

Beteiligung der Benutzer – nicht nur positiv?

Die ›Beteiligung der Betroffenen‹ ist gerade für die Mitbestimmung so etwas wie eine ›heilige Kuh‹. Geschlachtet werden muss sie nicht, aber ein kritischer Blick kann auch nicht schaden.

WENN ES ETWAS gibt, das im Hinblick auf ›gute‹ Software-Entwicklung unumstritten scheint, dann ist das wohl der Grundsatz, dass Software nur dann praxisgerecht und ›gebrauchstauglich‹ werden kann, wenn die (späteren) Benutzer an ihrer Entwicklung möglichst von Beginn an beteiligt waren (siehe Nedler: ›Software-Usability – eine Prüfung, die sich lohnt‹ in cf 5/04 ab Seite 14). Und bei so viel Einmütigkeit ist es ja geradezu erfrischend, wenn dann plötzlich – wie erst kürzlich in der Fachzeitschrift ›Fiff-Kommunikation‹ 1/04 geschehen – ein Artikel auftaucht mit dem provozierenden Titel: ›Die unverantwortliche Leichtfertigkeit der Benutzerbeteiligung‹. Geschrieben wurde er von Peter Ansorge und Uwe Haupt, beide renommierte Spezialisten für Software-Usability [→], also Vertreter jener Zunft, die immer besonders vehement für eben jene ›Beteiligung der Betroffenen‹ streitet.

Nun, wie es ja fast immer der Fall ist, wird auch hier nicht so heiß gegessen, wie gekocht wurde, und so geht es – genau genommen – auch gar nicht um ›die‹ Benutzerbeteiligung in Bausch und Bogen, sondern um deren gezielte und systematische Durchführung.

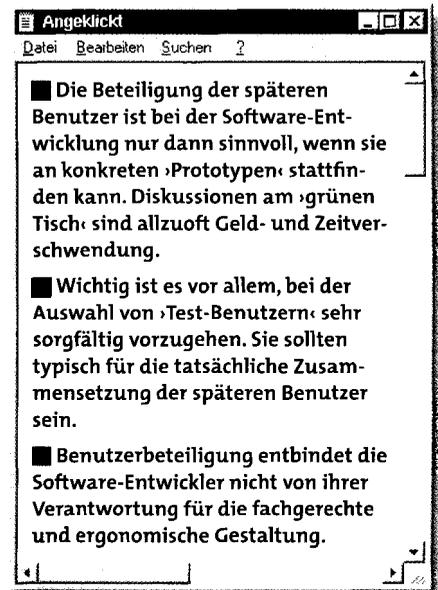
Das beginnt mit einer Klärung der Begriffe: So sind für viele Software-Entwickler (und nicht nur für diese) ›Anwender‹ und ›Benutzer‹ ein und dasselbe. Es wird nicht getrennt zwischen dem Kunden, der die gewünschten Eigenschaften einer in Auftrag gegebenen Software-Anwendung mehr oder weniger genau spezifiziert, dem Anwender also, und denjenigen, die später diese Software in ihrem Berufsalltag benutzen werden, eben den Benutzern. Und schon diese fehlende ›Trennschärfe‹ führt dazu, dass in der Praxis oft nicht unterschieden wird zwischen dem, was einerseits der Anwender an Anforderungen formulieren muss und an welchen Entscheidungen andererseits die späteren Benutzer sinnvoll zu beteiligen sein könnten. Oder wie Ansorge und Haupt es formulieren: »Neben den sauber strukturierten Projektplänen und vereinbarten Entscheidungsstrukturen wirkt die Benutzerbeteiligung als knetbare Masse [...] und an die Stelle eines systematischen Methodeneinsatzes tritt das Prinzip ›Reden wir mal drüber‹.«

Auf diesem Hintergrund ist denn auch die zentrale Behauptung, dass Benutzerbeteiligung keineswegs immer und in jedem Fall »zu besserer Software und verantwortlichen Entscheidungen führen« müsse, schnell und besser zu verstehen. Und dieses Verständnis verstärkt sich

noch, wenn man mit den beiden Autoren einen Blick auf die Praxis der Benutzerbeteiligung wirft ...

Keine Diskussionen am grünen Tisch!

Diskussionen am grünen Tisch bleiben auch dann ›Diskussionen am grünen Tisch‹, wenn sie mit den künftigen Benutzern einer geplanten Software oder auch eines kompletten IKT-Systems geführt werden. Benutzerbeteiligung ist also erst dann sinnvoll, wenn es bereits etwas zu begucken, anzufassen, auszuprobieren gibt. Leider hat sich jedoch das Prinzip des ›Prototyping‹ [→] noch längst



nicht allgemein durchgesetzt. Offensichtlich, so Ansorge und Haupt, »scheuen sich Projektleiter, nach 10 Prozent der Projektlaufzeit nahezu komplett erscheinende Prototypen zu präsentieren, weil dann kaum zu vermitteln ist, dass noch 90 Prozent der Arbeit ungetan ist. Langwierige, wenig produktive Besprechungen lassen sich besser verkaufen.«

Vorsicht vor ›Berufsbenutzern!‹

Was wir aus vielen anderen Bereichen der Interessenvertretung und Lobby-Arbeit kennen, scheint es bisweilen auch bei der Benutzerbeteiligung zu geben. Vor allem in großen Unternehmen oder auch Behörden, in denen häufiger einmal Software-Entwicklungen mit Benutzerbeteiligung stattfinden, gibt es die

Neigung, immer dieselben ›erfahrenen‹ Menschen als ›Testpersonen‹ einzusetzen – es gibt sogar einen Fachausdruck dafür: ›Power-User‹.

Dabei muss es gar nicht unbedingt so sein wie in dem von Ansorge/Haupt gewählten Beispiel, dass nämlich diese Benutzer schon längst nicht mehr an den Arbeitsplätzen tätig sind, an denen die neue Software-Version zum Einsatz kommen soll. Bereits wenn die ausgewählten Test-Benutzer im Hinblick auf Erfahrungen und Fachwissen Überdurchschnittliches mitbringen, sind sie nicht mehr repräsentativ für die eigentliche Zielgruppe. Deswegen muss ja noch nicht zwingend der ›DaU‹ (der ›Dümmste anzunehmende User‹) das Idealbild des Test-Benutzers sein. Wichtig wäre es vielmehr, dass die beteiligten Benutzer im Hinblick auf individuelle Erfahrungen, Fachkenntnisse und Fertigkeiten ein möglichst gutes Spiegelbild der Gesamt-Benutzerschaft abgeben.

Die Benutzer nicht zu ernst nehmen!

Benutzer – so haben Ansorge/Haupt festgestellt – entwickeln sehr schnell sehr genaue Vorstellungen, wie ›ihr‹ System auszusehen habe. Benutzerbeteiligung ist aber weit mehr als nur das Ausführen von Benutzerforderungen (»Hier muss ein Knopf her!«). Solche Forderungen mögen fachlich gut begründet sein, zwingend ist das aber nicht so. Vielmehr spielen da höchst individuelle Vorlieben eine Rolle, nicht allzu selten auch begrenzte Fachkenntnisse oder festgefahrene, nicht hinterfragte Gewohnheiten.

Es kommt also darauf an, den Software-Entwicklungsprozess als einen Dialog mit klar verteilten Rollen und Verantwortlichkeiten zu gestalten: Die Benutzer bringen ihre Kenntnisse von Arbeits- und Geschäftsprozessen ein, die Entwickler sorgen für die in sich stimmigen technischen Lösungen. Dies kann und sollte sogar zu Konflikten führen, die aber keinesfalls dadurch ›gelöst‹ werden dürfen, dass technische Lösungsvorschläge der Benutzer ›wörtlich‹ umgesetzt werden.

Die Verantwortung haben immer die Entwickler!

Man muss es wohl glauben, wenn Ansorge und Haupt berichten, dass Ungereimtheiten eines fertigen Software-Produkts oft mit der Behauptung entschuldigt werden, das hätten »die Benutzer so gewollt«. Richtig jedoch ist, dass Benutzer vor allem dann gute Hinweise auf Schwächen in der Software-Gestaltung geben können, wenn sie konkrete Testaufgaben zu bearbeiten haben und dafür praktikable Lösungsalternativen geboten bekommen. Die Verantwortung für diesen Prozess und damit auch für die Gestaltung des Endprodukts liegt deshalb immer bei den IKT-Spezialisten, niemals bei den Benutzern.

Interessant ist in diesem Zusammenhang auch der Hinweis, dass sogar die generell ja richtige Forderung nach Flexibilität (›Individualisierbarkeit‹) bei den Systemeinstellungen für ein Abschieben von Verantwortung missbraucht werden kann: Wer dem Benutzer ›jede‹ Möglichkeit der individuellen Systemgestaltung einräumt, könnte versucht sein, über wirklich praktikable Lösungswege gar nicht mehr nachzudenken ...

Benutzerbeteiligung auch nach der Fertigstellung!

Man kann es vermutlich drehen und wenden wie man will: Auch wenn die Software-Entwicklung noch so fundiert und systematisch und die Benutzerbeteiligung noch so gut organisiert war, viele Benutzungsprobleme werden sich erst im ›richtigen‹ Praxisbetrieb herausstellen. War der vorangegangene Entwicklungsprozess gut gestaltet, wird es sich dabei vor allem um Detailfragen handeln. Und in der Tat: Grundlegende Design-Entscheidungen werden sich zu diesem Zeitpunkt nicht mehr verändern lassen. Das muss aber durchaus nicht heißen, dass sich nicht begrenzte Detail-Prozesse mit vertretbarem Aufwand ›nachträglich‹ noch verändern ließen. Nach Meinung von Ansorge und Haupt sind es denn auch weniger die technischen oder wirtschaftlichen Notwendigkeiten, die eine fertig gestellte Software plötzlich ›starr‹ werden lassen, sondern oft allzu bürokratische Prüfungs- und

Genehmigungsverfahren. Hier zu flexibleren Lösungen zu kommen, ist also ebenfalls eine Aufgabe für eine systematische und sinnvolle Benutzerbeteiligung.

Ob es im Zuge von Software-Entwicklungsprozessen Benutzerbeteiligung gibt oder nicht und wie sie im Einzelnen gestaltet wird, ist ganz klar eine Frage der Mitbestimmung durch Betriebs- und Personalräte. Es liegt also auch, vielleicht sogar vor allem in ihrer Hand, dafür zu sorgen, dass eingefahrene Strukturen auf diesem Gebiet hinterfragt werden und dass nach neuen Lösungen gesucht wird – wobei sicher auch (Stichwort: ›Berufsbenedict‹) eine gewissen Bereitschaft zur Selbstkritik nicht schaden kann ...



Quelle:

Peter Ansorge, Uwe Haupt: Die unverantwortliche Leichtfertigkeit der Benutzerbeteiligung / Geschichten aus der Praxis; in ›Fiff-Kommunikation 1/04 ab Seite 52

⇒ Prototyping = Entwickeln von Produkten durch Prototypen = ein in der Software-Entwicklung besonders schwer umzusetzendes Prinzip, denn ein zu Testzwecken hergestellter voll funktionsfähiger Prototyp wäre die fertige Software, jede gravierende Veränderung wäre also mit komplettem Umprogrammieren und entsprechenden Kosten verbunden; deshalb sind Software-Prototypen keine funktionsfähige Software (mit funktionsfähiger Datenspeicherung, Berechnungen usw.), sondern ›Potemkinsche Dörfer‹ – abgebildet wird lediglich die Benutzungsoberfläche, auf der die Bedienung später zu programmierender Funktionen simuliert werden kann

⇒ Usability (Gebrauchstauglichkeit) = Bezeichnung dafür, dass eine Software in der Praxis auch tatsächlich so funktioniert, dass ihre Benutzer im Alltag damit gut zurecht kommen; um dies zu gewährleisten, werden unter anderem Tests mit ›typischen‹ Benutzern durchgeführt