

Integration von User Centred Design Aktivitäten in Agile Softwareentwicklung

Karsten Nebe
Universität Paderborn, C-LAB
Fürstenallee 11
33102 Paderborn
karsten.nebe@c-lab.de
www.c-lab.de/usability

Markus Düchting
Siemens IT Solutions, C-LAB
Fürstenallee 11
33102 Paderborn
markus.duechting@c-lab.de

Dirk Zimmermann
T-Mobile Deutschland
Usability Engineering Services
Landgrabenweg 151
53227 Bonn
dirk.zimmermann@t-mobile.de

Abstract

Viele existierende Ansätze zur Integration von Usability Engineering Aktivitäten, Artefakten und Rollen in das Software Engineering basieren auf den Grundcharakteristiken klassischer Softwareentwicklungsmodelle: sie beziehen sich in der Regel auf meist umfangreiche Dokumentationen und Artefakte und sind

ebenfalls phasenorientiert. Aufgrund dieser Gemeinsamkeit bieten diese Ansätze Potential für eine erfolgreiche Integration von SE und UE. In diesem Paper wird die Frage behandelt, ob sich solche Ansätze und Aspekte des Usability Engineering bereits im agilen Software Engineering wieder finden

bzw. sich diese übertragen lassen.

Keywords

User Centred Design, Agiles Software-Engineering, Integration, Anforderungen

1.0 Einleitung

Die Disziplin des Software Engineering (SE) zielt unter anderem darauf, systematische und planbare Vorgehensweisen für die Entwicklung von Software bereitzustellen. Durch den Einsatz existierender SE Modelle wird angestrebt, Kosten, Entwicklungszeit und Qualität der Ergebnisse über den gesamten Entwicklungsprozess hinweg abschätzen, steuern und kontrollieren zu können (Glinz 1999). Meist orientieren sich solche Modelle an festen Prozessphasen, den daraus resultierenden Ergebnissen (Artefakten) und entsprechenden Meilensteinen. Bestes Beispiel dafür ist das klassische Phasenmodell, auch Wasserfall-Modell genannt (Boehm 1988). Dies stellt insbesondere für das Management eine Methode dar, den Überblick und die Kontrolle über das Projekt zu behalten. Phasenorientierte und an Artefakten orientierte SE Modelle können als „klassische“ SE Modelle bezeichnet werden und haben sich in der Software-Entwicklung entsprechend etabliert.

Besonders in den letzten Jahren lässt sich ein neuer Trend im Bereich des SE beobachten: Weggehend von rigide de-

finierten Prozessen, hin zu leichtgewichtigeren Methoden und Vorgehensweisen, so genannten „agilen Vorgehensweisen“ oder auch „agilen Modellen“. Die Motivation hinter den agilen Modellen unterscheidet sich von klassischen Modellen in vielerlei Hinsicht. Es gibt keine monolithischen Phasen, es wird weniger Wert auf umfangreiche Dokumentationen und Artefakte gelegt, als vielmehr auf Kommunikation und die Interaktion aller Projektbeteiligten (Schwaber 2004). Die wichtigste Motivation für die agile Softwareentwicklung ist es jedoch anzuerkennen, dass die Entwicklung von komplexen Systemen eine hoch kreative und schöpferische Aufgabe ist (Schwaber & Beedle 2002). Agile Modelle gehen auf besondere Weise mit sich ändernden Anforderungen um, da sie von Anfang an davon ausgehen, dass es unmöglich ist alle Anforderungen an ein komplexes System im Vorfeld vollständig zu erfassen und dementsprechend alle Aktivitäten genau zu planen. Dies ist ein bekanntes Problem im Bereich der Softwareentwicklung, welches eine der häufig genannten Ursachen in gescheiterten Projekten ist (Larman 2004). Agile Modelle greifen üblicherweise ein iterativ-

inkrementelles Vorgehen auf, wobei das zu entwickelnde System in kurzen Zeiträumen jeweils um ein kleines Maß an Funktionalität wächst (Larman 2004).

Das Usability Engineering (UE) stellt, ähnlich dem Software Engineering, eine Reihe von Modellen und Methoden zur Sicherstellung der Nutzerorientierung zur Verfügung. In der Praxis orientiert sich der Einsatz dieser Methoden und den verbundenen Aktivitäten meist stark an bereits existierenden SE Modellen und dem zugehörigen Prozess (Nebe et al. 2006). Die UE Aktivitäten finden häufig nebenläufig dazu statt und demzufolge ist nicht immer eine Integration von Usability Engineering und Software Engineering gegeben. Dennoch gibt es einige Ansätze die dies in Form von integrierten Modellen versuchen, wie beispielsweise Mayhew (1999), Cooper (2003) oder IBM (1996).

Betrachtet man diese Ansätze im Detail so lässt sich feststellen, dass sie mit einigen Grundcharakteristiken aus dem klassischen SE korrespondieren. Sie beziehen sich in der Regel auf meist umfangreiche Dokumentationen und Artefakte und sind ebenfalls phasenorientiert. Aufgrund dieser Gemeinsamkeit

bieten diese Ansätze Potential für eine Integration von SE und UE.

Was im klassischen SE somit in Teilen gut funktioniert, gestaltet sich im agilen SE entsprechend schwieriger. Betrachtet man die neueren, agilen Ansätze, so wird deutlich dass diese nicht auf diesen Grundcharakteristiken und Gemeinsamkeiten aufbauen. Die Leichtgewichtigkeit der Prozesse, das Auflösen der Phasen und das Lossagen von umfangreichen Dokumentationen stehen hier im Vordergrund. Daraus resultierend stellt sich die Frage, inwiefern agile SE Modelle bereits Vorgehensweisen des UE aufgreifen und wie fehlende Prozessschritte hinreichend implementiert werden können.

2.0 Vorgehen

Mit der Fragestellung, inwieweit agile Modelle bereits Aspekte des Usability Engineering berücksichtigen, haben sich die Autoren bereits im Rahmen einer vorhergehenden Betrachtung beschäftigt (Duechting et. al. 2007).

In diesem Paper sollen die Ergebnisse dieser Untersuchung analysiert, und abgeleitete Optimierungs-Ansätze auf Effektivität und Sinnhaftigkeit überprüft werden. Die zugrunde liegende Analyse wurde exemplarisch für zwei agile Modelle, Scrum (Schwaber & Beedle 2002) und das Extreme Programming (Beck 2000), durchgeführt.

Ziel der Analyse war hierbei nicht, die beiden Modelle miteinander zu vergleichen und die Ergebnisse gegenüber zu stellen, sondern vielmehr die Erkenntnisse aus den Einzelbetrachtungen auf das agile Vorgehen im Allgemeinen zu übertragen. Dies wird insbesondere dadurch untermauert, dass der Fokus der beiden Modelle unterschiedlich ist:

Scrum ist vordergründig als eine Methodologie von Projekt Management Praktiken zu sehen, wobei das Extreme Pro-

gramming als eine Menge von Entwicklungspraktiken zu verstehen ist. Sie wirken nicht gegensätzlich zueinander sondern können sich in der Praxis sehr gut ergänzen.

Die Analyse bezieht sich auf die Modelle in Reinform, wie sie in der Literatur beschrieben sind.

Da viele der existierenden Integrationsansätze zwischen SE und UE auf den Grundcharakteristiken der klassischen Softwareentwicklung aufbauen war es wichtig eine gemeinsame Basis zwischen dem klassischen und dem agilen SE zu finden, um eine entsprechende Bewertung durchführen zu können. Ein wesentlicher Unterschied zwischen dem Klassischen und dem Agilen besteht in erster Linie im Vorgehen (phasenorientiert vs. evolutionär/inkrementell) aber insbesondere auch der Dokumentation (umfassend vs. leichtgewichtig). Schwergewichtige Dokumente, wie z.B. ein Lasten- oder Pflichtenheft im Phasenmodell sind nicht angemessen für agile SE Modelle. Um hier eine Schnittstelle zwischen

den Aktivitäten des UE und dem SE zu schaffen müssen die Ergebnisse granularer erfasst und dokumentiert werden, um den Vorgehensweisen des Agilen zu entsprechen. Eine geeignete Basis sind Anforderungen - und dies nicht nur als Parallele zwischen dem klassischen und dem agilen SE, sondern gleichermaßen zwischen dem SE und UE.

Anforderungen spielen in der Entwicklung eine große Rolle und haben eine besondere Relevanz während des gesamten Software-Lebenszyklus. Anforderungen, die beispielsweise in frühen Phasen der Analyse und Spezifikation erfasst worden sind, leben bis zu abschließenden Tests gegen eben diese. Durch die Definition und durchgängige Umsetzung von Anforderungen wird aus Sicht des SE sichergestellt, dass die entwickelte Lösung z.B. vollständig, korrekt und fehlerfrei ist. Aus Sicht des UE können Anforderungen dazu dienen die konkreten Bedürfnisse und Ziele der Nutzer abzubilden, aber auch um übergeordnete Usability-Kriterien zu berücksichtigen (bspw. Effizienz, Effektivität, Zufriedenheit, etc.).

Die übergreifenden „Usability Requirements“ werden in der „Context of Use“-Phase erfasst. Sie lassen sich beispielsweise aus Persona-Beschreibungen ableiten und enthalten unter anderem Ziele und Motive der Nutzer-(gruppen), welche in Requirements überführt werden können. Diese Requirements haben Einfluss auf die Effizienz, Effektivität und Nutzerzufriedenheit.

In der „Specify Requirements“ Phase werden „Workflow Requirements“ erfasst, die beschreiben wie ein zukünftiges System den Workflow der Nutzer unterstützt. Dies können auch aus Use Cases oder konkreten Szenarien abgeleitet werden. Sie beschreiben Optionen, Informationen und die systemseitige Unterstützung, die Nutzer zum erfüllen ihrer Aufgaben benötigen.

Des Weiteren werden „User Interface Requirements“ in der „Produce Design Solutions“ Phase erhoben, die auf der Entwickelten Spezifikation eines Systems beruhen. Diese können unter Anderem aus Wireframes, Storyboards, Prototypen und Interfacebeschreibungen abgeleitet werden. Sie repräsentieren funktionale und Nicht-funktionale Anforderungen bzgl. der Erscheinung und Interaktion eines zu entwickelnden Systems.

Kasten 1: [Zimmermann und Grötzbach, 2007] unterscheiden zwischen drei Arten von Requirements. Die Typen sind aus den UCD Ergebnissen der ersten drei Phasen abgeleitet, welche im Modell der DIN EN ISO 13407 beschrieben werden.

Um diese Kontinuität sicherzustellen haben Zimmermann & Grötzbach drei unterschiedliche Arten von Requirements definiert, in einem Framework verankert und verschiedenen Phasen des Entwicklungsprozess zugeordnet (Zimmermann et. al. 2007). Das Framework ermöglicht eine nahtlose Einbeziehung von Usability-bezogenen Requirements in den Software Engineering Prozess, wodurch die Usability eines Produktes sichergestellt werden kann. Zimmermann & Grötzbach unterscheiden zwischen Requirements, die sich jeweils aus Ergebnissen der ersten drei Phasen der DIN EN ISO 13407 ableiten: Usability Requirements, Workflow Requirements und User Interface Requirements (s. Kasten 1).

Um das Framework für die Bewertung von agilen Modellen zu nutzen galt es, die hinter den Requirements stehenden Aktivitäten und Phasen aufzulösen und dafür entsprechende Kriterien zu definieren anhand derer überprüft werden konnte, inwieweit agile Modelle den daraus resultierenden Anforderungen des Frameworks gerecht werden. Für die beiden Modelle wurde anschließend überprüft wurde, inwieweit agile Modelle bereits Aktivitäten (und Artefakte) des UE berücksichtigen bzw. umsetzen.

Im Falle einer Nichterfüllung für ein Kriterium durch das jeweilige Modell wurden anschließend Empfehlungen abgeleitet, die in der Praxis eine Erfüllung sicherstellen können. Dabei wurde darauf geachtet, dass weder die grundsätzlichen Eigenschaften des agilen Ansatzes, noch die Spezifika der beiden konkreten Modelle verletzt werden. Die Empfehlungen beziehen sich auf den Einsatz durchzuführender Aktivitäten, einzusetzenden Artefakte und Rollen, wie nachfolgend beschrieben.

3.0 Analyseergebnisse & Empfehlungen

Als ein übergeordnetes Ergebnis der Analyse zeigte sich, dass die beiden agilen Modelle Defizite in Bezug auf die Realisierung von UE Aktivitäten aufweisen. Weder während der Analyse-Aktivitäten, noch bei der Evaluation werden Usability Requirements adäquat repräsentiert. Dies folgt zum einen daraus, dass keine Analyse-Aktivitäten vorgesehen sind, die vergleichbar mit einer Nutzungskontextanalyse sind. Zum anderen sind keine Maßnahmen wie z. B. abschließende Usability Tests vorgesehen, durch welche übergeordnete Usabilityziele und konkrete Anforderungen validiert werden könnten. Auf der Ebene von detaillierteren Anforderungen besitzen die agilen Modelle zwar gewisse Stärken, aber auch Potential für die weitere Adaption von Techniken aus dem Usability Engineering. Die beiden Modelle bieten die Möglichkeit Workflow Requirements bezüglich ihrer Vollständigkeit und Korrektheit zu überprüfen, allerdings ist aus einer UE Perspektive nicht sichergestellt, dass sie in vorangegangenen Phasen richtig erfasst und weiterverarbeitet werden. Ähnliches gilt auch für die User Interface Requirements.

Die folgenden Empfehlungen stellen aus der Analyse abgeleitete Vorschläge dar die helfen sollen, ein nutzerzentriertes Vorgehen im Agilen stärker zu etablieren.

- **Usability Engineers and UI-Designers:** Experten als permanenter Teil des Entwicklungsteams.
- **Preliminary Exploration & Analysis:** Vorgelagerte Analysephase um erste Einblicke in den Workflow und den Nutzungskontext zu erhalten.

- **Real Users:** Echte Nutzer anstatt Repräsentanten.
- **Workflow Requirements:** Zentrales Artefakt zur Dokumentation der Nutzer-Arbeitsabläufe.
- **Evaluation Activities:** Kontinuierliche Überprüfung von (Teil-) Ergebnissen.
- **Gain quick user/customer feedback:** On-Site Customer/User für schnelles Feedback heranziehen.
- **Close Collaboration:** Kontinuierlicher Austausch aller Beteiligten.

4.0 Evaluation der Anwendung in der Praxis

Um die Zuverlässigkeit der Empfehlungen zu überprüfen, wurden sie mit drei Experten aus der Praxis evaluiert. Jeder der Experten verfügt über mehrere Jahre Erfahrung im Bereich der agilen Softwareentwicklung (insbesondere Scrum) sowie der Umsetzung von Aktivitäten zur Erstellung von nutzerzentrierten Lösungen. In dem international tätigen Unternehmen wurden bereits Schritte eingeführt, die agilen Vorgehensweisen mit UE Aktivitäten zu untermauern, wie beispielsweise regelmäßigem Testing, Nutzer-Partizipation, usw.

Bei der Befragung wurden insbesondere die Nützlichkeit und die Praktikabilität jeder Empfehlung überprüft. Da im Kreise der befragten Personen auch solche waren, die sowohl im Bereich der agilen Softwareentwicklung als auch im Bereich des UCD tätig waren konnte ermittelt werden ob die Empfehlungen in gleicher oder ähnlicher Weise bereits Einsatz finden oder bereits gefunden haben. Darüber hinaus konnten, im Falle einer Anwendung, auch Informationen über die Effektivität entsprechender Maßnahmen gesammelt werden.

Neben allgemeinen Anmerkungen zu jeder Empfehlung wurde zusätzlich nach

Erfahrungen über Auswirkungen bzw. Nebeneffekte (positive & negative) gefragt, die durch die jeweilige Empfehlung ausgelöst werden können bzw. aufgetreten sind.

Übergeordnet zeigte die Befragung, dass die Empfehlungen die Praxis widerspiegeln. Die meisten der Empfehlungen wurden als z. T. sehr nützlich und praktikabel bewertet. Einige der Empfehlungen wurden bereits in beschriebener oder vergleichbarer Form in der Praxis angewandt.

Nachfolgend werden die Aussagen und Einschätzungen der Teilnehmer zu den jeweiligen Empfehlungen vorgestellt.

Die Empfehlung „**Usability Engineers and UI-Designers**“ als Bestandteil des Teams einzuführen wurde einheitlich als sehr sinnvoll und nützlich bewertet und hat in vergleichbarer Form bereits Einsatz in dem Unternehmen gefunden. Positive Seiteneffekte sind, dass Usability-Aspekte unmittelbar berücksichtigt werden können und direkt in die Anforderungsdefinition einfließen können. Durch die enge Zusammenarbeit von Usability Engineers mit den Designern und Entwicklern kann ein schneller Austausch von Informationen gewährleistet werden. Dies spiegelt sich positiv in der Gebrauchstauglichkeit der Lösung wieder. Insgesamt wurde die Empfehlung daher als sehr effektiv angesehen und bewertet.

Die Empfehlung zur „**Preliminary Exploration & Analysis**“, im Sinne einer vorgelagerten Analysephase, wurde überwiegend als nützlich angesehen. Die Anwendung in der Praxis hat sich in mehreren Fällen als geeignet erwiesen. Die Berücksichtigung dieser Empfehlung hilft den Analysten und Entwicklern, auf die relevanten, richtigen Anforderungen und Bedürfnisse der Kunden und Arbeitsabläufe der Nutzer zu fokussieren. Im Einsatz zeigte sich, dass der Nutzen aus solchen vorgelagerten Analyseakti-

vitäten wesentlich größer ist als der tatsächlich dafür investierte Aufwand, bzw. der Aufwand für späte Änderungen aufgrund nicht berücksichtigter oder fehlinterpretierter Anforderungen.

Die Frage ob die Empfehlung von „**Real Users**“ als Vertreter der Zielgruppe im Projekt Einsatz finden sollten wurde von den Experten in Abhängigkeit der Menge an Nutzer-Systeminteraktion im jeweiligen Projekt unterschiedlich bewertet. Dies spiegelte sich auch im praktischen Einsatz wider. Im Falle eines Einsatzes wurde dies jedoch als sehr effektiv bewertet. Das Feedback der Kunden und Nutzer zu Konzepten und Designs hat in der Praxis positiv dazu beigetragen, auf Fragen bezüglich der Umsetzung von Nutzer- und Aufgabenzielen zu fokussieren, anstatt über die generelle Notwendigkeit und der Umsetzung einzelner Anforderungen zu spekulieren.

Bezüglich der Empfehlung „**Workflow Requirements**“ als Artefakt zu Dokumentation der Nutzer-Arbeitsabläufe ist zu sagen, dass auch hier ein großer positiver Nutzen in der Praxis entsteht und auch so durch die Experten bewertet wurde. Allerdings liegt eine Schwierigkeit darin, angemessene Werkzeuge für die entsprechenden Aufgaben zu finden bzw. zu wählen. Der Einsatz von „Workflow Requirements“ in der Praxis ist laut Experten in Abhängigkeit vom Projekt zu wählen. Dabei muss sichergestellt sein, dass nicht nur einzelne Arbeitsabläufe dokumentiert und implementiert, sondern auch die Beziehungen zwischen den Arbeitsabläufen im konkreten Handlungskontext berücksichtigt werden. In jedem Falle stellt die Verwendung der Empfehlung zu den „Workflow Requirements“ einen effektiven Weg dar, den Analysten, Produktmanagern und Nutzern ein ganzheitliches Bild von der Anwendung zu vermitteln und dadurch Anknüpfungspunkte und Problembereiche einfacher lokalisieren zu können.

Bei der Empfehlung „**Evaluation Activities**“, als Mittel zur kontinuierlichen Überprüfung, existiert ein differenziertes Meinungsbild der Experten. Generell wird bestätigt, dass eine regelmäßige Überprüfung, z. B. durch Usability Tests Sorge trägt, Nutzer- und Nutzungsprobleme zu identifizieren und zu beheben. Laut Aussagen der Experten muss jedoch nicht immer ein vollständiger, komplexer Arbeitsablauf implementiert sein, um Usability Tests durchzuführen. In Ausnahmefällen kann es sogar zu spät sein erst dann zu testen, wenn das System ausreichend komplex ist. Oft reichen schon kurze und einfache, dafür häufigere, Überprüfungsaktivitäten um wertvolle Informationen zu sammeln. Unumstritten galt, dass es sich bei den Evaluationsaktivitäten um eine effektive Maßnahme für die Praxis handelt.

Das frühe Einholen von Kunden- und Nutzerfeedback, wie in der Empfehlung „**Gain Quick User/Customer Feedback**“ beschrieben, wurde als nützlich, bis sehr nützlich bewertet und hatte in ähnlicher Form bereits Anwendung in der Praxis gefunden. Wenn Kunden und Nutzer früh und regelmäßig in die Entwicklung einbezogen werden hat dies zur Folge, dass nicht nur Probleme schnell erkannt und behoben werden können, sondern es führt auch zu einem größerem Engagement von Kundenseite. Es ist allerdings vom jeweiligen Projekt und den Rahmenbedingungen abhängig, ob diese Empfehlung in der Praxis umzusetzen ist. Ist dies jedoch der Fall wird es als sehr effektiv betrachtet.

Die in der Empfehlung „**Close Collaboration**“ geforderte enge Zusammenarbeit von Personen mit unterschiedlichen Fähigkeiten und Kompetenzen setzt in der Praxis natürlich eine bestimmte Art von sozialen Fähigkeiten der Beteiligten voraus. Wenn dies gegeben war, hat

Die in der Empfehlung „**Close Collaboration**“ geforderte enge Zusammenarbeit von Personen mit unterschiedlichen Fähigkeiten und Kompetenzen setzt in der Praxis natürlich eine bestimmte Art von sozialen Fähigkeiten der Beteiligten voraus. Wenn dies gegeben war, hat

sich diese Empfehlung im praktischen Einsatz als überwiegend sehr nützlich erwiesen. Ein positiver Seiteneffekt ist, dass beispielsweise das User Interface einer Anwendung von allen beteiligten Personengruppen besser akzeptiert wird wenn sie das Gefühl haben, Designentscheidungen mit beeinflussen zu können und daran mitgewirkt zu haben. Eine enge Zusammenarbeit muss jedoch nicht zwangsläufig eine räumliche Nähe implizieren. Bei verteilten Teams lässt sich die räumliche Entfernung auch durch z. B. tägliche Video- / Telefonkonferenzen und andere technische Hilfsmittel überbrücken. Die Anwendung erwies sich laut Aussagen der Experten als sehr effektiv.

5.0 Folgerungen

Durch Analyse konnten die Empfehlungen in Zusammenarbeit mit den Experten so überarbeitet und verbessert werden dass sie als Hilfsmittel zur Ergänzung agiler Entwicklungsmodelle dienen können. Die Empfehlungen sind nachfolgend beschrieben und zur Diskussion gestellt.

- **Usability Engineers and UI-Designers**
Usability Engineers und UI Designer sollten integraler Bestandteil der interdisziplinären Teams sein. Dies ist eine Grundvoraussetzung, damit übergeordnete Usability Anforderungen berücksichtigt, sowie das entsprechende Analyse-Aktivitäten (aus UE Sicht) richtig durchgeführt werden können.
- **Preliminary Exploration & Analysis**
Beide agilen SE Modelle fordern im Falle eines neu gestarteten Projektes eine Explorations-Phase, die allen anderen Entwicklungsbemühungen vorangeht. In dieser Phase beschäftigen sich die Entwickler mit Fragen bzgl. der zu verwendenden Technologien, Architekturen und der technischen Machbarkeit. Parallel dazu sollten Usability Engineers und Produkt Analytiker diese Zeit für grobe

Analysezwecke nutzen, um erste Einblicke in den Workflow und den Nutzungskontext zu erhalten. Um allerdings die Agilität zu bewahren muss beachtet werden, dass eine umfangreiche Dokumentation der Ergebnisse keine Angemessene Form für diese Aktivität ist. Vielmehr sollten leichtgewichtige Artefakte erstellt und das Wissen mit dem gesamten Team einfach und effizient geteilt werden.

- **Real Users**
Aufgrund der weichen Definition der Customer Stakeholder, in beiden agilen Modellen, ist nicht sichergestellt, dass tatsächlich reale Endnutzer in der Gruppe der Customer Stakeholder vertreten sind. Aus UE Sicht ist es entscheidend, dass reale Endnutzer als Quelle für Informationen bzgl. des Nutzungskontext und der Workflows dienen. Daher muss darauf geachtet werden, dass bei entsprechenden Aktivitäten Endnutzer statt diverser Kundenvertreter vor Ort sind, selbst wenn letztere ein gewisses Verständnis für die Bedürfnisse der Nutzer vorweisen. Eine Notwendigkeit für die Einbeziehung echter Nutzer besteht häufig nur dann, wenn es um die Entwicklung von Benutzungsschnittstellen und um die Implementierung von Arbeitsabläufen geht. Bei technisch dominierten Entwicklungen kann ggf. eine sporadische Einbeziehung realer Nutzer ausreichend sein.
- **Workflow Requirements**
Das Product Backlog (in Scrum) und die User Stories (in XP) sind angemessenen Artefakte, um Workflow Requirements festzuhalten. Durch die Eigenschaft der Artefakte, sukzessive und iterativ Anforderungen zu definieren besteht allerdings die Gefahr, die Sicht auf den Übergeordneten Zusammenhang der Einzelanforderungen zu verlieren (anders als beispielsweise in einem komplexen Use Case Dokument oder einer Kontextbeschreibung). Beide agilen Artefakte sind dafür gedacht, „High-Level“ Anforderungen zu dokumentieren, anstatt für vollständige Workflow-

Beschreibungen. Den Workflow mit Use Cases und/oder Szenario-Beschreibungen zu modellieren wäre angemessen, aber wird von keinem der Modelle angestrebt. Daher sollten zur Sicherstellung der ganzheitlichen Betrachtung die richtigen (leichtgewichtigen) Werkzeuge und Formate für die entsprechenden Aufgaben eingesetzt werden, so z. B. Essential Use Cases, Szenarios, etc. In der Praxis muss jedoch sichergestellt sein das nicht nur einzelne Arbeitsabläufe der Nutzer dokumentiert und implementiert werden, sondern auch die Beziehungen zwischen den Arbeitsabläufen der Nutzer, im konkreten Handlungskontext, berücksichtigt werden.

- **Evaluation Activities**
Weder Scrum noch Extreme Programming sehen überprüfende und bewertende Aktivitäten (wie z.B. Usability Tests) in irgendeiner Stufe der Entwicklung als Teil des Modells vor. Die explizite Durchführung von Testing-Aktivitäten sollten an einen Release Plan gekoppelt sein. Zum einen um ein ausreichend komplexes System vorliegen zu haben, zum anderen, dass das nutzerzentrierte Testen Bestandteil des Modells wird. Um jedoch die Agilität zu bewahren reichen Tests am Ende eines Releases nicht aus. Man muss hier differenzierter vorgehen und auch kleinschrittige Überprüfungen durchführen. Wichtig ist eben sicherzustellen, dass komplexere Arbeitsabläufe regelmäßig und ganzheitlich überprüft werden. Dazu muss das zu prüfende System natürlich ein hinreichendes Maß an Komplexität bieten.
- **Gain quick user/customer feedback**
Sowohl das Extreme Programming, als auch Scrum teilen sich das Prinzip des Customer On-Site. Dies bedeutet, dass immer ein Repräsentant der Kunden vor Ort ist und das Entwicklungsteam mit seinem Wissen über die Bedürfnisse der Nutzer zu unterstützen. Vorausgesetzt, dass es sich bei dieser Rolle um einen echten Endnutzer handelt, sollte diese Ressource genutzt werden, um beispielsweise, frühe Designentscheidungen, Workflow-Beschreibungen

und andere Arbeitsergebnisse zu validieren, bevor sie in die Entwicklung einfließen. Dies ermöglicht ein sehr schnelles und unkompliziertes Nutzerfeedback ohne den Entwicklungsprozess wesentlich zu verlangsamen. Solche Maßnahmen sollten ein integraler Bestandteil des Entwicklungsprozesses sein um die Usability zu gewährleisten.

- **Close Collaboration**

Die Agilen Modelle bieten eine gute Möglichkeit für die enge Zusammenarbeit von Programmierern und Interaktionsdesignern bzw. Usability Engineers. Aufgrund der sich überlappenden Entwicklungsphasen und der Interdisziplinarität der Teams, kann beispielsweise die Machbarkeit bestimmter Interaktionsmodelle bzgl. des Verhältnisses von Aufwand und Nutzen mit Programmierern bestimmt werden. Dadurch bekommen die Designer einen tieferen Einblick in die zugrunde liegende Architektur und können die Einflüsse auf die Entwicklung besser verstehen und somit auch wesentlichen Einfluss darauf nehmen, wie schnell die Implementierung voran geht. Designentscheidungen können gleichzeitig Usability Anforderungen und technische Beschränkungen berücksichtigen. Daher ist eine starke Kollaboration zwischen allen involvierten Parteien sehr wichtig und vom hohen Nutzen.

6.0 Zusammenfassung & Ausblick

In dem vorliegenden Paper haben sich die Autoren mit der Fragestellung auseinandergesetzt, in wie weit agile Vorgehensmodelle (hier: Scrum & XP) bereits User Centred Design (UCD) - Aspekte, wie Aktivitäten, Rollen und Artefakte, berücksichtigen. Die Analyse der theoretischen Beschreibungen der Modelle zeigte, dass diese Defizite in Bezug auf die Realisierung der UCD-Aspekte aufweisen. Um diese auszuräumen wurden entsprechende Empfehlungen erarbeitet und dokumentiert. Wichtige Voraussetzung dabei war, dass

durch die Anwendung weder die grundsätzlichen Eigenschaften des agilen Ansatzes, noch die Spezifika der beiden konkreten Modelle verletzt werden.

Die Empfehlungen wurden durch Experten-Reviews evaluiert, bewertet und anschließend diskutiert. Das Ziel war keine quantitative Auswertung, sondern vielmehr Tendenzen bezüglich der Validität und Praktikabilität der Empfehlungen zu erhalten, um sie in der Praxis effektiv anwenden zu können.

Übergeordnet zeigte sich, dass die Empfehlungen die Praxis widerspiegeln. Laut der Experten wurden einige der Empfehlungen bereits, in vergleichbarer Form in der Praxis angewandt und wurden als z. T. sehr nützlich und praktikabel bewertet.

Insgesamt zeigte die Analyse noch einmal die Notwendigkeit von UCD auch im agilen SE auf und bestätigte die Autoren darin, Hilfestellung und Anleitung für diesen Einsatz zu liefern. Die Empfehlungen sind ein sinnvoller Schritt und können in den jeweiligen Unternehmungen in konkrete Maßnahmen für die Praxis überführt und angewandt werden.

Die Praktikabilität und Effektivität der abgeleiteten Maßnahmen sollte im weiteren durch Folge-Untersuchungen, möglichst auch quantitativer Art, überprüft werden, um die Ansätze ggf. auch noch im Hinblick auf Effizienz zu optimieren, oder um konkrete Methoden und Vorgehensmodelle zu entwickeln.

In Zukunft werden die Autoren noch weitere Modelle untersuchen und die Empfehlungen, in Form von konkreten Maßnahmen, in der Praxis evaluieren, um daraus Rückschlüsse für weitere, ergänzende Empfehlungen zu ziehen.

7.0 Literaturverzeichnis

- Beck, K. (2000): *Extreme Programming explained*. Boston: Addison-Wesley.
- Böhm, B. (1988): *A Spiral Model of Software Development Enhancement*. IEEE Computer. Vol. 21. pp. 61-72.
- Cooper, A. (2003): *About Face 2.0*. Indianapolis: Wiley Publishing Inc.
- DIN EN ISO 13407 (1999): *Human-oriented design processes for interactive systems*. Brussels: CEN – European Committee for Standardization.
- Duechting, M., Nebe, K., Zimmermann, D. (2007): *Incorporating User Centered Requirement Engineering into Agile Software Development*. In: Jacko, J. (Hrsg.): *Proceedings of HCI 2007*. Heidelberg: Springer Verlag. (in prep).
- Glinz, M. (1999): *Eine geführte Tour durch die Landschaft der Software-Prozesse und – Prozessverbesserung*. In: Zehnder, C.A., Dittrich, K., Ragaz, N. (Hrsg.): *Informatique*. Zürich: Schweizerischer Verband der Informatikorganisationen
- IBM (1996): *Ease of Use Model*. abgerufen von http://www3.ibm.com/ibm/easy/e-ou_ext.nsf/publish/1996. (11/2004).
- Larman, C. (2004): *Agile & Iterative Development – A Manager's Guide*. Boston: Addison-Wesley.
- Mayhew, D. (1999): *Usability Engineering Lifecycle*. San Francisco: Morgan Kaufmann.
- Nebe, K., Grötzbach, L., Hartwig, R. (2006): *Integrating User Centered Design in a Product Development Lifecycle Process: A Case Study*. In: Arabnia, H., Reza, H. (Hrsg.): *Proceedings of 2006 International Conference on Software Engineering Research & Practice*. Las Vegas: CSREA Press, S. 695-701
- Schwaber, K., Beedle, M. (2002): *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall.
- Schwaber, K. (2004): *Agile Project Management with Scrum*. Redmond: Microsoft Press.
- Zimmermann, D., Groetzbach, L. (2007): *A Requirement Engineering Approach to User Centered Design*. In: Jacko, J. (Hrsg.): *Proceedings of HCI 2007*. Heidelberg: Springer Verlag.

AK-Berufsbild Usability