

Die unverantwortliche Leichtfertigkeit der Benutzerbeteiligung

Geschichten aus der Praxis

Peter Ansorge, Uwe Haupt

Wer DV-Profis nach Benutzerbeteiligung fragt, erntet Erstaunen darüber, dass die Frage überhaupt gestellt wird: Natürlich stehe man ständig im engen Kontakt mit dem Kunden, der Anwender lege schließlich die Anforderungen fest usw. Genaueres Nachfassen legt dann Unklarheiten offen. Es beginnt auf der Ebene der Begriffe: Zwischen Benutzer (dem mit der Software arbeitenden Menschen) und dem Anwender (der über den Einsatz entscheidenden Organisation) wird in der Regel nicht unterschieden. Neben den sauber strukturierten Projektplänen und vereinbarten Entscheidungsstrukturen wirkt die Benutzerbeteiligung als knetbare Masse. Es fehlen klare Vorstellungen darüber, was Benutzer sinnvollerweise entscheiden können und was nicht, und an die Stelle eines systematischen Methodeneinsatzes tritt das Prinzip „Reden wir mal drüber“. Und außerdem will man noch nicht viel von bevorstehenden Rationalisierungen offenbaren.

Ein zugegebenermaßen subjektiver Blick in die Praxis der Softwareentwicklung soll einige häufig anzutreffende Strukturen der Benutzerbeteiligung aufzeigen, die keineswegs zu besserer Software und verantwortlicheren Entscheidungen führen.

„Wie hätten Sie's denn gern?“

Benutzervertreter sollten über die Ausstattung mit mobilen Computern mitentscheiden. Mit hohem Engagement entwickelten die Beteiligten Lösungsvorschläge. Unterschiedliche Positionen standen sich unversöhnlich gegenüber – Tastatur oder Stifteingabe, große oder kleine Geräte. Über mehrere Tage wurde in zermürenden Sitzungen um die beste Lösung gerungen – ohne Ergebnis. Als die fraglichen Systeme besichtigt, angefasst und unter Realbedingungen – hier im fahrenden Auto – getestet werden konnten, stellten sich sämtliche zuvor vertretenen Standpunkte als hinfällig heraus. Innerhalb einer halben Stunde war eine tragfähige, allgemein akzeptierte Auswahl getroffen.

Die Diskussionen am grünen Tisch waren offenkundig substanzlos. Im hier skizzierten Fall der Hardwareauswahl war es einfach, eine Entscheidungsgrundlage zu schaffen. Die Hardware musste nur bereitgestellt werden. Schwieriger ist die Situation bei der Softwareerstellung, wenn die jeweilige Software noch gar nicht existiert. Die in der Literatur propagierte Lösung, mit Prototypen zu arbeiten, hat noch keine weite Verbreitung gefunden. Der Grund dafür liegt nicht nur im scheinbar hohen Aufwand. Vielfach scheuen sich Projektleiter, nach 10% der Projektlaufzeit nahezu komplett erscheinende Prototypen zu präsentieren, weil dann kaum zu vermitteln ist, dass noch 90% der Arbeit ungetan ist. Langwierige, wenig produktive Besprechungen lassen sich besser verkaufen.

Berufsbenutzertum

Ein komplexes System zur Verwaltung und Abrechnung staatlicher Leistungen sollte neu entwickelt werden. In der Behörde war Benutzerbeteiligung Tradition, schon bei der Entwicklung der Vorgängerversion vor mehr als 15 Jahren waren – damals eine Novität – Benutzer an der Entwicklung beteiligt. Wieder wurde ein Benutzerbeirat eingesetzt, der die Interessen der Benutzer in die Entwicklung einbringen sollte. Dazu wurden erfahrene Kräfte ausgewählt, die größtenteils schon bei der Entwicklung der Vorgängerversion als Benutzervertreter fungierten. Ein Teil der Benutzervertreter hatte in den letzten Jahren noch andere Entwicklungsprojekte begleitet, ein anderer Teil war in der DV-Schulung tätig gewesen.

Das Scheitern des Projektes war damit vorprogrammiert. Diese alten Hasen hatten seit vielen Jahren nicht mehr an den Arbeitsplätzen gearbeitet, an denen die Software eingesetzt werden sollte. Sie kannten die Anforderungen der Praxis und der Benutzer nicht mehr. Aufgrund ihrer DV-Erfahrung waren sie gleichzeitig beauftragt, die Interessen der Behörde – also des Anwenders – zu vertreten. Nicht genug, dass diese „Berufsbutzer“ ihre eigentliche Aufgabe nicht erfüllen konnten, ihre reine Existenz verhinderte die Einbeziehung „echter Benutzer“. Letztendlich wurde das Projekt erfolglos abgebrochen.

Berufsbutzern geht der Bezug zur Praxis, den sie eigentlich herstellen sollen, im Lauf der Zeit verloren. Stattdessen wirken sie als Trennwand zwischen Entwicklern und Benutzern. Ihr professioneller Umgang mit DV hat mit dem naiven Zugang eines Benutzers ohne tiefergehende DV-Ausbildung nichts mehr zu tun. Schließlich entsteht oft eine starke emotionale Identifikation mit dem zu entwickelnden DV-System, so dass die eigentliche erforderliche Distanz und Kritikfähigkeit vollends schwindet.

Die Tätigkeit als Berufsbutzer stärkt die Fähigkeit, sich in Gremien zu artikulieren und Positionen durchzusetzen, während einfachste Benutzungsprobleme zunehmend aus dem Blickfeld verschwinden. Lange unfruchtbare Sitzungen sind nicht untypisch für derartige Projekte.

... wörtlich genommen

Benutzer entwickeln schnell Vorstellungen darüber, wie „ihr System“ aussehen soll. Typisch sind die Situationen, in denen Benutzer spezielle technische Lösungen bis hin zu einzelnen Schaltflächen fordern. Schließlich kennt man sich mit Computern nicht zuletzt aus der häuslichen Anwendungen aus. Wenn Software-Entwickler aus falsch verstandener „Benutzerfreundlichkeit“ diese Forderungen wörtlich nehmen und umsetzen, werden sie ihrer Verantwortung nicht gerecht. Das Entwickeln einer oder mehrerer technischer Lösungen ist Aufgabe der Software-Entwickler. Nicht selten fordern spätere Benutzer „Hier muss ein Knopf“ hin. Gestalter seien vor solchen Forderungen gewarnt, hier ist die Erkundung der generischen Anforderung erforderlich, z.B. eine Detailsicht einsehen und ändern zu können. Derartige *technische* Forderungen sind zurückzuführen auf *fachliche* Anforderung – nur hierfür sind Benutzer Experten. Vorgefasste Lösungsformulierungen können gute Hinweise auf Bedarfslagen geben, bedürfen aber der Übersetzung und Anpassung an konkrete Software, am besten illustriert mit ähnlichen Lösungen bei bereits bestehenden Programmen.

Die Abgrenzung ist oft mit Konflikten verbunden. Werden – wie so häufig – diese Konflikte nicht ausgetragen, sondern technische Lösungsvorschläge der Benutzer „wörtlich“ umgesetzt, dann reden Benutzer über technische Lösungen und Entwickler kümmern sich um die Modellierung der zugrunde liegenden Geschäfts- und Arbeitsprozesse. Kurz: Jeder redet über genau das, was er nicht kennt und beherrscht. Vorgefasste Meinungen werden tradiert, Erkenntnisgewinn ist nicht festzustellen, das Ergebnis ist dann entsprechend.

Benutzer als General-Exkulpator

Fehler und Unzulänglichkeiten in fertigen Programmen werden oft achselzuckend mit der Bemerkung abgetan, „das hat der Benutzer so gefordert“. Hier wird Benutzerbeteiligung mit dem Abschieben von Verantwortung verwechselt. Aller Erfahrung nach können „echte Benutzer“ sehr fundierte Bewertungen vornehmen. Allerdings gelingt dies nur, wenn sie durch die Vorlage alternativer Lösungsmöglichkeiten entscheidungsfähig gemacht werden. Sie können detaillierte Hinweise auf Schwächen geben, wenn sie in Usability-Tests konkrete und praxisnahe Aufgaben bearbeiten. Bereits die Bearbeitung kleinster Testaufgaben offenbaren nicht selten durchgängige und systematische Probleme.

Die Verantwortung, Lösungsalternativen auf dem Stand der Technik zu präsentieren und angemessene Test- und Bewertungsverfahren einzuführen, liegt bei den DV-Spezialisten –, nie bei Benutzern.

Ein weiterer Mechanismus zur Verschiebung von Verantwortung wird in jüngster Zeit durch moderne, hochgradig konfigurierbare Software ermöglicht. Stolz präsentierten die Entwickler die äußerst flexiblen Einstellungsmöglichkeiten in einer Tabellendarstellung. Problemlos waren Spalten ein- und auszublenden, Umsortierungen vorzunehmen etc. Nach einer Vorführung waren Benutzer und Anwender ob solcher Flexibilität beeindruckt und veranlassten die Freigabe, scheinbar ermöglichte ihnen die Software schlichtweg „alles“.

Erst im Einsatz stellte sich heraus, dass die für die Arbeit erforderlichen Daten nicht in eine Tabellenzeile passten, wie immer sie auch angeordnet wurden. Die Anzeige in Tabellen war für die Aufgabe ungeeignet. Die Entwickler hatten Einstellungsmöglichkeiten demonstriert ohne nachzuweisen, dass es zumindest *eine* praktisch sinnvolle Einstellungsmöglichkeit gab.

Festgemauert in der Erden ...

In der Nutzungsphase, während der Benutzer besonders fundierte Verbesserungsvorschläge formulieren können, wird Software extrem starr. Zum einen ist dies sachlich nachvollziehbar, weil grundlegende Designentscheidungen zu diesem Zeitpunkt nicht mehr revidiert werden können. Zum anderen grassiert im Umgang mit laufender Software eine Bürokratisierung, die durchaus fragwürdig ist.

Ein einfacher Änderungswunsch eines Benutzers wurde zunächst daraufhin geprüft, ob durch das geltende Service-Level-Agreement zwischen IT- und Fachabteilung abgedeckt ist oder nicht. Da dies nicht ohne weiteres entschieden werden konnte, wurde ein „Trouble-Ticket“ geschrieben und in die korrekte Trouble-Ticket-Queue eingestellt. Aufgrund der ungeklärten Situation war eine Vorlage bei der nächsten Jahresbudgetplanung notwendig – immerhin musste „ein Paket von mehr als 40.000 Zeilen angepackt“ werden. Nur wenige Monate später erfolgte ohne große Diskussion die Freigabe in der Jahresbudgetplanung. Die Umsetzung der Benutzeranforderung – die Korrektur eines Tippfehlers in einem Textbaustein – war dann sehr schnell erledigt.

Gut, dass wir nicht nur drüber geredet haben ...

An der Notwendigkeit und Richtigkeit intensiver Benutzerbeteiligung besteht kein Zweifel. Nur ist dies eben entgegen einer verbreiteten Auffassung weder „leicht“ noch hemdsärmelig zu bewerkstelligen. Nicht nur die in diesem Zusammenhang so oft beschworene *soziale* Kompetenz spielt eine entscheidende Rolle, sondern auch die *technische* Kompetenz zur schnellen Präsentation von Lösungsalternativen auf dem Stand der Technik und die *methodische* Kompetenz, angemessene Tests, wie z.B. Usability-Tests, durchzuführen.

Benutzerbeteiligung setzt „echte“ Benutzer *und* beurteilbare Entscheidungsalternativen z.B. in Form von Entwürfen, Prototypen oder Vergleichssystemen voraus. Benutzer sprechen nicht die gleiche Sprache wie Entwickler, deswegen resultieren die brauchbarsten Hinweise zur Software-Gestaltung nicht aus langem Reden oder ermüdenden Anforderungspapieren, sondern schlichtweg aus der Beobachtung und Protokollierung des Umgangs mit bereits fertiggestellten Komponenten – und wenn es Interaktionen mit dem Finger auf einer Skizze sind. Die schlichte Methode *mSm* („machen Sie mal“) als Prinzip der Direkt- und Unmittelbarkeit: Reden ist Silber, ...

Unsystematische Benutzerbeteiligung führt dagegen zu Unwägbarkeiten und erhöhtem Aufwand, aber nicht zu besseren Systemen. Sie diskreditiert diesen wichtigen Ansatz.

Stand: 15.01.2004

Die Autoren Peter Ansorge und Uwe Haupt sind Geschäftsführende Gesellschafter der akziv gmbh in Bremen. Die akziv gmbh prüft die Ergonomie von Software und bietet weitere Dienstleistungen im Bereich Usability-Engineering und Software-Ergonomie an.

Weitere Infos unter: <http://akziv.de>