

Praktiken der Nutzerintegration im Entwicklungsprozess von KMU

Oliver Stickel¹, Corinna Ogonowski¹, Timo Jakobi¹, Gunnar Stevens², Volkmarr Pipek¹, Volker Wulf¹

Institut für Wirtschaftsinformatik, Universität Siegen¹

Betriebswirtschaft, insb. Wirtschaftsinformatik, Hochschule Bonn-Rhein-Sieg²

Zusammenfassung

Die Vorteile, Nutzer aktiv, früh und langfristig in Entwicklungsprozesse zu integrieren, um Fehlentwicklungen zu vermeiden und Nutzerbedürfnisse zu adressieren, sind nicht nur in der akademischen Forschung bekannt. Prozesse und Strukturen in Unternehmen der IKT-Branche sind bereits häufig agil implementiert. Dennoch schaffen es kleine und mittlere Unternehmen (KMU) oftmals nicht, die Potentiale einer Nutzerintegration konsequent auszuschöpfen. In Fallstudien wurden drei unterschiedliche KMU analysiert, wie sie die Stimme des Nutzers im Entwicklungsprozess berücksichtigen. Unterschiedliche Strategien der Nutzerintegration, die sich in Rollen und Werkzeugen, in Anforderungen und Problemen an das Nutzersample, Methoden und Datenaufbereitung widerspiegeln, werden beleuchtet. Unser Beitrag soll helfen, Herausforderungen und Probleme von KMU auf der Suche nach angemessenen und passgenauen Wegen der Nutzerintegration zu verstehen und Lösungen zu gestalten.

1 Einleitung

Software ist zu einem unverzichtbaren Bestandteil privaten wie unternehmerischen Handelns geworden. Für ihren erfolgreichen Vertrieb kommt der Gebrauchstauglichkeit und dem positiven Nutzererlebnis (Usability und User Experience, kurz UUX) eine immer höhere Bedeutung zu. Insbesondere im Sektor der kleinen und mittleren Unternehmen (KMU) herrschen hinsichtlich der Berücksichtigung dieser Qualitätsmerkmale und ihrer methodischen Absicherung jedoch noch deutliche Defizite (Hering et al., 2013). Oft mangelt es aus finanziellen, methodischen, logistischen oder historischen Gründen am Einbezug von Anwendern, dem gezielten und systematischen Einholen von Nutzerfeedback und/oder dessen Einbindung in den Entwicklungsprozess. Dabei gilt eine solche Integration über alle Phasen eines Softwareprojektes als einer der wichtigsten Faktoren für gute UUX (Hansson et al., 2006). Jedoch zeigt sich, dass die ISO-Norm sowie andere Prozessmodelle (z.B. User Centered Design) Lücken in der Umsetzung und der Einbettung in KMU-Prozesse aufweisen. Der

vorliegende Beitrag soll dazu beisteuern, Wege aufzuzeigen, diese Lücken zu schließen. Aufbauend auf einem Verständnis der Mensch-Computer Interaktion und der computerunterstützten Gruppenarbeit als sozio-technische und stark an Praktiken von Menschen orientierte Disziplinen (Wulf et al., 2011), wählten wir Fallstudien in drei möglichst kontrastiven IKT-entwickelnden Unternehmen, um ein breites Verständnis über existente Praktiken der Nutzerintegration in Entwicklungsprozessen und deren Herausforderungen zu eruieren: *Alpha*, eines der größten deutschen KMU-Entwicklungshäuser; *Gamma*, ein ebenfalls großes KMU, dessen UUX-fokussierte Prozesse noch wenig ausdifferenziert sind; und *Delta*, ein kleines, designgetriebenes Softwareunternehmen, das vor allem im Auftrag arbeitet. Darauf basierend diskutieren wir beteiligte Rollen, Kanäle und Werkzeuge sowie die Herausforderungen in der Auswertung und Analyse von Kundenfeedback – drei Kategorien, die die Nutzerintegration in die Entwicklung maßgeblich moderieren und nicht zuletzt auch von ihr moderiert werden.

2 Stand der Forschung

Die grundsätzlich hohe Relevanz der Integration von Nutzern in die Entwicklung von Softwareprodukten wird in einer Vielzahl wissenschaftlicher und praktischer Diskurse betont. Participatory Design (Ehn & Kyng, 1987) gilt hierfür als erste systematische Strömung, wenn auch mit einem politisch-gewerkschaftlichen als UUX-fokussierten Hintergrund. Auf Basis dieser Philosophie wurde jedoch schnell auch außerhalb der akademischen Welt, im Rahmen kommerzieller Entwicklungen mit der kontinuierlicher Nutzerintegration experimentiert (Floyd et al., 1989). Weitere Überführungen in systematische, prozessorientierte Modelle fanden beispielsweise mit der integrierten Organisations- und Technikentwicklung (Wulf & Rohde, 1995) statt. Auch diese Ansätze betonen die enge Kollaboration zwischen Nutzern und Entwicklern, die in der Praxis, insbesondere für KMU, aus verschiedenen Gründen (z.B. Logistik, Kosten, Aufwand, etc.) kaum umsetzbar ist (Hansson et al., 2006). Im aktuell häufig verwendeten Modell des User Centered Design (UCD, Mao et al., 2005), wird der Nutzer stärker als Ressource betrachtet, die prozessorientiert nach Bedarf eingebunden wird. Diese Herangehensweise kann eher als geeignet für KMU-Besonderheiten, wie z.B. stärker limitierte Ressourcen, betrachtet werden und wurde im Rahmen der DIN EN ISO 9241 auch im Sinne eines Standards kodifiziert. Durch die relativ offene Spezifizierung von UCD und ISO 9241 bleibt deren jeweilige konkrete Umsetzung jedoch vage. Gemeinsam mit Einschränkungen wie limitierten Ressourcen oder auch Traditionen sorgt dies dafür, dass KMU nutzerzentrierte Methoden oft nur in geringen Ausmaß anwenden (Hering et al., 2013).

Softwareentwickelnde Unternehmen - auch KMU -, strukturieren ihre Entwicklungsprozesse zunehmend anhand von Prinzipien der agilen Softwareentwicklung (Beck et al., 2001), die klar von klassischen Entwicklungsphilosophien wie z.B. dem Wasserfallmodell abweichen. Agile Methoden wie Scrum (Schwaber, 1995) oder Kanban (Anderson & Reinertsen, 2010) betonen Flexibilität, die Reaktionsfähigkeit auf unvorhersehbare Entwicklungen, lauffähige Prototypen, geringen Overhead, wie z.B. tiefe Dokumentationen, und auch die Zusammenarbeit mit dem Kunden. Zwar besteht zwischen UCD-Ansätzen und agilen Entwicklungsmodellen keine Deckungsgleichheit, jedoch wird durch die vielen geteilten Charakteristika wie z.B. iterative Entwicklung und die hohe Relevanz des Anwenders zumindest eine Artver-

wandtschaft deutlich (Chamberlain et al., 2006). Daher gilt die agile Strukturierung von Softwareentwicklung als relevanter Ansatzpunkt zur prozesshaften Nutzbarmachung von aktiver Einbindung von Anwendern auch für KMU. In diesem Bereich existieren bereits Vorarbeiten wie z.B. Versuche, explizite Design-Zyklen in agile Prozesse (Silva et al., 2012) oder UUX-Experten in agile Teams zu integrieren (Beyer, 2010). Es wurden auch bestimmte Rollen in der agilen Entwicklung beleuchtet: Beispielsweise wurde der Product Owner (PO in Scrum-Doktrin: zentrale, für die Produktvision verantwortliche Rolle), als die für UUX maßgebliche Rolle diskutiert. Singh (2008) argumentiert auch dafür, mehrere PO einzusetzen, wobei einer explizit auf UUX geschult werden sollte.

Insgesamt ist jedoch festzustellen, dass trotz thematischer Nähe und entsprechender Vorarbeiten noch große Lücken in der tatsächlichen praktischen Integration von nutzerzentrierter UUX-Evaluation und agiler Entwicklung bestehen, die nicht zuletzt auch auf mangelndes Verständnis tatsächlicher Entwicklungs- und Arbeitspraxis bei Software-Unternehmen zurückgehen (Draxler et al., 2014; Ferreira et al., 2007).

3 Fallstudien und methodisches Vorgehen

Alpha ist ein großer Mittelständler mit ca. 500 Mitarbeitern, der seit längerer Zeit einen verstärkten Fokus auf die Einbindung von Nutzern legt. Der Produktschwerpunkt liegt auf Softwareentwicklung für Endkunden, insbesondere im Bereich der Finanzverwaltung. Unsere Arbeit mit Alpha konzentrierte sich auf ein Finanzverwaltungsprodukt für Endnutzer, das agil entwickelt und plattformübergreifend, auch mobil, vertrieben wird. Es wurden 60-minütige Interviews mit dem verantwortlichen PO, dem Social Media Management (SMM), dem Leiter des Support-Teams, einem Support-Mitarbeiter, einem Entwickler sowie zwei Mitarbeitern des Usability-Labors geführt. Zudem fanden teilnehmende Beobachtungen bei Nutzertests in diesem Labor (3h) und bei einem Scrum-Planungsmeeting (4h) statt.

Gamma ist ein Unternehmen mit ca. 200 Mitarbeitern und ebenfalls am oberen Ende der KMU-Klassifikation einzuordnen. Durch die strategische Entscheidung, eine eigene Smart Home Produktlinie zu positionieren, rückten für das Unternehmen gegenüber dem klassischen Portfolio Fragen der Interface-Gestaltung stärker in den Vordergrund. Das bisherige Kerngeschäft bildet die Technologieentwicklung für Heim-Netzwerke. Unsere Forschung konzentrierte sich bei Gamma auf die für das neue Produkt geschaffenen Entwicklungsprozesse und das hausinterne Testsample zur Evaluation von Prototypen. Hierfür wurden 90- bis 120-minütige Interviews mit den Leitern der Entwicklungsabteilung, der Abteilung Design, Verification & Testing (DVT), dem Produkt Marketing sowie dem verantwortlichen PO geführt.

Delta ist ein wachsendes KMU mit 11 Mitarbeitern, dass Entwicklungs- und Beratungsleistungen sowie Design von innovativer Software, Apps und digitalen Produkten im Bereich Internet of Things, Energie und Elektromobilität anbietet. Um die Unternehmensstruktur, Führungsstrategie und die agilen Entwicklungsprozesse besser zu verstehen, wurden 60- bis 90-minütige Interviews mit zwei der drei Gesellschafter, die gleichzeitig als Scrum Mas-

ter/Head of Project Management (PM) und Creative Director (CD) agieren, dem Senior Art Director UI/UX (AD) sowie einem Mobile Developer geführt.

Unser methodisches Vorgehen orientiert sich an den durch Wulf et al. (2011) etablierten Design Case Studies. Die dafür notwendigen Interviews wurden anhand eines semi-strukturierten Gesprächsleitfadens geführt, Audio-Aufzeichnungen angefertigt und transkribiert. Weiterhin wurden Feldnotizen angelegt und Artefakte (z.B. interne Berichte von Nutzertests) gesammelt. Alle Daten gingen in die Analyse ein, die in Anlehnung an die Thematische Analyse (Braun & Clarke, 2006) vorgenommen wurde. Die thematische Analyse ist eine der Grounded Theory artverwandte, jedoch weniger auf extensive Theoriebildung fokussierte Art der qualitativen Analyse. Ähnlich einer Herangehensweise mittels Grounded Theory legten wir den Fokus auf feldgetriebene und unmittelbare Weiterentwicklung des Analyse- und Forschungsprozesses. Dies wurde durch induktive, kontinuierliche und iterative Codierung ab dem ersten Interview, das Schreiben von Memos sowie Diskussion und Kategorienbildung gemeinsam mit Forschern, die nicht in die Feldstudien involviert waren, (im Sinne der Inter-Coder-Reliabilität) sichergestellt. Auch die Auswahl der Interviewpartner fand feldgetrieben statt. Die im Folgenden berichteten Kategorien entwickelten sich durch diesen Analyseprozess als zentrale Aspekte der Nutzerintegration in Softwareentwicklung durch KMU.

4 Praktiken und Herausforderungen

4.1 Rollen

Bei Alpha zeigte sich, dass eine Vielzahl von Rollen bewusst im Kontakt mit Anwendern steht. Vor allem das hauseigene Support-Team wird täglich mit unterschiedlichsten Nutzungsproblemen konfrontiert, jedoch auch explizit auf das Aufnehmen von gewünschten Features geschult ist. Das Customer Lab führt auf Anfrage des Entwicklungsteams Nutzertests (hauptsächlich per Lautem Denken) mit einem kontinuierlich erweiterten Stamm an Teilnehmern durch. Weiterhin ist das Social Media Management in ständigem Kontakt mit Nutzern über soziale Medien, pflegt dort Informationen ein, überwacht und moderiert aktiv die Konversationen von Anwender-Communities und bemüht sich, ein Gefühl für die „Stimmung“ auf der jeweiligen Plattform zu bekommen. Auch der PO hat direkten Nutzerkontakt, konsolidiert jedoch auch das Feedback der anderen beteiligten Rollen und kanalisiert diese in *User Stories*. Bemerkenswert ist, dass für das Finanzverwaltungsprodukt ein Team aus zwei PO eingesetzt wird, wobei einer sich auf die Gesamtvision fokussiert und einer für tägliche Details, insbesondere User Stories, verantwortlich ist. Für Letzteres werden explizit auch alle anderen Rollen nach Bedarf mit einbezogen wenn Unklarheiten bestehen. Entwickler haben bei Alpha keinen direkten Nutzerkontakt.

Gamma's Ablauforganisation ist produktorientiert in Business Units aufgeteilt. Innerhalb der von uns untersuchten Einheit ist der direkte Kontakt mit Nutzern des Testsamples stark kanalisiert. Obwohl intern getestet wird, wurde mit Bugzilla ein Feedbacktool eingerichtet, das bei der Abteilung DVT aufgehängt ist. Diese ist im Wesentlichen dafür zuständig neue Soft-

ware-Builds auf ihre Vollständigkeit und Korrektheit zu überprüfen. Darüber hinaus informiert sie die Entwicklung über Fehlerberichte aus dem Feedbacktool. Die Entwicklungsabteilung hat, anders als bei den sonstigen Units, eine stärker koordinierende und konzeptionelle Aufgabe. Viele Entwicklungstätigkeiten wurden aufgrund des für Gamma neuen Aufgabenfeldes an externe Partner vergeben. Diese werden von der Entwicklungsabteilung instruiert. Für Feedback und Berichte zu UUX ist der PO verantwortlich, der grundsätzlich die alleinige Handlungs- und Entscheidungskompetenz innehat, sich jedoch auch von externen Agenturen beraten lässt. Weiterer direkter Nutzerkontakt mit dem PO oder der Entwicklung entsteht bei konkreten Rückfragen zu einzelnen Problemen. Für die Zeit nach dem Produktlaunch ist ein enger Support und weiterführende Beratung durch Online-Chats und Webinare geplant, um das Produkt durch schnelle Updates weiter ausreifen zu lassen. Im Marketing findet ähnlich wie bei Alpha ebenfalls Austausch über Social Media Kanäle statt.

Delta setzt auf eine flache Hierarchie, die sich in eine Design- und Entwicklungs-Unit sowie Social Media Marketing und die beiden Funktionen der CEOs aufteilen lassen. Vor allem die Rollen des Head of PM, des Creative und des Art Director stehen in engem Kontakt mit den Kunden, die in diesem Fall keine Endnutzer sind. User Centered Design wird bei Delta stärker auf den Kunden, aber auch peripher auf die tatsächlichen Endnutzer übertragen. Man denkt visionär und versucht immer wieder einen Schritt zurück zu treten, um die Konzepte und Lösungen durch eine Nutzerbrille zu betrachten. Für die interne Qualitätskontrolle der Konzepte und Softwarelösungen ist der PM und für UUX Aspekte der CD und AD zuständig. Die technische Güte wird durch klassisches Bug-Tracking, Performance- und Beta-Tests gesichert. Oft werden Mitarbeiter, die nicht in das Projekt involviert sind, mit internen Beta-Tests betraut. Zudem finden Feedbackschleifen dynamisch zwischen den Units und Rollen statt, die beim PM zusammenfließen. Am Ende jedes Sprints wird auch der Kunde um Feedback gebeten. Umfangreichere Nutzertests werden abhängig vom Projekt entweder an Dritte ausgelagert oder der Kunde führt diese selbst mit Hilfe eines eigenen Testsamples durch, das oftmals aus einem ausgewählten Anwenderkreis besteht. Die Entwicklung hat wie auch bei Alpha keinen direkten Kontakt zu Nutzern.

4.2 Kanäle und Werkzeuge

Im Falle von Alpha wird Microsofts TFS als internes Projektverwaltungswerkzeug eingesetzt, in das Artefakte wie z.B. User Stories, Bugs, etc. eingepflegt werden und das insbesondere für den PO das zentrale Werkzeug darstellt. Das Supportteam unterhält Kanäle per E-Mail, Telefon, Brief, Fax, Chat sowie in Form produktspezifischer Foren (letztere allerdings eher mit Fokus auf Kunden-helfen-Kunden). Feedback wird aus einem eigenen Support-Ticketssystem in TFS überführt. Zentrale Kanäle für das Social Media Management stellen Facebook und Twitter sowie produktspezifische Blogs dar. Featurewünsche oder Änderungsvorschläge werden kontextualisiert und durch Kundenstimmen untermauert an den PO weitergegeben, Routine-Probleme teilweise auch unmittelbar an das Entwicklungsteam weitergeleitet. Bemerkenswert ist, dass das SMM nicht geordnet am TFS partizipiert, sondern Feedback per Mail oder im persönlichen Gespräch an den PO gibt. Das Customer Lab unterhält als zentralen Kanal ein physikalisches Labor in dem, meist in Form des Lauten Denkens und entsprechender Video-Aufzeichnungen Nutzertests durchgeführt werden. Ergänzt wird

dies durch heuristische Evaluationen und Cognitive Walkthroughs, in jüngerer Zeit auch durch eigene Studien bei Teilnehmern Zuhause. Von den Testbatterien, die meist je 5-20 Teilnehmer umfassen, werden umfangreiche Berichte gestellt und ins TFS überführt. Der PO steht durch produktspezifische Feedback-Mailadressen (die ungleich der Support-Adressen sind) sowie durch aktives Beobachten der Rezensionen in mobilen App-Stores in Kundenkontakt – bemerkenswerterweise in manchen Fällen auch per Telefon – langjährige „Lead User“ rufen den PO teilweise direkt an. Unternehmensintern betonten weiterhin alle Interviewpartner die Wichtigkeit des direkten persönlichen Kontaktes und Austausches, vor allem informell, z.B. in Kaffee-Ecken – „Flurfunk“ sowie eine räumliche Nähe.

Bis zum internen Nutzertest hat Gamma ein hauseigenes Produktmanagement-Datenbanksystem verwendet, in dem alle Anforderungen und Eigenschaften des Systems vorgehalten und gepflegt werden. Im Falle der neuen Produktlinie werden dort auch Wireframes hinterlegt, die im Wesentlichen zum Abgleich von ausgelagerten Entwicklungsaufträgen herangezogen werden. Zum Zeitpunkt der Fertigstellung aller geplanten Features hat Gamma ein lokales Testbett errichtet, das nur sehr rudimentär UUX beforscht. Im Zentrum steht nicht mehr die Suche nach innovativen Features, sondern die Gestaltung der UI-Komponenten und der Interaktionsfluss. Das Sample umfasst ca. 30 Haushalte, die das Produkt vor Marktstart über mehrere Monate Zuhause testen. Test und Aneignung finden dabei ohne Anleitung statt. Die einzigen Richtlinien sind, sich zu einer aktiven und umfassenden Nutzung zu verpflichten und bei Updates gefixte Probleme zu testen. Als Feedbackkanäle dienen Bugzilla, um nutzerseitig Tickets zu erstellen, und der informelle Austausch zwischen „Tür und Angel“. Zudem werden sämtliche Logfiles eines Systems erfasst, um eine Reproduzierbarkeit des Problems möglichst sicherzustellen. Sobald das Produkt am Markt ist, berichtet der Support und das Marketing an den PO.

Delta arbeitet mit der Managementsoftware Jira und Confluence, um die Agilität im Entwicklungsprozess zu gewähren. Die Software dient zum einen dem PM zur internen Koordination und zum anderen als Schnittstelle für größtmögliche Transparenz gegenüber den Kunden. User Stories und Bugs werden maßgeblich vom PM eingepflegt. Intern wird Feedback beim Daily Stand-Up und auf kurzem Weg zwischen PM und Mitarbeitern des Designs und der Entwicklung ausgetauscht (Flurfunk). Überdies hinaus und je nach Konzeption der Projekte, kann der Kunde selbst Tickets mit Feedback erstellen. Beta-Tests werden im Bereich mobiler Anwendungen durch die Software Testflight realisiert. Zudem wird Software zum Tracking von Nutzerverhalten eingesetzt, um Interaktionskonzepte und UI zu evaluieren. Für aktives Nutzerfeedback stehen verschiedene Möglichkeiten zu Verfügung: Entweder der Kunde sammelt das Feedback seiner Nutzer, selektiert es und stellt es Delta zur Verfügung; Nutzerfeedback aus Beta-Tests wird ungefiltert in Jira eingepflegt; oder es wird eine E-Mail-Adresse bereitgestellt, über die Nutzerfeedback mitgeteilt werden kann. Bei den letzten beiden Optionen ist Delta für die Organisation, Pflege und Bewertung des Feedbacks verantwortlich. Ein weiterer genutzter Feedbackkanal sind wie bei Alpha die App-Stores mit ihren Kundenrezensionen. Oft werden diese vom Kunden selbst betreut und je nach Bedarf an den PM weitergeleitet. Qualitatives Nutzerfeedback zum UI wird bislang nicht oder unsystematisch, z.B. bei Treffen mit dem Kunden und Beta-Testern, gesammelt.

4.3 Herausforderungen von Auswertung und Analyse

Alle drei KMU stellen heraus, dass sie die Wichtigkeit von Nutzerintegration in die Entwicklung sehr schätzen, sich jedoch immer wieder mit Schwierigkeiten in der Auswertung und Analyse konfrontiert sehen, beispielsweise hinsichtlich Fragestellungen wie *was der Kunde sagt, ist nicht notwendigerweise was er auch meint*, der Repräsentativität von Kundenwünschen oder dem Abgleich von Feedback mit Kriterien der Umsetzbarkeit oder technischen Spezifika. Im Laufe der Firmengeschichte von Alpha wurde mit verschiedensten Ansätzen experimentiert. Mittlerweile verwendet man eine Triangulation aus qualitativen und quantitativen Aspekten aus einer Vielzahl von Kanälen und Perspektiven: Zentral ist der PO, der endgültige Entscheidungen in Abwägung mit der langfristigen und strategischen Produktvision trifft und weshalb letztlich auch zwei PO eingesetzt werden, die sich gegenseitig ergänzen. Die Entscheidungen werden jedoch systematisch durch die beteiligten Rollen und Kanäle ergänzt – das Social Media Management nimmt hierbei eine qualitative Rolle ein und versucht Stimmungen und möglichst viel Kontext sowie eigene interpretative Leistung in den Entscheidungsprozess einzubringen. Das Customer Lab interpretiert und filtert nicht vor, sondern aggregiert methodisch sauber abgesichertes qualitatives und quantitatives Nutzerfeedback in geordneter Berichtsform. Der Support hingegen reicht Feedback und Nutzerwünsche nicht nur durch, sondern diskutiert dieses innerhalb der für das jeweilige Produkt zuständigen Gruppe und entscheidet weitgehend selbständig auf Basis von Erfahrung, Schulung und Unternehmenskultur, welches Feedback über das TFS weitergereicht wird. Weiterhin sammelt der Support eine große Datenbasis, die explizit auch quantifiziert wird, was dem Triangulationsprozess unterstützend zugeführt wird. Als essentieller Faktor für die erfolgreiche Analyse von Nutzerfeedback wird eine Unternehmenskultur der Ermächtigung gewertet, die Einzelrollen ohne Gefahr persönlicher Konsequenzen teilweise hitzige Diskussionen mit dem PO führen lässt, welche letztlich dem Produkt zugutekommen.

Ähnlich wie bei Alpha nimmt auch bei Gamma der PO eine Schlüsselfunktion ein. Er bewertet und klassifiziert aufkommende Hinweise und Informationen verschiedener Rollen und Kanäle und entscheidet eigenständig und in Abwägung mit dem avisierten Produktziel, welche Aspekte zu behandeln sind. Grundsätzlich orientiert man sich an der Faustregel, dass wenn mindesten zwei bis dreimal das gleiche Feedback zu einem bestimmten Aspekt eingeht, wird dieser als relevant gewertet. Das Filtern und Klassifizieren der über Bugzilla eingehenden Meldungen wird vom DVT verantwortet, die die Vorfälle entweder zur Entwicklung oder zur Entscheidung beim PO durchreichen. Eine wichtige Möglichkeit, konkrete Rücksprachen zu halten, um Feedback besser bewerten zu können und Lösungsvorschläge abzuholen, ist sowohl für Entwickler als auch für den PO in UUX-Fragen das direkte Gespräch mit Testern über den Flurfunk. Eine Kombination aus qualitativen und quantitativen Daten findet somit auch bei Gamma Anwendung, wenn auch nicht ganz so klar strukturiert wie bei Alpha, da die Arbeit mit dem Testsample keiner Abteilung klar zugewiesen ist. Schwierigkeiten bestehen hier zum einen, bei der Betreuung der Testhaushalte und zum anderen bei der „Betriebsblindheit“ der Nutzer, die als kritisch bewertet wird. Hierfür müsste aus Sicht des PO das involvierte Sample ständig erweitert oder partiell ersetzt werden, um die Güte des Feedbacks zu sichern.

Im Gegensatz zu den beiden KMU befindet sich Delta in einer etwas anderen Situation der Bewertung von Feedback und dem Treffen von Entscheidungen. Durch die Position als Dienstleister stehen sie zum einen in Abhängigkeit zu und in ständiger Absprache mit ihren Kunden und zum anderen müssen Ergebnisse und Feedback intern diskutiert werden. Die flache Hierarchie und die Unternehmensgröße sind förderlich für eine schnelle interne Bewertung, die beim PM zusammenläuft und an den Kunden kommuniziert wird, der letztendlich finale Entscheidungen trifft. Gängige Praxis ist es bislang quantitatives Feedback durch Tracking der Nutzung mit qualitativem Feedback per Email zu triangulieren. Dieses Vorgehen lässt sich personell und zeitlich sehr gut integrieren. Entscheidungshilfen hinsichtlich der Priorisierung des Feedback werden ähnlich wie bei Gamma anhand der Häufigkeit der Meldung gehandhabt, obwohl ein verbaler Austausch mit Nutzern, vor allem im Hinblick auf UUX-Aspekte, als wertvoller bewertet wird. Die nur sehr eingeschränkt vorhandenen personellen Ressourcen für Nutzungsstudien lassen dies aber nur schwer realisieren und bringen weitere Herausforderungen mit sich. Feedback was vom Kunden selbst oder durch Dritte erhoben und selektiert wird, bietet nur begrenzt Möglichkeiten Nachfragen zu stellen oder fehlende Kontextinformationen im Nachgang zu erheben. Vertrauen in die Validität der Daten stellt hierbei eine nötige Voraussetzung dar.

5 Diskussion und Zusammenfassung

Anhand der kontrastiven Fallstudien lässt sich zeigen, dass insbesondere die Größe des vorgehaltenen Samples an Test-Nutzern sowie die Ausdifferenzierung von Kanälen (intern wie extern) zur Nutzerintegration mit der Größe des Unternehmens skalieren können. Problematisch ist vor allem, dass kleinere Unternehmen eine große Notwendigkeit sehen, eine bessere Nutzerintegration vorzunehmen. Man behilft sich hier mittels interner Tests mit Kollegen aus anderen Projekten oder einem mentalen Fokus auf den Nutzerblick, ist sich jedoch der Unzulänglichkeit solcher Methoden steigend bewusst. Insbesondere für Unternehmen, die nicht direkt für Endkunden, sondern im Auftrag entwickeln, kommt die Problematik hinzu, nicht nur interne Stakeholder, sondern auch zahlende Kunden von Investitionen in die Nutzerintegration zu überzeugen. Ein möglicherweise probates Mittel wäre die Externalisierung bestimmter Aspekte der Zusammenarbeit mit Nutzern, beispielsweise durch Crowd-Testing oder realweltliche Tests *as a service* (z.B. in Living Labs (Ley et al., 2015)) – einerseits mit Bezug auf rationalisierungsökonomische Faktoren, andererseits auch im Sinne von Überzeugungsarbeit am Kunden, die solche fokussierten Dienstleister für Nutzerintegration im Idealfalle durch umfangreiche Portfolios und harte Zahlen (ROI) leisten können. Eine ähnliche Situation zeigt sich in den beteiligten Rollen, die mit der Unternehmensgröße skalieren können. Die Ausdifferenzierung verschiedener Rollen, die aktiv und aus unterschiedlichen Blickwinkeln an der Nutzerintegration arbeiten und in strukturierte Prozesse eingebunden sind, scheint ein zentraler Erfolgsfaktor zu sein, wobei die agile Strukturierung dieser Prozesse ein gutes Rahmenkonstrukt bieten kann. Wichtig scheint weiterhin, dass die Rollen eine gewisse Ermächtigung erfahren und bei Entscheidungen auch tatsächlich berücksichtigt werden. Die PO-Rolle ist hierbei zentral und ausgesprochen mächtig, was durch die Ermächtigung anderer Rollen sowie UUX-fokussierter Schulung von PO moderiert werden kann.

Außerdem sollte eine genuine Unternehmenskultur geschaffen werden, die auf den Nutzer fokussiert und einen schnellen, informellen Austausch im Unternehmen fördert. Gerade dieser Aspekt ist für kleinere KMU gut umsetzbar. Die hier aufgeführten Aspekte sollen beim Umgang mit und der Analyse von Nutzerfeedback helfen, denn auch wenn in einem Unternehmen aktiv Nutzer in die Entwicklung integriert werden, bleiben Unsicherheiten im Umgang und der Analyse komplexer Herausforderungen. Die Ausdifferenzierung von Rollen, Werkzeugen und entsprechender Unternehmenskultur, gerahmt von agiler Strukturierung der Entwicklung hilft, wobei qualitative und quantitative Daten vom und Blickwinkel auf den Nutzer von zentraler Bedeutung sind. Erfolgreiche Nutzerintegration scheint sich so unter dem Schlagwort „*Multi-Channel Feedback*“ (Draxler et al., 2014) subsumieren zu lassen, welche mit der Unternehmensgröße skaliert und in weiteren Forschungsarbeiten Thema sein wird.

Danksagung

Diese Arbeit wurde gefördert vom Bundesministerium für Wirtschaft und Energie (FKZ: 01MU14001A und 01MU12026A).

Literaturverzeichnis

- Anderson, D. J., & Reinertsen, D. G. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, Washington: Blue Hole Press.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Agile Manifesto. Retrieved from <http://agilemanifesto.org/>
- Beyer, H. (2010). User-Centered Agile Methods. *Lectures on Human-Centered Informatics*.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3, 77–101.
- Chamberlain, S., Sharp, H., & Maiden, N. (2006). Towards a Framework for Integrating Agile Development and User-Centred Design. In *Extreme Programming and Agile Processes in Software Engineering* (Vol. 4044, pp. 143–153).
- Draxler, S., Stickel, O., Winter, D., & Stevens, G. (2014). Nutzerintegration in Softwareprojekte durch Multi-Channel Feedback. In *Proc. MuC* (pp. 175–184).
- Ehn, P., & Kyng, M. (1987). The collective resource approach to system design. In *Computers and democracy: A Scandinavian challenge* (pp. 17–57). Avebury.
- Ferreira, J., Noble, J., & Biddle, R. (2007). Agile Development Iterations and UI Design. In *Proc. AGILE 2007* (pp. 50 - 58).
- Floyd, C., Mehl, W.-M., Reisin, F.-M., Schmidt, G., & Wolf, G. (1989). Out of Scandinavia: Alternative Approaches to Software Design and System Development. *Human-Computer Interaction*, 4(4), 253–350.
- Hansson, C., Dittrich, Y., & Randall, D. (2006). How to Include Users in the Development of Off-the-Shelf Software: A Case for Complementing Participatory Design with Agile Development. In *Proc. HICSS*.

- Hering, D., Kraft, X., Schwartz, T., & Wulf, V. (2013). Usability-Hindernisse bei Software entwickelnden KMU. In *Proc. MuC* (pp. 9–18).
- Ley, B., Ogonowski, C., Mu, M., Hess, J., Race, N., Randall, D., Rouncefield, M., & Wulf, V. (2015). At Home with Users: A Comparative View of Living Labs. In *Interacting with Computers*. 27, 1, 21–35.
- Mao, J.-Y., Vredenburg, K., Smith, P. W., & Carey, T. (2005). The State of User-centered Design Practice. *CACM*, 48(3), 105–109.
- Schwaber, K. (1995). SCRUM Development Process. In *Proc. OOPSLA* (pp. 117–134).
- Silva, T., Silveira, M. S., Maurer, F., Hellmann, T., Paulo, U. D. S., Carlos, C. D. S., Universidade, P. (2012). User Experience Design and Agile Development: From Theory to Practice. *JSEA*, 2012(October), 743–751.
- Singh, M. (2008). U-SCRUM: An Agile Methodology for Promoting Usability. In *Proc. AGILE* (pp. 555 - 560).
- Wulf, V., & Rohde, M. (1995). Towards an Integrated Organization and Technology Development. In *Symposium on Designing Interactive Systems '95* (pp. 55–64).
- Wulf, V., Rohde, M., Pipek, V., & Stevens, G. (2011). Engaging with Practices: Design Case Studies as a Research Framework in CSCW. In *Proc. CSCW* (pp. 505–512).

Kontaktinformationen

Oliver Stickel
Universität Siegen, Institut für Wirtschaftsinformatik
Kohlbettstr. 15, 57068 Siegen
Mail: oliver.stickel@uni-siegen.de