

Use of Design Methods, Team Leaders' Goal Orientation, and Team Effectiveness: A Follow-Up Study in Software Development Projects

**Sabine Sonnentag
Michael Frese
Felix C. Brodbeck
Torsten Heinbokel
University of Giessen**

This article reports an empirical study on effectiveness of software development teams. It was predicted that both the extent to which design methods are used and team leaders' goal orientation is related to team effectiveness. Follow-up data (6-12 months after first assessment) were available from 25 German and Swiss software development projects. Hierarchical regression analysis showed that use of design methods and team leaders' goal orientation predicted team efficiency and changeability of the software product. In addition, goal orientation predicted maintaining schedule and budget. Practical implications of this study refer to the necessity to consider both the use of design methods and leadership issues when aiming at effective software development projects.

1. INTRODUCTION

The quality of a software system and the degree to which users get along with it are to a great extent determined by what is happening in the software development process (Grudin, 1991). Once the software has been developed, the way users have

Sabine Sonnentag and Michael Frese are now at the University of Amsterdam. Felix C. Brodbeck is now at the University of Munich. Torsten Heinbokel is now at the Bundeswehrhochschule Hamburg.

This article is based on the project IPAS (a German acronym for 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). The authors take responsibility for this publication. IPAS consists of a computer science part (University of Marburg: U. Bittner, W. Hesse [principal investigator], J. Schnath), a social science part (SPG Sozialwissenschaftliche Projektgruppe, Munich: F. Weltz, R. Ortmann), and a work psychology part (University of Munich and University of Giessen: F. C. Brodbeck, M. Frese [principal investigator], A. Fritz, T. Heinbokel, S. Sonnentag, W. Stolte).

Requests for reprints should be sent to Sabine Sonnentag, Department of Psychology, University of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands. E-mail: ao_sonnentag@macmail.psy.uva.nl.

to do their tasks is relatively fixed. Later changes are expensive and difficult (Boehm, 1981). Thus, it is of great practical importance to examine which factors have an impact on the effectiveness of software development teams.

1.1. The Importance of Design Methods

From a technical point of view, the question of effectiveness in software development is examined and prescribed by the discipline of software engineering, which in turn emphasizes the development and application of software design methods (Budgen, 1994; Coad & Yourdon, 1991; Pressman, 1991; Wasserman, 1980; Yourdon & Constantin, 1979). Such methods can be defined as general guidelines or work rules for software designers, suggesting what to do, how to proceed, and how to represent both the initial problem and the solution (Budgen, 1994).

Budgen (1994) summarized the technical advantages of design methods as follows: First, the use of a design method may help an inexperienced designer to compensate for lack of domain knowledge by guiding the formulation and exploration of essential design features. Second, design methods guarantee a certain degree of consistency in the final product by providing a common standard for a design team. Third, design methods make maintenance easier by requiring the production of standardized representations and records during the development process. Finally, the use of methods may reduce logical errors in the design by helping to consider all relevant aspects of the initial problem. Thus, Brodbeck (1993) conceptualized the lack of methods and rules as a "handicap" in the software development process.

Experimental research on software design methods showed that the use of specific design methods has both an effect on the design process and the product (Boehm-Davis & Ross, 1992; Lee & Pennington, 1994). Empirical work that studied the effect of design methods in professional software development projects suggests that the use of adequate methods and tools is related to project success (Saarinen, 1990). In addition, Kraut and Streeter (1995) found formal, impersonal procedures (i.e., documents and memos; modification request projects; and error tracking procedures) to be positively related to managers' evaluations of the success of software development projects. Thus, it can be hypothesized that the use of software design methods is positively related to effectiveness in software development teams.

1.2. The Role of Team Leaders' Goal Orientation

Are formalized procedures such as software design methods alone sufficient for high effectiveness in software development? Empirical work on performance in research and development settings in general suggests that this is not the case. A variety of studies show that leadership issues play an important role in predicting research and development effectiveness, both at the group and at the individual

level (Hill, 1970; Knorr, Mittermeir, Aichholzer, & Waller, 1979; Scott & Bruce, 1994; Thamhain & Wilemon, 1987).

Leadership issues can be approached within a wide range of conceptualizations. In this article, we concentrate on team leaders' action styles, namely, goal orientation. Frese, Stewart, and Hannover (1987) conceptualized goal orientation as an action style, that is, as a propensity to act. Thus, goal orientation can be regarded as a personality concept implying the existence of individual differences in the extent people set goals and pursue them. Highly goal-oriented persons develop long-range and clear goals. They are persistent in pursuing them, especially when difficulties occur. Therefore, goal orientation is assumed to be an important prerequisite of effective leadership.

Previous research has shown that a person's goal orientation was related to his or her performance in individual settings (Frese et al., 1987; Frese et al., 1995). In a recent cross-sectional analysis of the study reported in this article, team leaders' goal orientation was found to be related to team performance (Sonnetag, Frese, Stolte, Heinbokel, & Brodbeck, 1994). However, within this analysis, it remained unanswered whether team leaders' goal orientation has a positive effect on team effectiveness in addition to other factors, such as software design methods.

Here, one can think of two different lines of arguments. First, Kerr and Jermier's (1978) "substitute for leadership" model suggests that certain subordinate, task, and organizational variables impact on the relation between leader behavior and outcome measures, such as performance or satisfaction. Variables conceptualized as substitutes for task-oriented and goal-emphasizing leadership are, among others: high abilities and "professionalism" of subordinates; unambiguous and routine tasks; methodologically invariant tasks, including standardized work procedures; and formalization of organizational procedures by explicit goal, plans, and guidelines. One might argue that software design methods contribute to the standardization of work procedures, provide goals and guidelines, and act therefore as substitutes for leadership. Thus, one could assume that in face of the use of design methods, team leaders' goal orientation is of minor importance for team effectiveness.

However, we have two objections. First, empirical support for this model is rather weak in several different samples, including professionals (Podsakoff, MacKenzie, & Fetter, 1993; Podsakoff, Niehoff, MacKenzie, & Williams, 1993). Second, even if design methods are applied, much of the work in software development remains ill-defined, ambiguous, and nonroutine in nature (Brodbeck, 1993; Frese & Hesse, 1993; Glass, Vessey, & Conger, 1992) necessitating leadership. Thus, the substitutes for leadership model cannot argue convincingly that leaders' goal orientation is unimportant.

The second line of arguments starts from the assumption that the use of design methods and team leaders' goal orientation have a differential impact on the development process. The application of design methods aims at the improvement of the *technical features* of the development process by providing guidelines with regard to design procedures and notations. Thus, using design methods should have a positive effect on the software product. In addition, it should increase the

efficiency of the development process by providing common procedures that reduce team members' uncertainty and assure a high degree of consistency.

In contrast, team leaders' goal orientation affects *organizational features*. High goal orientation helps when one has to clarify goals, set priorities, and pursue goals in case of difficulties. Goal orientation should therefore have a positive effect on the development process and the final product. This is important because there are not only technical but also organizational problems in the software development projects. One might think of conflicts with the customer or among team members, difficulties in the use of scarce resources, or time pressure. These problems cannot be overcome with purely technical measures, such as the use of design methods. Thus, it can be hypothesized that both team leaders' goal orientation and the use of design methods are important for effectiveness in software development projects.

1.3. Summary of Hypotheses

The hypotheses can be summarized as follows:

1. The extent to which design methods are used in the software development process is positively related to team effectiveness.
2. Team leaders' goal orientation is positively related to team effectiveness, in addition to the use of design methods.

2. METHOD

2.1. Sample

An empirical study was performed in 29 software development projects from 19 German and Swiss companies. The projects produced software for a variety of application domains, such as administration software for small and big companies; software for telephone and communication purposes; software for banks, insurance companies, and traffic institutions; and process control software. At Time 1, 34% of the projects were performing requirement analysis and software design, 28% were in the phase of coding and testing, and 38% accomplished mainly installation and maintenance tasks. The mean project size was 10 members ($SD = 4.8$).

A total of 200 software professionals participated in the first data collection (Time 1). Of these, 62.1% were systems analysts or programmers; 25.6% were team leaders or subteam leaders; 9.9% were user representatives; and 2.5% had other, mainly administrative tasks in the projects. On average, participants had spent 5.7 years in professional software development. The mean age was 33 years. Seventy-five percent of the participants were men.

Six to 12 months after the first data collection, a follow-up was performed in which 26 (89.7%) of the 29 projects participated. This follow-up sample comprised 135 persons, partially overlapping (88.1%) with the sample of Time 1. Of this Time 2 sample, 58.1% were systems analysts or programmers, 25% were team or subteam leaders, 5.1% were user representatives, 10.3% were managers from outside the

projects studied, and 2% had other functions. Managers were included in the sample in order to assess team effectiveness from a point of view additional to that of the directly involved team members. Due to missing data in one of the central variables (team leaders' goal orientation), data analysis was based on a total of 25 projects.

2.2. Measures

Means, standard deviations, alpha reliabilities, agreement among project members (ϵ^2), and intercorrelations are shown in Table 1. Use of design methods, team leaders' goal orientation, and context conditions were assessed at Time 1, and team effectiveness measures were collected at Time 2. All variables except context conditions were ascertained by questionnaires. Context conditions were assessed in an interview. All scales were in German.

Use of Design Methods. For assessing the extent to which design methods were used in the software development process, a 5-item scale was developed. A sample item is "Criteria concerning the product quality: *do not exist* (1) ... *are specified in instructions of procedure* (5)." Cronbach's alpha for this scale was .73 and ϵ^2 was .61, indicating a high agreement within the projects. For every project a use-of-design-methods score was computed by averaging individual responses.

Goal Orientation of Team Leaders. Goal orientation of team leaders was measured with a 6-item scale developed by Frese et al. (1987). Items were formulated in a stimulus response format (Endler & Hunt, 1966), for example: "I take all my goals very seriously" versus "My goals might be important, but sometimes I lose sight of them." Team and subteam leaders were asked to relate their answers to specific actual work tasks, mainly team leading (for details, see Sonnentag et al.,

Table 1: Means, Standard Deviations, Agreement Among Project Members, Alpha Reliabilities, and Intercorrelations

Variables	M	SD	ϵ^2	1	2	3	4	5	6	7
1 Favorable context ^b	.48	.51	—	(—)						
2 Use of design methods ^c	2.87	.43	.61	.05	(.73)					
3 Team leaders' goal orientation ^c	3.70	.62	—	-.01	.22	(.70)				
4 Changeability of product ^d	6.37	1.15	.48	.25	.62**	.43*	(—)			
5 Team effectiveness ^e	6.91	1.50	.52	.61**	.49**	.47**	.69**	(—)		
6 Maintaining schedule and budget ^c	3.36	.67	.70	.41*	-.02	.39*	.41*	.69**	(.58)	
7 Success of project ^c	3.42	.48	.61	.42*	.18	.24	.41*	.73**	.66**	(.63)

Note. $N = 25$ projects.

^aCronbach's alpha based on individual data and shown in parentheses. No alpha reliabilities can be provided for favorable context-index and single item-measures. ^bDichotomous variables (0 = unfavorable, 1 = favorable). ^cRange: 1-5. ^dRange: 1-9. ^eRange: 0-10.

* $p < .05$. ** $p < .01$.

1994). In projects with more than one team or subteam leader, the average goal-orientation score of all of these leaders was used in analysis. Cronbach's alpha in this team leader sample was .70.

Team Effectiveness. Team effectiveness was measured at Time 2 with four variables: changeability of the software product, team efficiency, maintaining schedule and budget, and success of project. Changeability of the software product is an important indicator of modern software systems (Budgen, 1994). It was measured with a single-item indicator. For assessing team efficiency, participants were asked to indicate their perceived effectiveness of the project team on an 11-point scale. Keeping time and costs was measured with two items, one relating to scheduled costs and one relating to scheduled time. Intercorrelation of these two items was .58 ($p < .01$). Success of project was assessed with a 4-item scale, including innovations, coping with unforeseen events, quality of contacts, and overall success. Cronbach's alpha for this scale was .63. For every project, team effectiveness measures were computed by averaging individual scores. This is methodologically justified in that ϵ^2 values of all four effectiveness variables were high, ranging between .48 and .70.

Favorable Context Conditions. Because the context in which software development takes place may have an impact on team effectiveness (Heinbokel, Frese, Stolte, Brodbeck, & Sonnentag, 1996), an index was developed consisting of three context indicators: size of project team, phase of development process at Time 1, and user involvement. Context conditions were coded as favorable when there was no user involvement and when not more than one of the other unfavorable conditions (large in size, late phase) was present. All other projects were regarded as having to deal with an unfavorable context.

2.3. Analyses

Data were analyzed using hierarchical regression analyses that were computed separately for each of the four team effectiveness measures. In the first step, favorable context was entered in the equation as a control variable. In the second and third step, use of design methods and team leaders' goal orientation, respectively, were included in the equation.

3. RESULTS

Table 2 presents the results from hierarchical regression analyses. Favorable context conditions explained significant proportions of variance in team efficiency (30%) and success of project (20%), but not in the other two effectiveness measures (changeability of the software product and maintaining schedule and budget).

Table 2: Results of Hierarchical Regression Analysis for Predicting Team Effectiveness

Criterion	Predictors	R ²	R ²	B	SE B	β
Changeability of product	Step 1	.07	.07	.60	.47	.26
	Favorable context					
	Step 2	.43***	.36***	.53	.38	.23
	Favorable context			1.64	.45	.60***
	Use of design methods					
	Step 3	.55***	.12**	.60	.35	.26
	Favorable context			1.42	.42	.52***
	Use of design methods			.68	.30	.35**
	Goal orientation					
	Step 1	.36***	.36***	1.78	.51	.60***
Team effectiveness	Favorable context					
	Step 2	.57***	.21***	1.71	.43	.58***
	Favorable context			1.62	.50	.46***
	Use of design methods					
	Step 3	.72***	.16***	1.81	.35	.61***
	Favorable context			1.30	.42	.37***
	Use of design methods			1.01	.30	.41***
	Goal orientation					

(continued)

Table 2 (Continued)

Criterion	Predictors	R ²	R ²	B	SE B	β
Maintaining schedule and budget	Step 1	.16*	.16*	.53	.26	.40*
	Favorable context					
	Step 2	.16*	.00	.53	.27	.40*
	Favorable context					
	Use of design methods			-.06	.31	-.03
	Step 3	.33**	.18**	.58	.24	.43**
Success of project	Favorable context			-.22	.29	-.14
	Use of design methods			.48	.21	.43**
	Goal orientation					
	Step 1	.20**	.20**	.43	.18	.44**
	Favorable context					
	Step 2	.22*	.02	.42	.19	.43**
Use of design methods	Favorable context			.18	.22	.15
	Use of design methods					
	Step 3	.29*	.07	.44	.18	.46**
	Favorable context			.11	.22	.09
	Use of design methods			.22	.22	.09
	Goal orientation			.22	.16	.28

* $p < .10$. ** $p < .05$. *** $p < .01$.

Use of design methods was entered in the second step and significantly explained additional variance of changeability of the product and team efficiency. Thus, projects that used software design methods systematically arrived at a higher changeability of the product and higher team efficiency. However, use of design methods was not a significant predictor for either maintaining schedule and budget or success of project. Thus, Hypothesis 1 was partially supported by the data.

When including team leaders' goal orientation in the third step, an additional 12% of the variance in changeability and 16% of team efficiency was explained. Thus, team leaders' goal orientation contributed substantially to these effectiveness measures, in addition to the use of design methods.

Furthermore, team leaders' goal orientation accounted for an additional 18% of variance of maintaining schedule and budget. Thus, teams with highly goal-oriented leaders were more successful in maintaining schedule and budget than were teams with low goal-oriented leaders. However, team leaders' goal orientation was not a significant predictor for success of a project. From these results, it can be concluded that Hypothesis 2 was also partially supported by the data.

It is important to mention that both use of methods and high goal orientation in team leaders contributed independently to high team effectiveness as shown by the fact that use of design methods stayed significant after goal orientation was entered into the regression equation.

4. DISCUSSION

The study showed that the use of software design methods predicted changeability of the software product and team efficiency. In addition, team leaders' goal orientation accounted for a substantial proportion of changeability, team efficiency, and maintaining schedule and budget. This indicates that both the application of design methods and team leaders' action style are important for high effectiveness in software development.

Thus, at least for changeability of the product and team efficiency, both factors—use of design methods and goal orientation—made an independent contribution to the criterion measures. This implies that these factors do not replace but rather complement each other. A reason for this complementary effect is that software design methods and goal orientation play different roles in the development process: Methods provide guidelines on how to proceed in technical respects and therefore lead to a certain degree of standardization. In contrast, team leaders' action styles affect the way of pursuing goals in the daily work process, for example, in case of difficulties and obstacles.

The differential effect of the two independent variables may also explain that maintaining schedule and budget is only influenced by team leaders' goal orientation but not use of design methods. There are three possible explanations: First, goal-oriented team leaders seem to use adequate organizational strategies more frequently. Second, the development of methods and procedures itself takes time (Curtis, Krasner, & Iscoe, 1988). Therefore, possible gains associated with the use of methods are reduced. Third, projects that utilize design methods to a high extent

may plan time and costs more tightly because they think methods economize the development process.

No effects of the design methods and goal orientation on project success were found. This might be due to the nature of this effectiveness measure. Project success depends not only on the processes that take place within the project, such as methods used or leadership behavior. Overall success of a project is to some extent also associated with factors over which team members do not exert any control (Brodbeck, 1996). For example, Gladstein (1984) found that outcome effectiveness of sales teams was mainly predicted by market growth and not by factors within the teams studied. Within the context of this study such situational constraints might have been the situation in the users' department or users' organization, marketing policy, or power relations inside and outside the organization.

Studies that use only questionnaire measures face the problem of common method variance (Campbell & Fiske, 1959) and therefore might overestimate the relation between variables studied. This applies to a certain degree to this study as well. However, by using aggregated measures, individual response bias (e.g., between use of methods and effectiveness measures) was reduced. In addition, the samples in each of the participating groups in Time 1 and Time 2 were only partially overlapping. Nevertheless, this does not rule out the possibility of team response bias reflecting, for example, a sort of positive or negative team climate (cf. Brodbeck, 1996). However, use of design methods was assessed in a rather "objective" way (Frese & Zapf, 1988) by asking specifically about the existence of certain written criteria, procedures, or rules. Thus, when answering the items, respondents had to refer to "hard facts" and not just to the climate within the team. Thus, the results are most likely due to substantial relations and not only to methodological artifacts.

This study suggests the following practical implications. First, software development projects should apply design methods systematically; that is, provide or develop specific procedural rules that are standardized for all team members. Such methods should be available in written form. Second, team leaders' goal orientation should be taken into consideration during selection procedures. If necessary, one could also think of improving goal orientation by training. Important for practical reasons is that lack of goal orientation in team leaders cannot be compensated by the extensive use of design methods. However, lack of methods can also not be replaced by high goal orientation in team leaders. Thus, for making software development projects effective, it is crucial to optimize both the technical and managerial aspects of the development process.

REFERENCES

- Boehm, B. W. (1981). *Software engineering economics*. Englewood Cliffs, NJ: Prentice Hall.
- Boehm-Davis, D. A., & Ross, L. S. (1992). Program design methodologies and the software development process. *International Journal of Man-Machine Studies*, 36, 1-19.
- Brodbeck, F. C. (1993). *Kommunikation und Leistung in Projektarbeitsgruppen: Eine empirische Untersuchung an Software-Entwicklungsprojekten* [Communication and performance in software development projects]. Doctoral dissertation, University of Giessen, Department of Psychology, Giessen, Germany.

- Brodbeck, F. C. (1996). Criteria for the study of work group functioning. In M. A. West (Ed.), *Handbook of work group psychology* (pp. 285–315). New York: Wiley.
- Budgen, D. (1994). *Software design*. Wokingham, England: Addison-Wesley.
- Campbell, D. T., & Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, *56*, 81–105.
- Coad, P., & Yourdon, E. (1991). *Object-oriented design*. Englewood Cliffs, NJ: Prentice Hall.
- Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*, 1268–1287.
- Endler, N. S., & Hunt, J. M. (1966). Sources of behavioral variance as measured by the S–R inventory of anxiousness. *Psychological Bulletin*, *65*, 336–346.
- Frese, M., Albrecht, K., Kreuscher, R., Papstein, P. V., Prümper, J., & Schulte-Göcking, H. (1995). Handlungsstile und Leistungsverhalten: Die Rolle von Plan- und Zielorientierung in Problem- und Lernsituationen [Action styles and performance: The role of planfulness and goal orientation in problem solving and learning studies]. *Zeitschrift für Arbeits- und Organisationspsychologie*, *39*, 67–77.
- Frese, M., & Hesse, W. (1993). The work situation in software development: Results of an empirical study. *Software Engineering Notes*, *18*(3), 65–72.
- Frese, M., Stewart, J., & Hannover, B. (1987). Goal orientation and planfulness: Action styles as personality concepts. *Journal of Personality and Social Psychology*, *52*, 1182–1194.
- Frese, M., & Zapf, D. (1988). Methodological issues in the study of work stress: Objective vs. subjective measurement of work stress and the question of longitudinal studies. In C. L. Cooper & R. Payne (Eds.), *Causes, coping, and consequences of stress at work* (pp. 375–411). Chichester, England: Wiley.
- Gladstein, D. L. (1984). Groups in context: A model of task group effectiveness. *Administrative Science Quarterly*, *29*, 499–517.
- Glass, R. L., Vessey, I., & Conger, S. A. (1992). Software tasks? Intellectual or clerical? *Information & Management*, *23*, 183–191.
- Grudin, J. (1991). Systematic sources of suboptimal interface design in large product development organizations. *Human-Computer Interaction*, *6*, 147–196.
- Heinbokel, T., Frese, M., Stolte, W., Brodbeck, F. C., & Sonntag, S. (1996). Don't underestimate the problems of user centredness in software development projects—There are many! *Behaviour & Information Technology*, *15*, 226–236.
- Hill, S. C. (1970). A natural experiment on the influence of leadership behavior patterns on scientific productivity. *IEEE Transactions on Engineering Management*, *17*, 10–20.
- Kerr, S., & Jermier, J. M. (1978). Substitutes for leadership: Their meaning and measurement. *Organizational Behavior and Human Decision Processes*, *22*, 375–403.
- Knorr, K. D., Mittermeir, R., Aichholzer, G., & Waller, G. (1979). Leadership and group performance: A positive relationship in academic research units. In F. M. Andrews (Ed.), *Scientific productivity: The effectiveness of research groups in six countries* (pp. 95–120). Cambridge, England: Cambridge University Press.
- Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM*, *38*, 69–81.
- Lee, A., & Pennington, N. (1994). The effects of paradigm on cognitive activities in design. *International Journal of Human-Computer Studies*, *40*, 577–601.
- Podsakoff, P., Mackenzie, S. B., & Fetter, R. (1993). Substitutes for leadership and the management of professionals. *Leadership Quarterly*, *4*, 1–44.
- Podsakoff, P. M., Niehoff, B. P., MacKenzie, S. B., & Williams, M. L. (1993). Do substitutes for leadership really substitute for leadership? An empirical examination of Kerr & Jermier's situational leadership model. *Organizational Behavior and Human Decision Processes*, *54*, 1–44.

- Pressman, R. S. (1991). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.
- Saarinen, T. (1990). System development methodology and project success. *Information & Management, 19*, 183-193.
- Scott, S. G., & Bruce, R. A. (1994). Determinants of innovative behavior: A path model of individual innovation in the workplace. *Academy of Management Journal, 37*, 580-607.
- Sonnentag, S., Frese, M., Stolte, W., Heinbokel, T., & Brodbeck, F. C. (1994). Goal orientation of team leaders: Its effects on performance and group interaction in software development projects. *The European Work and Organizational Psychologist, 4*, 153-168.
- Thamhain, H. J., & Wilemon, D. L. (1987). Building high performing engineering project teams. *IEEE Transactions on Engineering Management, 34*, 130-137.
- Wasserman, A. I. (1980). Information system design methodology. *Journal of the American Society for Information Science, 31*(1), 5-24.
- Yourdon, E., & Constantine, L. L. (1979). *Structured design*. Englewood Cliffs, NJ: Prentice Hall.