BEHAVIOUR & INFORMATION TECHNOLOGY, 1996, VOL. 15, NO. 4, 226–236

Don't underestimate the problems of user centredness in software development projects—there are many!

TORSTEN HEINBOKEL, SABINE SONNENTAG, MICHAEL FRESE, WOLFGANG STOLTE and FELIX C. BRODBECK

University of Giessen, Department of Psychology, Germany

Abstract. On the basis of a longitudinal field study of 29 commercial software development projects, the pros and cons of user centredness in software development were analysed. We looked at two concepts: user participation—an organizational device—involving a user representative in the team, and user orientation—a cognitive-emotional concept—which pertains to positive attitudes towards users. Both were found to be associated with project difficulties relating to process and product quality as well as overall project success. We suggest that the issue is no longer whether or not to involve users, but instead to develop a realistic understanding of the difficulties associated with user centredness.

1. Introduction

Within software engineering there is a growing interest in the notion of user centredness (Gould and Lewis 1983, 1985, Shackel 1985, Ulich 1993). In a weak form, user centredness involves consideration of users' needs, as for example in Norman and Draper's (1986) *User Centred System Design* or in design guidelines for the user interface (e.g. Brown 1989, Smith and Mosier 1986, Shneiderman 1992). In a strong form, user centredness implies active participation of users in the design process. Some researchers have suggested iterative design principles that allow the user to test the development at certain intervals (rapid prototyping approach, e.g. Boehm 1987, Budde *et al.* 1992, Floyd 1984, Jörgensen 1984). Others have proposed that there should be at least one user representative on the design team (Bjerknes *et al.* 1987, Briefs *et al.* 1983), preferably from the onset of the project (Williges *et al.* 1987).

User involvement methodologies span a continuum from consultative to representative and, at the most intensive level, consensual design (Mumford 1981). Differences between methodologies can be found with respect to the primary focus of the development (improved computer systems versus improved work places). However, two of the most prominent methodologies of user involvement, the Scandinavian Participatory Design approach and the North American Joint Application Design, inevitably link the development of high quality systems to continuous user participation in the design process (Carmel *et al.* 1993).

The arguments for the strong form of user centredness, involving the users in the design team, are convincing. One such argument stems from the fact that software designers usually are unfamiliar with the users' work tasks and are therefore unable to design useful tools for the users (Curtis 1988). Moreover, it can be argued that only users really know what is best for them, so only through their active participation can usable software be developed (Spinas 1990). Thus, both functionality and usability of software depend on a transfer of knowledge from users to software designers. As this knowledge transfer is a complicated process, and software designers often do not have the time, motivation, or prior knowledge to get to know the users' needs (Grudin 1991), what better way is there to establish a consistent flow of knowledge than to involve the users themselves in the process. Some authors have additionally argued that user involvement, specifically rapid prototyping approaches, lead to higher efficiency, as measured both in cost and in time (Baroudi *et al.* 1986, Bewley *et al.* 1983, Boehm *et al.* 1984, Gomaa 1983, Karat 1990, Mantei and Teorey 1988, Strohm 1991).

Interestingly, in spite of these arguments for a higher degree of user centredness in the design process, user participation and prototyping are rarely practised in commercial software development (Aschersleben and Zang-Scheucher 1989, Rosson *et al.* 1987). Why is user participation not more commonly applied in software development? One explanation may be that the process of software development is disturbed by user participation. New methodologies such as prototyping bring about changes in the developers' work situation (Budde *et al.* 1992). This may cause opposition and communication barriers. Moreover, developers claim that users lack computer knowledge and have little ability to describe their work tasks (Peschke 1986). On the other hand, users are often dissatisfied because they feel they have little influence and therefore might reduce their commitment to the design process. Consequently, there is some evidence that certain forms of intensive user involvement may lead to project goals being missed more often (Selig 1986). Despite a vast number of practices for implementing user involvement (Muller *et al.* 1993), there is no method which guarantees the successful implementation of user involvement (Kensing and Munk-Madsen 1993).

Moreover, there is little empirical proof of the superiority of user involvement in commercial software development (Ives and Olson 1984). Many of the earlier projects of participatory design were small-scale, did not intend to develop marketable software (Clement and Van den Besselaar 1993) and were funded by public authorities (Weltz and Ortmann 1992). Moreover, many findings on technological gains of user participation practices have been criticized for being *post-hoc* estimates lacking explication of their methodological basis (Carmel *et al.* 1993).

2. Goal of the study

We tested empirically whether or not user centredness is associated with problems in the software design process. We distinguish between *user orientation* and *user participation*. *User orientation* pertains to the software developers' individual value of producing software for the user and thinking of the user during the development process. Thus, user orientation is an attitudinal measure of user centredness. In contrast, *user participation* indicates that at least one user representative is actually part of the software development team. Thus, we were concerned with both the organizational implementation (user participation) and the individual cognitive-emotional representation (the attitude of user orientation) of user centredness. These two concepts do not have to be independent of each other, for example, high user orientation can result in user participation. However, we assume that they should not be highly related, because the organizational implementation of user centredness is not necessarily reflected in software designers' attitudes towards users.

We wanted to test the alternative hypotheses of whether user participation and user orientation were actually an obstacle to the smooth functioning of the software development process, or whether they actually helped this process. In line with Boehm (1981), who describes the primary goals of a software project to be conducting a successful process and achieving a successful product, we looked at quality factors of the process and the product (cf. Budgen 1994, Humphrey 1989). These quality factors were assessed in two measurement periods during the development process. In the second period, part of the projects was already finished.

A better research design would have been to wait until all the products were developed and in use and then to assess their functionality and usability. Such a design would have to cope with nearly insurmountable problems, however. First, the projects studied were developing software products for different application domains (e.g. process control, office administration) and different user groups (e.g. engineers or bookkeepers). There is little chance to develop a common metric to measure functionality or usability across these different areas. Second, the software produced by the projects were often included only as parts of full software packages. Thus, it would be difficult to isolate users' assessments towards these specific parts of the full programs. Third, many of these projects were not finished and in use until several years later. Within the constraints of a project financed by the government, it would have been uneconomical to wait until all the projects were finished. A solution would have been to study only full projects within one domain with similar users in mind. However, this would have resulted in a sample that would have been too small.

Thus, we had to compromise. First, we concentrated on meaningful measures that matched the expertise of our respondents. Since they were mainly software engineers, we felt confident that they could answer engineering criteria related to efficiency and maintainability of the software with good accuracy. Second, to study longer term effects, we had the second wave at a time when the projects had moved further along in their development.

This implies that our study does not answer questions on the influence of user participation on usability. Rather, we concentrate on whether or not the 'smooth functioning' of software development is influenced by user centredness.

3. Methods

3.1. *Sample*

The projects were recruited by using either informal contacts through the German computer science association 'Gesellschaft für Informatik' or by direct mailing to companies with software departments. A total of 29 application software development projects from Germany and German speaking Switzerland participated in the study representing a broad range of different project types. Table 1 shows that the sample included projects of different sizes, that the projects were in different phases of their life cycles and that there was a good distribution of in-house as well as external developments.

In each project, the team leader(s), the users' representative (if applicable), and some of the actual developers were interviewed. Between 30% and 100% (average 70%) of all team members participated. As often in applied research we did not have full control over how many team members could be involved in our research (since the companies had to pay for the subjects' time).

A total of 200 persons participated in this study; 186 took part in interviews for approximately three hours, 180 responded to a questionnaire. Usable data of both types were obtained from 166 persons. Of the participants, 62.1% were systems analysts and programmers, 25.6% team or subteam leaders, 9.9% user representatives, and 2.5% held other functions (e.g. project secretary). The average age was 33 years, the average experience in software development was 5.7 years. Twenty-five per cent of our sample were females.

3.2. *Procedure*

For each project, the team leader was interviewed in detail as to the organizational structure, the tasks involved, the team structure, the procedures to be used, and also on whether or not a user representative existed in the project. Subsequently, the team members were interviewed and questionnaires were distributed. This was the first measurement period.

In a second measurement period, 6–12 months after the first period, the project members were asked to fill in another questionnaire to ascertain the success of the project. Since the projects were further advanced at this stage (see table 1), questions on product quality were now more

Table 1. Features of the 29 field study projects.

Number	User orientation	Project type	Project size	Phase (time period 1)	Phase (time period 2)	Users' computer knowledge
(1) *Strong level of user participation*						
02	low	external	middle	installation	(finished)	high
03	middle	in-house	middle	installation	installation	low
07	low	in-house	large	installation	installation	low
15	middle	external	large	installation	installation	low
18	middle	in-house	small	specification	code and test	low
20	high	in-house	middle	specification	code and test	low
21	middle	external	small	code and test	installation	low
25	high	in-house	middle	specification	installation	low
(2) *Intermediate level of user participation*						
04	low	external	small	code and test	(finished)	high
05	low	external	small	installation	specification	high
06	middle	external	small	installation	(finished)	high
13	middle	in-house	middle	specification	code and test	low
14	low	external	small	specification	code and test	low
16	middle	external	small	code and test	(finished)	low
22	middle	in-house	large	specification	code and test	low
23	high	external	small	installation	installation	low
26	high	in-house	small	installation	installation	low
28	high	in-house	middle	specification	code and test	low
29	middle	external	large	installation	installation	high
(3) *Low level of user participation*						
01	middle	external	small	installation	(finished)	high
08	middle	external	large	code and test	installation	low
09	low	in-house	middle	code and test	(finished)	low
10	middle	external	middle	specification	code and test	low
11	low	external	small	specification	code and test	high
12	middle	external	small	code and test	—	low
17	high	in-house	small	installation	—	low
19	middle	in-house	large	code and test	installation	low
24	high	external	small	specification	—	low
27	middle	in-house	large	code and test	installation	high

Note: user orientation (low = lower 25%; middle = middle 50%; high = upper 25%); project size (small = up to 8 members; middle = 9 to 12 members; large = 13 members or more); a dash indicates missing data.

meaningful. At the second measurement period, 26 (89.7%) of the projects participated. The follow-up sample comprised 135 persons. 112 of those who participated in the first measurement wave filled in the questionnaire again. In addition, 23 people (out of 15 projects) participated for the first time. These were new group members as well as managers who had a profound knowledge of these projects. They gave us an added advantage because they had not yet been involved in the prior research (often they had not been team members at that time) and, therefore were potentially less biased.

3.3. *Measures*

3.3.1. *Measures of user centredness:* There were two measures of user centredness: *user participation and user orientation*.

User participation—an organizational measure—was ascertained on the basis of different sources of information. In the first step we identified whether there were user representatives within each team by asking team leaders. Examples of user participation were: that a customer is a member of the project team, or that a task area specialist is on the project team.

In addition sociometric measures were used to identify whether there were user representatives within the project. Specifically, subjects were asked to name everyone within and outside of their teams they interacted with, to give the function of each person (e.g. user, developer, team leader) and to rate the quality of this interaction. Someone named at least twice to be a user representative and at least twice to be a team member was counted to be a user representative belonging to the team.

Thereafter we distinguished three groups with different levels of user participation. Projects with at least one user representative within the team were regarded to have a high degree of user participation. An intermediate level of user participation implied that the projects had no user representatives within the team but that at least one third of the team members interacted with users. The lowest level of user participation existed if there was only infrequent contact with users by a few project members.

An indication of validity of user participation is that the percentage of team members who had gained their knowledge about the application domain directly from some users (measured by a questionnaire item) significantly correlated with the level of user participation ($r = 0.62$, $N = 29$, $p < 0.05$).

User orientation—a cognitive-emotional attitude measure—consisted of 5 Likert-scale items: (a) 'A good software developer tries first to produce for the user not the machine'; (b) 'For software development it is most important that one can put oneself into the position of the user'; (c) 'At least once a day every software developer should think about how useful the product will be for the user'; (d) 'While working I try to image how the user will deal with the product of my work' and (e) 'A good software developer tries to better the work conditions of the users'. Cronbach's Alpha is 0.63 which is a low but still acceptable reliability for research at its start (cf. Nunnally 1978). User orientation is conceptualized to be an individual cognitive-emotional attitude. This would actually speak against aggregation. However, in order to compare it to user participation and to use aggregated dependent variables, we also aggregated user orientation. The Eta is 0.49 which is at the lower end of acceptability of using aggregated scores. A confirmation of the legitimacy of using an aggregated score is the correlation between the team leaders' user orientation[1] and the average user orientation of the team excluding the team leaders: $r = 0.43$ ($p < 0.05$, $N = 26$ projects). These results indicate that there is at least a certain amount of cohesiveness in a team's user orientation.

3.3.2. *Measures of context variables:* Context features of the project were ascertained on the basis of the interview and questionnaire. The *size of the project* was determined by asking 'how many people are in your team?'. On the basis of the answers given three groups were distinguished: small projects with up to 8 team members, medium-sized projects with 9 to 12 team members and large projects with more than 13 team members. The *phases of the life cycle* of the project were measured in time period 1 and 2 by asking all team members 'which phase is your project currently in?'. Since phases normally were overlapping within each project three main categories were used: specification (comprising analysis, specification and design), code and test (comprising implementation and test), installation (comprising on-site installation and maintenance). Additionally, in time period 1 each members' *team tenure* was ascertained and his or her *participation in each phase of the project* was determined. The extent of *users' knowledge of computers* was rated by the interviewer on the basis of information provided by team leaders.

3.3.3. *Measures of process quality (time period 1): Stressors at work* were ascertained with a 20 item scale (e.g. 'I have too much work' taken from Frese (1988)) with an internal consistency of 0.88. This measure was taken as an indicator of process quality related to the working conditions in the team.

Two single item measures were related to inefficiencies in the ongoing process: *Time inefficiency* (item: 'Time was often wasted') and the frequency of *revision of decisions*.

Quality of team interaction was measured with a 22 item scale constructed on the basis of Watson and Michaelson (1988). Items referred to democracy, openness to criticism,

Table 2. Intercorrelations between user participation, user orientation, outcome measures, and context variables.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1 user participation	(–)																		
2 user orientation	– 0.03	(0.24)																	
Time period 1																			
3 stressors	0.11	0.55**	(0.31)																
4 time inefficiency	– 0.22	0.36##	0.32##	(0.30)															
5 revision of decisions	– 0.28	0.34##	0.19	0.69**	(0.39)														
6 team effectiveness	– 0.07	– 0.49*	– 0.19	– 0.55**	– 0.45*	(0.46)													
7 quality of team interaction	– 0.05	– 0.47*	– 0.52**	– 0.45*	– 0.29	0.46*	(0.38)												
8 changeability	– 0.14	– 0.19	– 0.25	– 0.25	– 0.33##	0.45*	0.60**	(0.44)											
9 modularity	– 0.20	– 0.33##	– 0.42*	– 0.38*	– 0.33##	0.29	0.55**	0.52**	(0.30)										
10 quality of documentation	0.25	– 0.29	– 0.30	– 0.48*	– 0.56**	0.32	0.49**	0.56**	0.38*	(0.63)									
Time period 2																			
11 overall success	– 0.47*	– 0.44*	– 0.36##	– 0.08	0.11	0.54*	0.07	0.16	0.24	– 0.13	(0.50)								
12 on timelin budget	– 0.31	– 0.33	– 0.36##	– 0.49*	0.02	0.50*	0.26	0.24	0.29	0.11	0.59**	(0.70)							
13 innovations	– 0.40*	– 0.27	– 0.20	0.13	0.17	0.14	0.14	– 0.34##	0.36##	0.16	0.59**	0.35##	(0.37)						
14 flexibility	– 0.44*	– 0.26	– 0.29	– 0.25	– 0.05	0.34	0.04	0.25	0.36##	– 0.03	0.73**	0.52**	0.46*	(0.45)					
15 team effectiveness	– 0.45*	– 0.54*	– 0.53**	– 0.53**	– 0.33##	0.65**	0.42*	0.62**	0.56**	0.43*	0.64**	0.67**	0.40*	0.70**	(0.52)				
16 changeability	– 0.34##	– 0.11	– 0.10	– 0.30	– 0.18	0.48*	0.25	0.66**	0.30	0.50*	0.28	0.29	0.32	0.48*	0.66**	(0.48)			
Context variables																			
17 size of project	0.11	– 0.09	– 0.12	0.07	– 0.06	– 0.21	– 0.52**	– 0.18	– 0.13	– 0.07	0.15	– 0.00	0.25	0.18	– 0.03	– 0.19	(–)		
18 phase	0.11	– 0.10	0.08	– 0.05	– 0.17	0.05	– 0.29	– 0.48**	– 0.08	– 0.35##	0.21	0.09	– 0.35##	– 0.09	– 0.09	– 0.42*	0.01	(–)	
19 team tenure (in months)	0.09	0.28	0.43*	0.16	0.04	– 0.11	– 0.09	– 0.25	– 0.05	– 0.45*	– 0.10	– 0.10	– 0.40*	– 0.20	– 0.37##	– 0.54**	– 0.20	0.55**	(0.28)
M	—	3.87	2.34	2.49	2.84	6.72	3.74	6.33	4.23	3.56	3.62	3.31	3.34	3.51	6.92	6.46	1.66	2.03	18.58
SD	—	0.35	0.36	0.58	0.71	1.42	0.39	1.22	0.38	0.72	0.64	0.72	0.62	0.62	1.47	1.21	0.97	0.87	10.85
N	29	29	29	29	29	24	29	29	28	29	25	26	26	26	26	26	29	29	29
Range	1–3	1–5	1–5	1–5	1–5	1–10	1–5	1–7	1–5	1–5	1–5	1–5	1–5	1–5	1–10	1–7	1–3	1–3	—

Note: Values in parentheses are Eta^2. $*p < 0.05$ $**p < 0.01$ $\#\#p < 0.10$.

competition (reversed) and dominance (reversed). Cronbach's Alpha for the scale was 0.92.

Team effectiveness was based on a ten-point interview item.

3.3.4. *Measures of product quality (time period 1):* The product quality measures all related to software engineering criteria. Since the subjects were software engineers, we assumed that we could trust their judgement on these dimensions but not on issue of functionality or usability. These measures were developed with computer scientists with whom we collaborated in this project (Bittner *et al.* 1995).

Quality of documentation was measured with five Likert items (e.g. 'The documentation is very detailed') with a Cronbach's Alpha of 0.87. All items referred to the documentation of development, not to any type of documentation dedicated to the user.

Modularity of the target software was estimated with four items, such as 'For every application procedure one can tell by which component it is realized'. Cronbach's Alpha was 0.71.

Changeability of the target software was ascertained with a ten-point item present in the interview ('Estimate the changeability of the whole product developed in your project').

3.3.5. *Measures of project success (time period 2):* At time period 2 measures of project success were ascertained. The *overall success* of the project, *innovations* made during development, *flexibility of the project* (i.e. reaction of the project to unpredicted events) were measured by single Likert items. The measure of *on time/in budget* was ascertained by two items referring to fulfilment of time and budget requirements ($r = 0.58$; $N = 103$; $p < 0.01$). *Team effectiveness* and *changeability* of the target software that had been already measured in time period 1 were measured at time 2 again.

3.4. *Aggregation*

In the analyses aggregated project scores were used. To test the agreement within the projects groups, *Eta*s were computed and are displayed in table 2 (cf. James 1982). Using aggregated scores has several advantages. First, there is little influence of response bias because idiosyncratic responses are eliminated. Second, not everyone in a team is really able to respond to all aspects of the development process and quality, but collectively team members have a good knowledge of all these aspects. Third, the appropriate unit of analysis in research on team performance is the team. Thus, the aggregated scores reflect the average level of every variable within a project.

4. Results and discussion

4.1. *Descriptive results*

User participation took different forms. For example, there was an in-house project developing software for the company's administration. The overall tasks were jointly defined by the software team and the department of the future users. The latter was supposed to contribute one third of the total effort of about 100 person years. In another case, members of the software team moved to the customer, and built a team together with some user representatives. There were also projects without this strong form of participation. Some teams included a user representative in some consulting position. Some had an interdisciplinary advisory board installed. Others had a user representative located in a position between the development team and the application domain.

User representatives usually specialized in certain tasks, for example, specifying or testing, while involvement in implementation was negligible (Brodbeck *et al.* 1993). Often they were involved in the project from the very beginning. More than fifty per cent of the user representatives had a longer team tenure than the average team member and nearly seventy per cent had already participated in the early phases of the project's life cycle.

User orientation was high throughout. Only two per cent of the software developers considered user orientation to be of low or no importance. Other results also suggest that there was a positive attitude toward the users, e.g. user friendliness of the software was estimated to be one of the most important criteria. However, there was still an appreciable variation in user orientation between the projects with a range from $M = 2.90$ to $M = 4.60$ ($SD = 0.35$) on a five-point scale.

4.2. *Problems associated with user participation and user orientation*

Table 2 presents the full intercorrelation matrix. Projects with high *user participation* showed lower overall success, fewer innovations, a lower degree of flexibility, and lower team effectiveness—all measured at time period 2. None of the correlations of time period 1 were significant. This indicates that negative features associated with user participation do not become apparent immediately but only later in the process.

User orientation correlated positively with stressors and negatively with team effectiveness and quality of team interaction at time period 1. These three process variables mainly refer to the work situation in the projects. But also other, non-significant correlation coefficients were in the same direction. With respect to time period 2 measures, user orientation was negatively correlated to overall success and team effectiveness.

Thus, the data suggest that projects high on user orientation and user participation were not running smoothly.

Potential mediator effects. Further inspection of the intercorrelation matrix shows that team effectiveness—measured at time period 1—is related to outcome measures at time period 2. This suggests that team effectiveness at time 1 might function as a mediator in the relationship between user orientation and outcome measures at time period 2. Although we did not hypothesize this mediator effect it would make sense that user orientation leads to a lower degree of team effectiveness which in turn leads to lower quality and success. Additionally, stressors and quality of team interaction might be mediators—especially in the relationship between user orientation and team effectiveness at time period 2. Here the argument would be similar to the above mediator effect. User orientation could effect team interaction and stressors which affect later success measures.

In order to test this, a set of hierarchical multiple regression analyses were performed (Cohen and Cohen 1975). In the first set of analyses only user orientation was used as a predictor for overall success and team effectiveness (time period 2) respectively. In the second set of analyses stressors, team effectiveness (time period 1), and quality of team interaction were entered in the first, user orientation in the second step. The percentage of variance of overall success explained by user orientation was reduced from 21 per cent to 2 per cent when stressors, team effectiveness (time period 1), and quality of team interaction were entered first into the equation, indicating a strong mediator effect. Similarly, the percentage of variance of team effectiveness (time period 2) explained by user orientation was reduced from 16 per cent to 0 per cent after stressors, team effectiveness (time period 1), and quality of team interaction were entered first into the equation. These results show that the relationship between user orientation and outcome measures at time period 2 is mediated by problems in the ongoing process such as stressors, low team effectiveness and low quality of team interaction. No such strong mediator effects were found for the relationship between user participation and outcome measures of time period 2.

Neither user participation nor user orientation was significantly correlated to changeability, modularity, quality of documentation at time period 1 suggesting that short-term product quality is to a weaker extent related to user centredness than is short-term process quality.

With respect to measures of time period 2, all correlation coefficients show a negative sign, although not all reach the significance criterion. Furthermore, it must be noted that some of the time period 2 measures are highly correlated to their corresponding time period 1 measures (e.g. time inefficiency—on time/on budget; changeability) indicating that decisions and events occurring early in the process can hardly be compensated for later.

4.3. *Relationship between user orientation and user participation*

The two variables of user centredness shared no common variance ($r = -0.03$, n.s.). This indicates that the organizational practice of user participation is not necessarily reflected in individuals' attitude to user orientation. Furthermore, there was no mediation. When user participation was controlled for, there was little change in the correlations between user orientation and outcome variables. The same is true if user orientation was factored out of the correlations between user participation and the other variables. Thus, it is not the higher user orientation which made user participation difficult.

4.4. *User centredness and task difficulty*

One might assume that user participation is mainly done in large and complex projects that might be more difficult to coordinate and therefore might be associated with low success (cf. Kraut and Streeter 1995). However, there were no significant correlations of user orientation and user participation with context factors like project size, phase of the life cycle, and team tenure (cf. table 2). To control for a potential bias due to life cycle characteristics, we computed the correlations between user participation and outcome variables, excluding projects which were classified as maintenance projects. However, the correlation pattern remained almost the same.

5. Overall discussion

The results suggest a clear pattern: user participation and user orientation were negatively related to features of process and product quality. User participation was related to low overall success, few innovations, little flexibility, low team effectiveness, and low changeability of the software. User orientation was related to high stressors, low quality of team interaction, low overall success, and low team effectiveness.

User orientation was mainly correlated with measures of the ongoing process, as measured at time period 1. In contrast, user participation showed significant correlations with measures of time period 2 when projects had advanced much further. We consider the differences between user orientation and user participation to be related to this time difference. User orientation seems to be more intimately related to ongoing events, that in turn are related to the project outcome. User participation is more directly related to overall success measures. Moreover, user orientation captured the more subjective and cognitive side of user centredness, whereas user participation was a more objective organizational measure.

Our results on the negative effect of user centredness contradict those positions in the literature suggesting a positive effect of user orientation and user participation on the software development process (Baroudi *et al.* 1986, Bewley *et al*. 1983, Boehm *et al.* 1984, Gomaa 1983, Karat 1990, Mantei and Teorey 1988, Strohm 1991). Four interpretations for these results will be discussed: (1) it is all in the developers' mind; (2) projects with problems turn to users for help; (3) misattribution—projects with problems develop higher user orientation; (4) user participation/user orientation lead to more objective problems.

(1) *Interpretation 1: It is all in the developers' mind*

Project success variables were assessed by the software developers themselves. One could argue that we only measured what was in the software designers' minds and that these data bear no relationship to reality. There is the danger that we only measured the developers' subjective theory of the impact of users on the development process. This subjective theory may be of an individual nature or it may be a stereotype common to all software developers. There is no doubt that there is some truth to this explanation. Whenever one does a study based on interviewees' responses, there is the danger of getting a subjective representation that distorts reality. However, this explanation certainly does not explain all of the results. Since we analysed aggregated group measures, it cannot be the result of individual distortions. A collective distortion being operative in all software developers implies that these common stereotypes are also highly consistent across time. Such a consistency was not apparent in our data; the negative effects of user participation appeared only at time period 2.

Moreover, there was a clear correlation between subjective assessments and 'objective' project characteristics. Those projects that were thought to be inefficient were, in fact, less often finished by time 2 than were projects that were thought to be efficient (time inefficiency at time period 1 correlated with completion at time period 2 with $r = -0.40$, $N = 26$, $p < 0.05$).

(2) *Interpretation 2: Projects with problems turn to users for help*

One might think that in cases of extreme difficulties software projects would turn to users for help. We think it is highly unlikely that users can actually help in such a situation. This interpretation implies that users would have been involved only late in the development process. However, this is not the case, because user representatives were not different in team tenure from other participants and normally participated in the early phases as well.

(3) *Interpretation 3: Misattribution—projects with problems develop higher user orientation*

Lower quality projects may attempt to be more user oriented to be able to attribute their failure externally. In a way people answer the question of why they have not been more successful by saying 'but we had to be so much more concerned with the users'. This may eventually lead to a higher degree of user orientation in those projects. We do not believe that this is really a plausible explanation. In any case, this reasoning can only be applied to the more subjective concept of user orientation and not to the more objective user participation.

(4) *Interpretation 4: User participation or user orientation lead to more objective problems*

As an alternative to the interpretations considered above, this interpretation posits that user participation and user orientation lead to various objective problems. Qualitative data from the interviews convinced us that this interpretation is viable. The participation projects had to deal with several problems related to developer–user relations that were not present in projects without user participation.

First, the users developed more sophisticated ideas in the course of the development process and, therefore, they intervened more frequently later on. The software developers were not able to anticipate these ideas, which are often very difficult to incorporate into the software in a late stage of the development. Second, users often feared job loss or worsened working conditions as a result of the new software and, therefore, were not interested in participating constructively. This might have lead to misinformation. Third, user representatives were often unpredictable. In one project, late in the development process, users demanded a direct manipulation interface which they had just seen with a new standard software. In another case, a user representative demanded changes at the moment that developers wanted to start testing the software. This type of intervention might disrupt the software development process. Fourth, user orientation may lead to higher levels of aspiration and, therefore, to a higher degree of stressors and a lower degree of team effectiveness. This might especially be the case when only little knowledge and experience are available on *how* to put user orientation into practice.

Thus, we think that there are objective problems with user orientation and user participation. There are more interests to be balanced, and more relationships to be managed. Therefore, the smooth functioning of a project is more likely to be impaired—a result that is of practical importance for software designers and their managers.

(5) *Strengths and limitations*

The strengths of our study are: first, we used aggregated scores ruling out individual biases. Second, data collection was done at two points in time, allowing the assessment of long term relationships. Third, the sample of projects was heterogeneous and went beyond the typical case study approach, allowing better generalization to the population. Finally, two different aspects of user centredness were assessed. The cognitive-emotional user orientation was found to relate immediately to negative quality factors, while the problems associated with organizational user participation became obvious only in the longer run.

There are two limitations of our study. First, one could argue that our measure of user participation is rather coarse and may not be interval scaled. We think that the tri-partite measure is really 'cutting nature at its joints' because software development projects naturally fall into three categories. However, using a dichotomous variable of user representative in the project versus not, leads essentially to the same results (since point-biserial correlations were used for this dichotomous variable, the scaling argument can be refuted). Thus, we think that these problems should not lead one to discount our results.

The second limitation is that the success measures stem from the software designers themselves. We have not studied users' responses to the program developed by the projects. Thus, our results are mute with respect to the question, whether or not user centredness is related to better usability or functionality. Thus, the results do not contradict those positions who propose user participation as a useful methodology to implement computer systems in working life.

However, our results imply that naive statements suggesting that user centredness is all positive, need to be modified. We do not want the reader to go away with the conclusion that one should not design software with the help of user representatives at all, or that user orientation is completely negative. As a matter of fact, one could very well argue that the problems of user participation apparent in our research are by-products of user centredness. User participation may be more uncomfortable and difficult for software developers, but still be the better choice for the user (Mumford and Weir 1979). Thus, our results may point out that there is a trade-off between engineering and usability criteria.

In the future, it is necessary to develop a much clearer picture of what is happening in software development (Brodbeck and Frese 1994). Since problems appear in projects with user participation, the research agenda should be to get a clearer conceptualization of precisely which kinds of problems are most prominent and how one can cope with these problems more effectively.

Acknowledgements

This paper is based on project IPAS (a German acronym of 'Interdisciplinary Project about the Work Situation in Software Development'). The project is supported by a grant from the Work and Technology Fund of the Ministry of Research and Technology of the Federal Republic of Germany (No. 01 HK 319). Project IPAS consists of computer scientists (University of Marburg: Udo Bittner, Wolfgang Hesse (principal investigator), Johannes Schnath), sociologists (SPG Sozialwissenschaftliche Projektgruppe, Munich: Rolf Ortmann, Friedrich Weltz (principal investigator)) and work psychologists (University of Giessen: Felix C. Brodbeck (now at University of Munich), Michael Frese (principal investigator, now at University of Amsterdam), Angelika Fritz, Torsten Heinbokel (now at University of Bundeswehr Hamburg), Sabine Sonnentag (now at University of Amsterdam), Wolfgang Stolte). We wish to thank anonymous reviewers and Chris Clegg for their very helpful comments on earlier versions of this paper.

Note

[1]If there was more than one team or subteam leader their average score was taken.

References

Aschersleben, G. and Zang-Scheucher, B. 1989, Der Prozeß der Software-Gestaltung—eine Bestandsaufnahme in Wissenschaft und Industrie (The process of software design—a stock-taking in science and industry), in S. Maaß and H. Oberquelle (eds), *Software-Ergonomie '89* (Teubner, Stuttgart), 244–253.

Baroudi, J., Olson, M., and Ives, B. 1986, An empirical study of the impact of user involvement on system usage and information satisfaction, *Communications of the ACM*, **29,** 232–238.

Bewley, W. L., Roberts, T. L., Schroit, D., and Verplanck, W. L. 1983, Human factors testing in the design of XEROX's 8010 'Star' office workstation, *Proceedings of CHI '83 conference on Human factors in computing systems* (ACM, New York), 72–77.

Bittner, U., Hesse, W., and Schnath, J. 1995, *Praxis der Software-Entwicklung* (*Practice of Software Development*) (Oldenbourg, München).

Bjerknes, G., Ehn, P., and Kyng, M. (eds) 1987, *Computers and Democracy: A Scandinavian Challenge* (Avebury, Aldershot).

Boehm, B. W. 1981, *Software Engineering Economics* (Prentice-Hall, Englewood Cliffs).

Boehm, B. W. 1987, Improving software productivity, *IEEE Computer*, **20,** 43–57.

Boehm, B. W., Gray, T., and Seewaldt, T. 1984, Prototyping versus specifying: a multiproject experiment, *IEEE Transactions on Software Engineering*, **10,** 224–236.

Briefs, U., Ciborra, C., and Schneider, L. (eds) 1983, Systems design for, with and by the users, *Proceedings of the IFIP WG 9.1 Working Conference* (North-Holland, Amsterdam).

Brodbeck, F. C. and Frese, M. (eds) 1994, *Produktivität und Qualität in Software-Projekten* (*Productivity and Quality in Software Projects*) (Oldenbourg, München).

Brodbeck, F. C., Sonnentag, S., Heinbokel, T., Stolte, W., and Frese, M. 1993, Tätigkeitsschwerpunkte und Qualifikationsanforderungen in der Softwareentwicklung. Eine empirische Untersuchung (Main tasks and qualification requirements in software development. An empirical investigation), *Softwaretechnik-Trends*, **13,** 31–40.

Brown, C. M. 1989, *Human–Computer Interface Design Guidelines* (Ablex, Norwood, New Jersey).

Budde, R., Kautz, K., Kuhlenkamp, K., and Züllighoven, H. 1992, *Prototyping. An Approach to Evolutionary Ssytem Development* (Springer, Berlin).

Budgen, D. 1994, *Software Design* (Addison-Wesley, Wokingham).

Carmel, E., Whitaker, R. D., and George, J. F. 1993, PD and joint application design: a transatlantic comparison, *Communications of the ACM*, **36,** 40–48.

Clement, A. and Van den Besselaar, P. 1993, A retrospective look at PD projects, *Communications of the ACM*, **36,** 29–37.

Cohen, J. and Cohen, P. 1975, *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences* (Erlbaum, Hillsdale, NJ).

Curtis, B., Krasner, H., and Iscoe, N. 1988, A field study of the software design process for large systems, *Communications of the ACM*, **31,** 1268–1287.

Floyd, C. 1984, A systematic look at prototyping, in R. Budde, K. Kuhlenkamp, L. Mathiassen and H. Züllighoven (eds), *Approaches to Prototyping* (Springer, Berlin), 1–18.

Frese, M. 1988, *Die Führung der eigenen Person. Streßmanagement* (*Leading One's Own Person. Stress Management*) (Mensch und Arbeit, München).

Gomaa, H. 1983, The impact of rapid prototyping on specifying user requirements, *ACM Sigsoft Software Engineering Notes*, **8,** 17–28.

Gould, J. D. and Lewis, C. 1985, Designing for usability: Key principles and what designers think, *Communications of the ACM*, **28,** 300–311.

Gould, J. D. and Lewis, C. 1983, Designing for usability—Key principles and what designers think, *Proceedings of CHI'83 Conference on Human Factors in Computing Systems* (ACM, New York), 50–53.

Grudin, J. 1991, Systematic sources of suboptimal interface design in large product development organizations, *Human–Computer Interaction*, **6,** 147–196.

Humphrey, W. S. 1989, *Managing the Software Process* (Addison-Wesley, Reading, Mass.).

Ives, B. and Olson, M. H. 1984, User involvement and MIS success: a review of research, *Management Science*, **30,** 586–603.

James, L. R. 1982, Aggregation bias in estimates of perceptual agreement, *Journal of Applied Psychology*, **67,** 219–229.

Jörgensen, A. H. 1984, On the psychology of prototyping, in R. Budde, K. Kuhlenkamp, L. Mathiassen and H. Züllighoven (eds), *Approaches to Prototyping* (Springer, Berlin), 278–289.

Karat, C. M. 1990, Cost-benefit-analysis of iterative usability testing, in D. Diaper, D. Gilmore, G. Cockton and B. Shackel (eds), *Human–Computer Interaction—INTERACT '90* (Elsevier, Amsterdam), 351–356.

Kraut, R. E. and Streeter, L. A. 1995, Coordination in software development, *Communications of the ACM*, **38,** 69–81.

Kensing, F. and Munk-Madsen, A. 1993, PD: Structure in the Toolbox, *Communications of the ACM*, **36,** 78–85.

Mantei, M. M. and Teorey, T. J. 1988, Cost/benefit analysis for incorporating human factors in the software lifecycle, *Communications of the ACM*, **31,** 428–439.

Mumford, E. 1981, Participative systems design: Structure and method, *Systems, Objectives, Solutions*, **1,** 5–19.

Mumford, E. and Weir, M. 1979, *Computer systems in work design—the ETHICS method* (Associated Business Press, London).

Muller, M. J., Wildman, D. M., and White, E. A. 1993, Taxonomy of PD practices: a brief practitioner's guide, *Communications of the ACM*, **36,** 26–27.

Norman, D. A. and Draper, S. W. (eds) 1986, *User Centred System Design: new Perspectives on Human–Computer Interaction* (Lawrence Erlbaum Associates, New Jersey: Hillsdale).

Nunnally, J. 1978, *Psychometric Theory* (McGraw-Hill, New York).

Peschke, H. 1986, *Betroffenenorientierte Systementwicklung. Prozess und Methoden der Entwicklung menschengerechter Informationssysteme* (*System Development Oriented Towards the Concerned Persons. Process and Methods for the Development of Information Systems Adequate to Humans*) (Peter Lang, Frankfurt).

Rosson, M. B., Maass, S., and Kellogg, W. A. 1987, Designing for designers: an analysis of design practice in the real world, *Proceedings of CHI '87 conference on Human factors in computing systems* (ACM, New York), 137–142.

Selig, J. 1986, *EDV-Managment. Eine empirische Untersuchung der Entwicklung von Anwendungssystemen in deutschen Unternehmen* (*EDP Management. An Empirical Research on the Development of Application Systems in German Companies* (Springer, Berlin).

Shackel, B. 1985, Human factors and usability—whence and whither, in H. J. Bullinger (ed.), *Software-Ergonomie '85* (Teubner, Stuttgart), 13–31.

Shneiderman, B. 1992, *Designing the User Interface: Strategies for Effective Human–Computer Interaction.* (Addison-Wesley, Reading, Mass.).

Smith, S. L. and Mosier, J. N. 1986, *Guidelines for Designing User Interface Software* (MITRE, Bedford, Mass.).

Spinas, P. 1990, Benutzerfreundlichkeit von Dialogsytsemen und Benutzerbeteiligung bei der Software-Entwicklung (User friendliness of dialogue systems and user involvement in software development), in F. Frei and I. Udris (eds), *Das Bild der Arbeit* (Huber, Bern), 158–171.

Strohm, O. 1991, Projektmanagement bei der Software-Entwicklung: Eine arbeitspsychologische Analyse und Bestandsaufnahme (Project management in software development. A work psychological analysis and stock-taking), in D. Ackermann and E. Ulich (eds), *Software-Ergonomie '91* (Teubner, Stuttgart), 46–58.

Ulich, E. 1993, Von der Benutzungsoberfläche zur Arbeitsgestaltung

(From the user interface towards work design), in K.-H. Rödiger (ed.), *Software-Ergonomie '93*, (Teubner, Stuttgart), 19–29.

WATSON, W. E. and MICHAELSON, L. K. 1988, Group interaction behaviors that affect group performance on an intellective task, *Group and Organisation Studies*, **13,** 495–516.

WELTZ, F. and ORTMANN, R. 1992, *Das Software-Projekt* (*The Software Project*) (Campus, Frankfurt/Main).

WILLIGES, R. C., WILLIGES, B. H., and ELKERTON, J. 1987, Software interface design, in G. Salvendy (ed.), *Handbook of Human factors* (Wiley, New York), 1416–1449.