Experience. In: Computerworld – die Schweizer IT-Plattform für IT-Professionals. IDG Verlag, erschienen 21.11.2009: 24-25

Larman, C. (2003): Agile and Iterative Development: A Manager's Guide. Addison-Wesley.

Medlock, M.C., Wixon, D., McGee, M., & Welsh, D. (2005): The Rapid Iterative Test and Evaluation Method: Better Products in Less Time. In: Bias, G., & Mayhew, D. (Hrsg.), Cost Justifying Usability. San Francisco, Morgan Kaufmann: 489-517

Obendorf, H.; Finck, M. (2008): Scenario-based usability engineering techniques in agile development processes. CHI Extended Abstracts 2008: 2159-2166

Obendorf, H.; Finck, M. (2007): Szenariotechniken & Agile Softwareentwicklung. Mensch & Computer 07: 19-28

Patton, J. (2007): Twelve emerging best practices for adding UX work to Agile development. Online unter http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html

Zimmermann, D.; Obendorf, H.; Memmel, T.; Nufer, S. (2009): Agile User-Centered Design Tutorial. In: Brau, H., Diefenbach, S., Hassenzahl, M., Kohler, K., Koller, F., Peissner, M., Petrovic, K., Thielsch, M., Ullrich, D. & Zimmermann, D. (Hrsg.). Usability Professionals 2009. Fraunhofer Verlag, Stuttgart. Online unter <http://www.slideshare.net/dirkzimmermann/agile-ucd-up09-tutorial-2135008>

Branchenreport Usability