

Partizipative Softwareentwicklung

Harald Raum, Sabine Baronick, Dresden

Zusammenfassung

Auch die Berufsberatung orientiert sich auf computergestützte Beratungsformen. Die Beobachtungen bei der Entwicklung eines Programms zur Selbstinformation für Schüler boten Gelegenheit, die Möglichkeiten und Effekte einer auf Nutzerpartizipation beruhenden Entwicklungsstrategie zu demonstrieren. Befragungs- und Beobachtungsdaten belegen, wie die Modifikationen des textlichen Aufbaus und der didaktischen Struktur zu einem immer nutzerfreundlicheren Programm führen.

1. Anliegen

Bei der Entwicklung von Software ist die Nutzerbeteiligung unerlässlich, weil ein theoretisch befriedigendes Modell fehlt, an dem sich der Softwareentwickler über die Nutzungserfordernisse orientieren könnte. Das betrifft besonders die aufgabenspezifischen Handlungsabläufe, aber auch viele der übergreifenden Forderungen zur Schnittstellen- und Dialoggestaltung.

Ungeachtet dessen wird die Nutzerbeteiligung von den Softwareentwicklern häufig unterschätzt und fehlverstanden. Das drückt sich in verschiedenen Bewältigungsmustern aus, von denen die drei folgenden (rein oder kombiniert) am häufigsten anzutreffen sind:

- Das "Stellvertreterprinzip" läuft darauf hinaus, den tatsächlichen Programmnutzer durch Vertreter (Arbeitswissenschaftler oder Meister) zu ersetzen und nur mit diesen zu kooperieren.
- Das "Lokalisierungsprinzip" besteht darin, die Zusammenarbeit auf wenige Aktionen zu begrenzen. Unter Bezug auf das überlebte "Wasserfall-Modell der Systementwicklung" (BOEHM 1983) und den "linearen Software-Lebens-Zyklus" (MELZER 1989) wird der Nutzer meist nur zur "Anforderungsanalyse" herangezogen.
- Das "Mittäter-Prinzip" schließlich degradiert die Nutzerbeteiligung zu einem "psychologischen" Mittel der Akzeptanzsicherung. Es kommt nur zur Scheinkooperation. Der Entwickler hofft auf die spätere widerspruchslose Abnahme des Programms.

Das folgende Fallbeispiel soll zeigen, wie echte Nutzerbeteiligung realisiert und mit welchen Effekten gerechnet werden kann.

2. Ein Fallbeispiel

2.1. Konkreter Anlaß der Programmentwicklung

Die Berufsberatungszentren sind an Auskunftssystemen für die Selbstinformation der Schüler über Ausbildungsmöglichkeiten und zur Unterstützung der Arbeit der Berufsberater interessiert. Daraus entstand der Auftrag, ein solches Programm zu entwickeln.

1. Es soll die Informationen anbieten, die für den Schüler bei seiner Berufswahl interessant bzw. unerlässlich sind.
2. Der Schüler soll am Ende Berufe bzw. Ausbildungsstellen angeboten erhalten, für die er sich erfolgreich bewerben kann.
3. Die Abarbeitung des Programms soll die Einsicht fördern, daß man sich bei der Berufswahl nicht nur von Wunschträumen, sondern auch von den realen Gegebenheiten leiten lassen sollte.

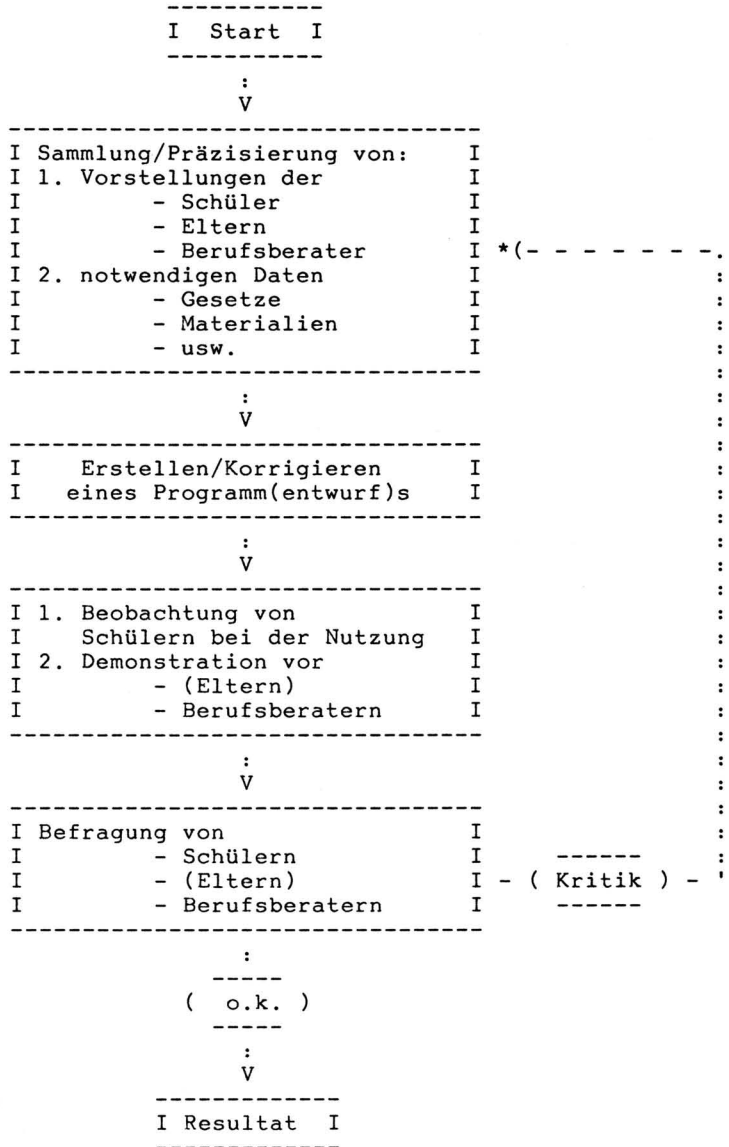


Abb. 1: Iterative, rückgekoppelte Verbesserung des Programms
HAUPT durch Nutzerbeteiligung (BARONICK 1989)

Um diese Ziele zu erreichen, war es nötig, Schüler, Eltern und Berufsberater in die Erarbeitung einzubeziehen und die Nutzerfreundlichkeit des Programmes iterativ zu verbessern.

2.2. Programmerarbeitung im iterativen Prozeß

Die Programmentwicklung folgte grundsätzlich dem in Abb. 1 dargestellten Algorithmus.

Die jeweils erste Aktivität innerhalb eines Iterationszyklus betraf die Sammlung und Ordnung von Informationen, die die Datenbank enthalten sollte, und von Vorstellungen über den Dialog.

- Berufsberater, Schüler und Eltern wurden befragt, die Beschaffbarkeit der Informationen überprüft und Klassifikationsvorschläge gesammelt. Spätere Iterationen dienten der Präzisierung dieser Forderungen.
- Besonders um das "pädagogische Konzept" entbrannte der Streit und konzentrierte sich auf die Frage der "freien Wahl der Informationsabfolge". Die Berufsberater forderten, bestimmte Aspekte nicht ignorieren zu lassen (z.B. individuelle Voraussetzungen, territoriale Möglichkeiten), auch wenn der Bewerber dafür wenig Verständnis aufbringen sollte. Außerdem spielte das Problem der Textgestaltung (Formulierung und Layout von Informationen und Handlungsanweisungen) eine große Rolle.

Als nächstes erfolgte die Umsetzung in ein Programm. Der Streit um das pädagogische Konzept führte zur Variantenentwicklung:

- a) total freie, menügestützte Auswahl der Auskünfte
- b) Zwangsabfolge nach den Vorschlägen der Berufsberater oder
- c) entsprechend einer Vorsortierung durch Schüler und Eltern
- d) Kompromiß (erzwungener Anfangs- und menügestützter Hauptteil)

Erst nach der 4. Iteration fand man zur Form (d), die eine "geleitete" Suche mit der Wahlmöglichkeit verband.

Merkmale/Item	2. Etappe	3. Etappe	4. Etappe	5. Etappe
=====				
1. STRATEGISCHES KONZEPT (Variantenangebot)	- 1 Menü	* 3 Menü Abfolge BB Abfolge Sch	- 3 Menü Abfolge BB Abfolge Sch	* 1 Kombination von fixiertem Start u. Menü
2. INHALT/VOLUMEN	- 20 Berufe	* 80 Berufe	* 165 Berufe	- 165 Berufe
3. VERBALISIERUNG				
- Texte	- Urform	* verbessert	* verbessert	* verbessert
- Schlußinstruktion	- Urform	- Urform	* erweitert	* präzisiert
KATEGORISIERUNG				
- Berufsrichtungen	- 29 Klassen	* 17 Klassen	- 17 Klassen	* 17 Kl (+Hilfen)
- Spezialis-Richtgn.	- als Ausbild.- besonderheit	- als Ausbild.- besonderheit	- als Ausbild.- besonderheit	* als Abschluß- information
TEXTANORDNUNG				
- Berufsrichtungen	- mehrspaltig	* einspaltig	- einspaltig	- einspaltig
- Lstgsvoraussetzgn	- 1 BS-Seite	- 1 BS-Seite	* 2 BS-Seiten	- 2 BS-Seiten
4. HILFEN				
- Berufsrichtungen	- ohne	- ohne	- ohne	* Entschdgs-Hilfe
- Lehrstellenangebot	- ohne	- ohne	* mit einfacher	* mit Hilfs-Seite
5. EINGABEMODUS				
- Antwortkundgabe	- "Ja"/"Nein"	- "Ja"/"Nein"	* teils "J"/"N"	* Markierung
- Eingabekontrolle	- ohne	- ohne	* realisiert	- realisiert
- Eingabenabforderung	- ungleichmäßig	- ungleichmäßig	* gleichmäßig	- gleichmäßig
- Eingabeabschluß	- erforderlich	- erforderlich	* selbsttätig	- selbsttätig
- Interpreterkommandos	- bei Fehlern	- bei Fehlern	* ausgeschaltet	- ausgeschaltet
=====				

Tabelle 1: Übersicht über die wichtigsten Veränderungen am Programmpaket im Verlaufe der iterativen Prozedur der Programmentwicklung (BB = Berufsberater, BS = Bildschirm, Sch = Schüler)
(ausgelassen: Programmtechnische Verbesserungen wie Dateisicherung, Rollbild u.ä.)

Das Programm wurde nun Schülern, Eltern und Berufsberatern zur ungebundenen verbalen Beurteilung vorgestellt und an Schülern praktisch erprobt. Die Erprobung erfolgte ab zweitem Iterationszyklus (1987 bis 1989) unter quasiexperimentellen Bedingungen an 188 Schülern der Klassenstufen 7, 8 und 9 im Rahmen von studentischen Praktika und Diplomarbeiten. Die Gruppen bildeten "anfallende Stichproben", wurden aber pro Alterstufe nach Notendurchschnitt, Konzentrationsfähigkeit, Computererfahrung und Geschlechterverteilung kontrolliert.

Beobachtet bzw. erfaßt wurden:

- Ungewollte Programmabbrüche ("Absturz")
- Fehleingaben
- Rückfragen / Bemerkungen
 - bezüglich der Instruktionstexte
 - zu Einzelaussagen (Items)
 - im Anschluß an Fehleingaben
- Lesezeiten pro Ausgabeeinheit
- Selbstaussagen über sachliche Schwierigkeiten bei der Abarbeitung des Programms (z.B. bei der Angabe von Tauglichkeitseinschränkungen des Bewerbers)
- Allgemeine Werturteile und Hinweise zum Programm

Die Befragungen und Beobachtungen führten dann zu der Entscheidung über eine neuerliche Iteration (Kritik) oder den Abschluß der Softwareentwicklung. Im vorliegenden Beispiel waren 5 Iterationszyklen erforderlich, ehe ein akzeptiertes Entwicklungsergebnis vorlag.

2.3. Ergebnisse der Nutzerbeteiligung

2.3.1. Veränderungen am Programm

Wenn man die in Tab.1 aufgelisteten Veränderungen am Programm zusammenfaßt, gelangt man zu einer quantitativen Bilanz der durch Nutzerbeteiligung erzielten Effekte (Tab. 2).

Veränderungen	Etappe				
	2	3	4	5	gesamt
Strategisches Konzept	1	1		1 (1)	3 (1)
Textgestaltung/Formulierungen	6	3	3 (1)	4 (3)	16 (4)
Einfügen von Hilfen			1	2 (1)	3 (1)
Eingabemodus	4		5	1 (1)	10 (1)
Gesamt	11	4	9 (1)	8 (6)	32 (7)

Tab.2: Übersicht über die Anzahl und Zielrichtung von Veränderungen während der Programmentwicklung (vgl. Tab 1)
 () = Wiederholte Änderungen
 Formulierungsänderungen an Instruktionen und Texten sind nur global berücksichtigt.
 Volumenänderungen des Datenpools sind weggelassen

Es wird deutlich, daß sich neben den Formulierungspräzisierungen das Gros der programmtechnischen Änderungen auf die allgemeine Form des Programmaufbaus (Konzept, Hilfen), die Dialogführung (Eingabemodus) und die Art und Weise der Strukturierung des Datenangebots (Textgestaltung) bezog.

Es zeigt sich auch, daß in 7 der 32 erreichten Änderungen erst mehrere Versuche zum gewünschten Ergebnis führten. Hauptgrund dafür war die Wandlung der Nutzervorstellungen, nicht deren unvollkommene Umsetzung durch den Softwareentwickler.

Interessant ist, daß sich entgegen landläufiger Erwartungen von Iteration zu Iteration kein deutlicher Trend zur Verminderung der Veränderungsanzahlen ergibt. Das unterstreicht die bleibende Bedeutsamkeit der Nutzerbeteiligung über den gesamten Entwicklungszeitraum hinweg.

2.3.2. Die Wirksamkeit der Programmänderungen

Die durch Nutzerbeteiligung erreichten Programmänderungen sind erst dann wirklich zu begrüßen, wenn sich ihr Nutzen auch "hart" nachweisen läßt. Eine höhere Zufriedenheit der Beteiligten steigert zwar die aktuelle Akzeptanz, ist aber eine "vergängliche" Größe und kann deshalb nicht alleiniger Maßstab für die Sinnfälligkeit der Programmänderungen sein.

Hier sollen die Verständlichkeit der Informationen und die Handhabbarkeit (Fehlerrobustheit, Selbsterklärungsfähigkeit) des Programms durch den Schüler exemplarisch herausgegriffen und deren Verbesserungen anhand der Beobachtungsdaten gezeigt werden.

Diese Daten belegen:

Die Verständlichkeit der Instruktionen und des (textlichen) Informationsangebots hat sich verbessert (Tab. 3). Rückfragen und Fehlhandlungen werden seltener. Rückfragen zu den Instruktionstexten tragen mehr und mehr nur noch rückversichernden Charakter.

	3. Etappe	4. Etappe	5. Etappe

Instruktionsrückfragen/Vp	2,2	1,2	1,9
davon:			
- Rückversicherungen %	21,4	33,8	42,6
- Mißverständnisse %	78,6	66,2	57,4
Itemrückfragen/Vp	0,9	0,8	0,3
Fehlhandlungen/Vp	1,8	0,8	0,3

Tab. 3: Veränderung der Rückfragebedürfnisse und der Fehlhandlungen der Schüler von Entwicklungsetappe 3 zu 5

Verständlichkeitsgewinne sind streng genommen nur auf die Gesamtheit aller zwischenzeitlichen Programmänderungen zurückzuführen. Trotzdem kann man am Beispiel der Textänderungen verdeutlichen, worauf die Nutzer besonderen Wert legten. Sie achteten auf die Anzahl und die Wahl der Worte. Je knapper und treffender eine Information formuliert werden konnte, um so größer war die Chance, daß sie von den Schülern auch verstanden wird. Tab. 4 zeigt das anhand der Lesezeiten.

Bildschirmseite	3. Etappe		4. Etappe		5. Etappe	
	Silben-	Lese-	Silben-	Lese-	Silben-	Lese-
	anzahl	zeit	anzahl	zeit	anzahl	zeit
	n	n/sec	n	n/sec	n	n/sec
höchste Silbenanzahl	291	3,0	262	4,0	188	4,4
beste Lesbarkeit	153	3,5	186	5,6	135	5,8
kleinste Silbenzahl	70	2,3	54	3,3	68	4,4
schlechteste Lesbkt	79	1,0	80	3,2	140	3,1
Mittelwert	127	2,3	137	4,2	130	4,2
Streuung	65	0,7	59	0,9	32	0,9
Streuung %	51	32	43	22	24	22

Tab. 4: Silbenanzahl (Silben/Bildschirmseite) und Lesegeschwindigkeit (Silben/Sekunde) für ausgewählte Bildschirmseiten der Programmvarianten der Entwicklungsetappen 3 bis 5

Man erkennt, wie von Etappe zu Etappe trotz etwa gleicher mittlerer Silbenanzahl/Bildschirmseite, aber ausgeglichenerer Packungsdichte die Lesegeschwindigkeiten wachsen und sich gleichmäßiger verteilen (Streuungen!).

3. Schlußfolgerungen und Nachbemerkenungen

Das Fallbeispiel zeigt einerseits die Nützlichkeit der Nutzerbeteiligung im Softwareentwicklungsprozeß, macht aber auch auf einige Probleme aufmerksam, die die eingangs beschriebenen Vorbehalte der Softwareentwickler nachvollziehbar machen.

1. Durch Nutzerbeteiligung werden nicht nur Zufriedenheitsgewinne erzielt. Nutzerbeteiligung führt auch zu solchen Programmveränderungen, die sich in "harten" Leistungsdaten niederschlagen. Die Nutzer verfolgen dabei intuitiv Gestaltungsgrundsätze, die zwar theoretisch aus der Literatur bekannt sind, sich aber kaum ad hoc umsetzen lassen.
2. Es zeigte sich, daß "Nutzer" ein Sammelbegriff für durchaus unterschiedliche Interessentengruppen sein kann, zwischen denen ein Kompromiß herbeigeführt werden muß. Hier könnten Gruppenprozeduren hilfreich sein. Der Softwareentwickler geriete jedoch in die Rolle eines Moderators oder gar in Interessenkonflikte (er soll die Forderungen der Gruppe ja umsetzen!) und könnte dadurch überfordert sein.
3. Nutzerwünsche unterliegen selbst Wandlungen im Prozeß der Befassung mit der entstehenden Software. Diese Dynamik mag zum Teil in der Inkonstanz der Bedürfnisse selbst begründet sein, ist aber wohl zuerst der tatsächlichen Umgangserfahrung geschuldet, die der Nutzer mit den (Zwischen-)Ergebnissen der Umsetzung seiner Wünsche macht. Es ist deshalb eine Illusion, Nutzerbeteiligung auf einmalige Vorbefragungen und Problemdiskussionen in der Phase der Aufgabenanalyse und des Erstellens eines Forderungskatalogs an das Endprodukt beschränken zu wollen. Auch der Nutzer braucht den Umgang mit Zwischenresultaten, um sich kompetent einbringen zu können.
4. Die Effektivität der Nutzerbeteiligung scheint von der Systematik der projektbegleitenden Zwischenauswertungen abhängig zu sein. Der Aufwand dafür gerät in die Nähe von Felduntersuchungen und macht die Mitarbeit z.B. eines Arbeitswissen-

schaftlers wünschenswert. Er könnte auch als Moderator wirksam werden und damit den Softwareentwickler entlasten.

5. Schließlich muß darauf verwiesen werden, daß die Geduld und Verfügbarkeit von Nutzern für partizipative Prozeduren nicht stillschweigend angenommen werden kann. Der Untersucher war zu großen Teilen seiner Zeit damit beschäftigt, Testpersonen zu gewinnen und Gesprächspartner zusammenzuführen. Die Vermittlung von Einsichten in die Nützlichkeit der Partizipation sollte sich daher nicht einseitig auf den Entwickler konzentrieren. Auch dem Nutzer sollte klar sein, daß partizipatives Design wichtig, zeitraubend und arbeitsintensiv ist.

Literaturverzeichnis

Baronick, S. (1989). Erprobung einer Methodik zur Softwaregestaltung. Fo.-ber., TU Dresden, WB Psychologie

Boehm, B. W. (1984). Seven Basic Principles of Software Engineering. Journ. Systems and Software, Heft 3

Melzer, W. (1989). User Participation in Software Design - Problems and Recommendations.

In: Teikari, V.; Hacker, W.; Vartiainen, M. (ed.): Psychological Task Analysis, Design, and Training in Computerized Technologies.

Otaniemi (Finnl.), Rep. No. 113, S. 109 - 120

Prof. Dr.sc. nat. Harald Raum, Dipl.-Psych. Sabine Baronick
Technische Universität Dresden
Fakultät für Naturwissenschaften und Mathematik
Abteilung Human- und Biowissenschaften
Institut für Psychologie
Mommsenstr.13
O-8027 Dresden