

„Walled Spaces for Open Design“ – Grenzerfahrungen aus zwei agilen Software-Migrationsprojekten

Hartmut Obendorf

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

Abstract: Dieser Beitrag versucht Grenzen für die Einrichtung eines gemeinsamen Designraums nachzuziehen, die dem Autor bei der Durchführung zweier ausgewählter agiler Entwicklungsprojekte aufgezeigt wurden. Die Erfahrungen wurden aus externer Sicht eines Beraters gesammelt, der sowohl den agilen Prozess mit eingeführt hat als auch selbst als Softwarearchitekt an der Entwicklung beteiligt war. Organisationseigene Rahmenbedingungen, das Zusammenspiel der involvierten Interessensgruppen und die explizierten Ziele der hier beschriebenen Software-Migrationsprojekte hatten dabei einen wesentlich einschränkenden Einfluss auf den Freiraum für partizipatives Design. Der Artikel berichtet von aufgetretenen Schwierigkeiten und beschreibt Lösungsansätze, die dabei verfolgt wurden.

1 Einleitung: Kontext

Das Konzept von Open Design Spaces leitet sich, motiviert durch die erfolgreiche Einbindung von Anwendern in der „social software“ des Web 2.0, in wesentlichen Zügen von partizipativer Softwareentwicklung (PD) ab, einer wiederum ursprünglich eng mit dem Empowerment von Arbeitern, der Mit- und Umgestaltung von Arbeit verbundenen Bewegung (vgl. [BV95]). Aus dieser Entstehungsgeschichte erklärt sich das Verständnis von partizipativer Gestaltung als ein Wert an sich. Agile Softwareentwicklung – hier das Umfeld für den Versuch, partizipatives Design durchzuführen – hat dagegen eine andere Geschichte: XP als vielleicht bekanntester Vertreter [BA04] ist im Umfeld innerbetrieblicher Softwareentwicklung entstanden und zielt vor allem darauf ab, die Arbeitsbedingungen für Entwickler und ihr Umfeld zu verbessern und die Planbarkeit und Qualität von Softwareentwicklung zu erhöhen.

Damit stammen agile Methoden aus einem Umfeld, in dem (vergleichsweise) begrenzte Ressourcen und wenige Benutzer, trotzdem aber viele Prozessbeteiligte die Regel sind. Wenig Geld bedeutet in agilen Projekten oft, dass Design mit (nichtfunktionaler, daher negativ konnotierter) Ästhetik gleichgesetzt wird und damit keine Notwendigkeit für Design gesehen, Usability als Luxus begriffen wird. Viele Prozessbeteiligte – neben Projektleitung, Entwicklern und Dienstleistern oft externe Dienstleister, Produktverantwortliche und in der finanziellen Verantwortung stehende Kunden – sind oft organisato-

risch enger eingebunden als die eigentlichen Anwender; die zukünftigen Benutzer sind zwar im Gegensatz zu Webprojekten bekannt, werden aber in der Organisation oft nicht wichtig genommen.

Die hier verwendeten Fallbeispiele, die Entwicklung einer Webanwendung als internes Informationssystem für ein spartenübergreifendes Servicecenter in einer Versicherung (V) und die Entwicklung einer vorbuchhalterischen Verwaltung für Kindertagesstätten (K), sind zudem Migrationsprojekte. Damit kommen eine hohe technische Komplexität durch das Zusammenspiel alter und neuer Technologien und damit eine verhältnismäßig große Unklarheit über entstehende Einschränkungen und notwendige Zusatzaufwände für Entwicklung zu den bestimmenden Faktoren hinzu. Der Raum für Design wird zudem durch vermeintlich genaue fachliche Vorgaben, die durch Altsysteme dokumentiert sind, zusätzlich begrenzt.

Das Ziel von Partizipation in diesem Umfeld ist die Öffnung der durch die Umstände zunächst sehr begrenzten Designräume, um eine Verbreiterung der betrachteten Möglichkeiten für eine Umsetzung zu erreichen. Die Motivation für die Einbeziehung von NutzerInnen besteht in den betrachteten Fallbeispielen aus zwei Komponenten: zum einen sollte mit der gemeinsamen Gestaltung für das Projekt geworben werden, zum anderen war die Risikominimierung durch frühes Erkennen von Designproblemen ein wichtiges Ziel. „Open Design Spaces“ können in diesem Zusammenhang zurückhaltend interpretiert als Medium für ein gemeinsames Design von Nutzern, Kunden und Entwicklern betrachtet werden.

Methodisches Vorgehen

In beiden Projekten, dem Web-Informationssystem für die Versicherung (V) wie der Buchhaltungsanwendung für die Kindertagesstätten (K), wurde der Autor hinzugezogen, um einen bestehenden Entwicklungsprozess und eine existierende Anwendung weiter zu entwickeln. Das Entwicklungsteam war zahlenmäßig begrenzt (V: 7, K: 4), die zeitlichen Vorgaben wenig großzügig. Um intern eine Verbesserung in der Selbstorganisation des Entwicklungsteams und extern eine gezielte Kommunikation des Projektfortschritts sowie eventueller Verzögerungen zu erreichen, wurde ein agiler Entwicklungsprozess mit einem „planning game“ gewählt, in dem regelmäßig eine Aushandlung und Priorisierung von Anforderungen zwischen Kunden und Entwicklern gefordert wurde. Um den Informationsbedürfnissen der Kunden bzw. der organisatorischen Vorgesetzten zu genügen, wurde ein „gläsernes Projektbüro“ mit einer „Brandmauer“ für das Entwicklungsteam installiert: Fortschritt der Entwicklung, aktuell definierte Anforderungen und deren Anpassungen sind jederzeit auch von außen einsehbar, Implementationsdetails und Leistungen einzelner Entwickler aber unsichtbar. Aus technischer Sicht wurde in beiden Fallbeispielen ein agiler Entwicklungsprozess implementiert, der viele Techniken des Extreme Programming aufgreift – das Mittel der Kommunikation nach außen waren dabei Stories, intern basierte die Planung auf Tasks.

Nutzerpartizipation wurde in beiden Fallbeispielen durch Workshops sowie ausführliche, mit Nutzern intensiv diskutierte Szenarien initiiert (ausführlicher wurde das Vorgehen in

[OF07, OF08, ON08] beschrieben). In frühen Phasen wurde nach Möglichkeit im Beisein und unter Mitwirkung der Nutzer ein Entwurf erarbeitet, in kurzen Workshops oder auf Basis von Mockups. Auch während der Umsetzung bemühten sich die Projektverantwortlichen um die Einbeziehung der Nutzer. Dabei wurde, soweit möglich, bestehende Infrastruktur genutzt, um dem „Raum“ für Co-Design einzurichten; in beiden Projekten wurde dafür ein Wiki genutzt, auf welches Kunden und Benutzer Zugriff erhielten und kritisch die dort abgelegten Anforderungen und Entwürfe überarbeiteten.

Grenzen partizipativen Designs

Trotz der vorstehend beschriebenen Bemühungen bleibt die Einbeziehung von Nutzern im Projekt V auf wenige Aspekte des Entwurfs beschränkt, im Projekt K konnte eine Nutzergruppe bisher gar nicht eingebunden werden. Unterschiedliche organisatorische Filter und Hindernisse können dafür verantwortlich gemacht werden:

- Interne Anwender verstehen sich selten als IT-Benutzer. Ihr Fokus liegt auf ihrer inhaltlichen Arbeit und nicht auf den dafür verwendeten Arbeitsmitteln. Software ist bestenfalls ein „effizienzsteigerndes Arbeitsmittel“, welches die tägliche Arbeit weniger lästig macht, kann aber auch, wie im Projekt K, als Symbol für ungewollte und lästige Aufgaben verstanden werden. Die Gestaltungsmöglichkeiten für die Anwender sind nicht gleich offensichtlich, da bereits ein altes System besteht und dieses auch nur teilweise ersetzt wird. Im Projekt V wurde dabei ein neues System neben das bestehende gestellt, im Projekt K ein Neusystem übergangslos in das alte integriert.
- Usability wird als Managementaufgabe begriffen, ein Produktverantwortlicher steht zuletzt auch für die Usability ein. Dies widerspricht Participatory Design in verschiedener Hinsicht: das Sicherheitsbedürfnis des Verantwortlichen führt dazu, „Amateuren“ wenig Vertrauen entgegenzubringen, seine fachliche Perspektive wirkt als Filter für Anforderungen und seine Designvorgaben können entscheidende Festlegungen erzwingen. Software kann zudem als Durchsetzungsmittel verstanden werden, um Geschäftsprozesse zu ändern, nicht immer im Sinne der zukünftigen Anwender.
- Auch agile Entwicklungsprozesse fokussieren auf die Organisation der Tätigkeitsform Softwareentwicklung. Damit werden Benutzer zwar nicht ausgeblendet, aber mehr als Informationsquelle denn als gleichwertige Handelnde definiert¹. Die Entwickler sind eine gewisse Gestaltungsfreiheit gewohnt und wünschen diese auch. Dabei werden in Einzelgesprächen oft sinnvolle individuelle Speziallösungen entworfen, die aber oft weder vom Design noch softwaretechnisch verallgemeinerbar sind.

¹ Im Scrum-Prozess gibt es ein Bild von Schwein und Huhn (Chicken & Pig), welches eigentlich auf die guten Hinweise unbeteiligter Managementebenen abzielt, aber auch Benutzern keine Stimme gibt: Rederecht auf den täglichen Meetings wird nur denen eingeräumt, die mit ihrer Haut für das Produkt herhalten müssen und nicht nur am Rande beteiligt sind (wie das Schwein beim englischen bzw. amerikanischen Frühstücksschinken).

- In der Entwicklung steht dem unvoreingenommenen Austausch mit Benutzerinnen oft auch Wissen um fachliche Vorgänge im Weg, das wesentlich detaillierter ist als das der Anwender (z.B. buchhalterische Auswirkungen von Buchungsaktionen), gepaart mit einem Qualitätsbewusstsein, welches die Veröffentlichung unfertiger Entwürfe verhindert. Dadurch entsteht leicht im kleinen Kreis zwischen fachlich versierten AnwenderInnen und Entwicklung eine Lösung, die anderen BenutzerInnen nur noch präsentiert wird.

Grenzöffnungsmaßnahmen und Zwischenfazit

In beiden Projekten war es nicht möglich, eine größere Anzahl von AnwenderInnen organisatorisch in das Projekt einzubinden. Daher spielen – in unterschiedlicher Form – Pilot-Benutzer und die Einschränkung der Partizipation auf bestimmte Anwendungsaspekte bzw. Benutzergruppen eine wesentliche Rolle:

Im Projekt V war die Strategie sehr erfolgreich, einer intern gut vernetzten Anwenderin eine inoffizielle verantwortliche Rolle im Projekt zu verschaffen; sie konnte als Mittlerin auch IT-scheue KollegInnen mit in die Diskussion ziehen und Kommunikationsbrücken bauen. Als Nachteil dieses Vorgehens wurde dabei in Kauf genommen, dass durch die Fokussierung der Partizipation auf diese Anwenderin ein neuer Filter geschaffen wurde. Aufgrund der zeitlich begrenzten Verfügbarkeit der Kontaktperson war ein Vorgehen notwendig, in dem gezielt als wichtig bzw. neu erachtete Interaktionen entworfen wurden; andere Teile der Anwendung wurden dann konsistent mit diesen Vorgaben ohne Nutzerpartizipation entwickelt. Voraussetzung für den organisatorischen Erfolg dieses Vorgehens war dabei eine hohe Flexibilität des Managements der Anwender und das Geschick der Projektleitung, bei Kunden und Produktverantwortlichen um Einverständnis für dieses inoffizielle Vorgehen zu werben.

Im Projekt K folgte auf eine mit Szenarien und Papierprototypen durchgeführte Designstudie ein enger Austausch mit den fachlich versierten Anwendern in der Hauptverwaltung. Neben regelmäßigen Designworkshops führt die regelmäßige Auslieferung von Softwareprototypen zu einer engen Kopplung der Designvorschläge und ihrer Umsetzung. Eine Partizipation mit BenutzerInnen aus den Kindertagesstätten konnte dagegen nicht etabliert werden: Produktverantwortlichkeit und Projektleitung waren zu eng verbunden, unfertige Lösungen sollten nicht gezeigt werden, um den guten Ruf des Produktes nicht zu beschädigen; die Benutzer nahmen die fatalistische Haltung von Endverbrauchern ein. Um mögliche Fehler im Design zu entdecken, wurde auf eine intensive Einbindung der Schulungsleitung bzw. des Supports für die Anwender geachtet und konventionelle Techniken wie Benutzertests eingesetzt.

Auch vor dem Hintergrund der zeitlichen und organisatorischen Vorgaben und der daraus entstehenden Einschränkungen sind beide hier geschilderten Projekte erfolgreich und konnten im praktischen Einsatz eine deutliche Verbesserung der Akzeptanz erzielen bzw. haben bisher gutes Feedback bekommen. Dennoch bleiben – auch ob der Unsicherheit, ob die beschriebenen Projekte noch unter den Begriff Participatory Design fallen –

Fragen offen: Inwieweit sind die aufgetretenen Probleme verallgemeinerbar? Welche grundsätzlichen Lösungswege lassen sich finden? Wie kann der in der Forschung formulierte Anspruch an PD und „Open Design Spaces“ relativiert werden?

Quellenangaben

- [BA04] Beck, K.; Andres, C.: Extreme Programming Explained: Embrace Change, Second Edition, Addison-Wesley, 2004.
- [BV95] Braa, K.; Vidgen, R.: Action Case – Exploring The Middle Kingdom in IS Research Methods. In: Proceedings of Third Decennial Conference Computers in Context: Joining Forces in Design, Aarhus, Denmark, 1995; S. 50-60.
- [OF07] Obendorf, H.; Finck, M.: Szenariotechniken & Agile Softwareentwicklung. In: Tagungsband Mensch & Computer 2007; S. 19-28.
- [OF08] Obendorf, H.; Finck, M.: Scenario-Based Usability Engineering Techniques in Agile Development Processes. In: Extended Abstracts CHI 2008; S. 2159-2166.
- [ON08] Obendorf, H.; Naumann, R.: Extreme Gebrauchstauglichkeit. In: Tagungsband Usability Professionals 2008; S. 233-237.