Assessing the Effects of Facilitated Workshops in Requirements Engineering

Joost Schalken Sjaak Brinkkemper Hans van Vliet
Vrije Universiteit
Department of Computer Science
De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands
E-mail: {jjp.schalken,s.brinkkemper,jc.van.vliet}@few.vu.nl

Abstract

The role of facilitated workshops in the development of information systems is widely acknowledged, but apart from some anecdotic evidence little objective information about the effectiveness of facilitated workshops is available. This paper describes a study within a single organization in which the effectiveness of requirements engineering in projects using facilitated workshops is compared with the effectiveness of past projects that used one-on-one interviews to gather requirements instead of facilitated workshops.

In the study the duration, effort and satisfaction of 49 DSDM projects using facilitated workshops have been compared with 20 projects that used the Method/1 method, which use one-on-one interviews to gather the requirements. For small projects, Method/1's one-on-one interviews are found to be more efficient, whereas for larger projects the efficiency of one-one-one interviews is surpassed by DSDM's facilitated workshops. Other quantitative effects were not found, and subjective ratings of stakeholders do not indicate a preference for DSDM projects either. We conjecture that these findings may be partly due to the fact that we only measured short-term effects (such as customer satisfaction immediately after a project is completed), and that the true benefits of facilitated workshops could require longitudinal studies.

1. Introduction

Facilitated workshops are intensive meetings in which technical staff, end-users and management collaborate on information systems development tasks, such as project planning, requirements specification and user interface design. Facilitated workshops fit in the general tendency to increase the involvement of stakeholders in the requirements engineering process. The use of facilitated workshops in information system development has been introduced by Chuck Norris in 1977 [17], based on the work of Michael Doyle and David Strauss [8]. Facilitated workshops are best known for being a crucial component of *Joint Application Development* [12, 11] and *Participatory Design* [4], but facilitated workshops are also used in other development methodologies.

Facilitated workshops are led by a *facilitator*. A facilitator is a leader of workshops, trained in group dynamics, that is 'free from vested interests' [14] as cited in [10]. Although not all facilitators perform the same role, some common patterns in the role of a facilitator can be distinguished.

1.1. Evaluating the effectiveness of facilitated workshops

The costs of facilitating workshops are considerable. It is therefore reasonable to ask what benefits can be expected from the introduction of facilitated workshops. The costs consist of both introduction costs (training workshop facilitators and educating IS staff and end users about facilitated workshops) and operating costs (the time of both participants in the facilitated workshops and that of the workshop facilitator plus the required facilities).

In manuals on rapid application development and facilitation in the popular press many benefits of facilitated workshops have been reported. These benefits include [13]: commitment from top executives to the software planning-process, shortening requirements specification phase, eliminating features of questionable value, helping to get the requirements right the first time, helping to get the user interface right the first time, reduction of organizational infighting. Some attempts have been made in manuals and the popular press to quantify time savings, costs savings, completeness of requirements collection and satisfaction (see for example [4, 11]).

Despite the claims made, little is known about the *real efficacy* of the workshop approach [7]. In most popular studies no details are disclosed about the research methodology used and the data obtained [4], which makes the assessment of the validity problematic at best. This lack of objective data on facilitated workshops might be explained because facilitation of workshops has been developed outside the academic world [4] and because most organizations lack the required data on systems development to allow evaluation of facilitated workshops [7]. More academic effort has been put into the evaluation of facilitation in the context of group support tools (see for example [6]). Facilitated workshops supported by tools and focusing on analysis models (such as ER-diagrams and UML diagrams) have been termed *Rapid JAD* by Martin [12]. The debate whether facilitated workshops focusing on models and tools are always an improvement is still open [7].

1.2. Research questions and results

This study attempts to add to the body of knowledge about facilitated workshops by evaluating the effectiveness of facilitated workshops, by asking the following three questions:

1. What is the advantage of facilitated workshops in terms of the quality of the requirements when compared to one-on-one interviews?

2. What is the advantage of facilitated workshops in terms of calendar time when compared to one-on-one interviews?

3. What is the advantage of facilitated workshops in terms of effort (in requirement engineering) versus quality (of the requirements) when compared to one-on-one interviews?

Our results only show a significant effect of facilitated workshops for larger projects. For smaller projects, the added effort needed to conduct facilitated workshops seems not to outweigh possible other positive effects. We did not find the expected positive effect of facilitated workshops on stakeholder satisfaction.

2. Related work

Our study focuses on the use of facilitated workshops in the context of traditional software development. Studies on the use of facilitation in the context of meetings supported by Computer Supported Cooperative Work tools or Integrated CASE tools, as discussed in section 1, are not taken into consideration.

The first scholarly study we are aware of is the work performed in 1977 by Unger and Walker [19]. This study reports a case study on an operating systems course, in which the students partially implemented an operating system as a practical assignment for the course. The course, in this form, has been offered four times. In two of the courses, the students were supported in their communications by a professional facilitator schooled in group dynamics. The two groups that were assisted by a professional facilitator were approximately twice as productive (based on the number of lines of code delivered and the amount of effort expended) when compared with the groups that did not receive any facilitation.

This study covered the full development life-cycle and not just the requirements engineering phase. Since the students had to come up with the requirements (lacking an external sponsor) the problems faced during the project also included requirements analysis. The effort spent by the students during the practical assignment has not been formally tracked, but instead estimated based on the study load of the students. This makes the measurement of the expanded time less reliable. Another drawback of this study is that it is performed on college students and therefore the external validity of the study is not evident.

In 1999 Davidson, observing the lack of transparent, objective evaluations of the benefits of using Joint Application Development to gather requirements specifying systems properties, performed a study into the efficacy of JAD as a software process improvement method [7]. As the organizations investigated by Davidson did not collect statistical data on development time, costs and errors, no quantitative evaluation of the benefits were possible. Instead, Davidson conducted 34 interviews with facilitators and managers (covering 20 distinctive projects) to investigate the users' evaluations of the JAD method. This study reports that only 10% of the informants believed the requirements had been specified faster using JAD and none of the informants believed the total system development time had been reduced.

The strongest point of Davidson's study is that it does not only take productivity into account, but also the less tangible factor of user satisfaction, which is a derived measure for the quality of the process. The study also attempts to explain why the benefits of JAD have not been fully reached in the projects examined. Unfortunately no objective, quantified measurements are available to determine whether development time and costs have dropped or not. As this study contradicts previous studies that claim higher productivity, it would have been useful to be able to determine if the development time and costs really remained the same or were only believed to remain the same. This is why objective, quantified metrics are needed in method evaluations. In the study described in this paper we therefore use objective, quantified measurements to objectively assess the schedule and productivity effects.

Hubbard, Schroeder and Mead performed a comparison between facilitator-driven requirements collection processes and conventional interview methods [9]. In this study the researchers compare eight projects using facilitator-driven requirements collection with four projects using the conventional, unstructured interview technique. For each of the projects the yield (number of requirements), effort and duration was registered. Based on an analysis of the data, the researchers found that the effort to obtain the same amount of function points was reduced by a factor of 2.8 for facilitator-driven requirements collection and the duration per function point has decreased by a factor of 9.8 (both results are significant at the 0.05 α-level).

Our study is similar to the one performed by Hubbard et al. We pay attention to not only the quantitative indicators of the performance of the project, but also to subjective information. By repeating a quantitative study into the effects of facilitator-driver requirements collection, confidence is built that the result obtained by Hubbard, Schroeder and Mead will also be obtained in a different context.

3. Research methodology

3.1. Context of the research

This study has been performed within an internal Information Technology department of a large financial institution. In this department over 1500 people are employed. The organization primarily builds and maintains large, custom-built, mainframe transaction processing systems, most of which are built in COBOL and TELON (an application-generator for COBOL). Besides these mainframe systems, a large variety of other systems are implemented, constructed and maintained by the organization. These systems are implemented in a large variety of different programming languages (such as Java and COOL:Gen), run under various operating systems (such as Microsoft Windows and UNIX) and are distributed over different platforms (batch, block-based, GUI-based and browser-based).

The organization is currently undertaking a major software process improvement program (SPI) to improve the internal IT processes and cooperation with the business. The SPI program includes the introduction of the Dynamic Systems Development Method (DSDM) [18], a quality system that complies with the requirements of CMM [16] level 2, the introduction of a software metrics program, and a culture change program.

DSDM is a rapid application development method that is suitable for incremental and iterative development, but the method can also be used for linear development projects. In the organization DSDM replaces Method/1 Custom Systems Development, a proprietary method of Arthur Andersen [1, 2]. The organization's previous method, Method/1, had been tailored to meet the organization's need of the time. Through the introduction of DSDM a new technique for eliciting requirements was institutionalized as well: facilitated workshops.

In the organization a policy states that projects must start by gathering requirements using facilitated workshops in the Business Study phase and the Functional Model Iteration phase (see section 3.2 for more information on the phases of DSDM). The organization has a number of trained workshop facilitators, who are available for these workshops (more on the facilitation process in use can be found in [3, 20]).

3.2. Comparing different development cycles

To investigate the effects of facilitated workshops projects performed with the Method/1 Custom Systems Development method must be comparable to ones that are run with the new DSDM method. A mapping of the two methods to a more generic model is therefore required to allow meaningful comparisons between the old and new projects (see figure 1). Based on the intent of the phases and the intent of the deliverables of the phases, the Quick Scan and Project Proposal phases of Method/1 have been mapped to the Feasibility Study and Business Study phases of DSDM. Furthermore the Functional Design phase of Method/1 has been mapped to the Functional Model Iteration phase of DSDM. This way two requirements engineering phases are created: a *high-level requirements engineering phase* and a *detailed-requirements engineering phase*.

3.3. Data collection

Two sources of evidence are used to assess the merits of facilitated workshops: the required effort and the duration of the low-level requirements engineering phase and the size (in NESMA function points [15]) is extracted from the organization's project administration database. The satisfaction of the project leaders, customers and IT employees about the project is determined from the organization's project evaluation database.

To assess whether projects that employed facilitated workshops are really more efficient in terms of required effort, the organizations' project database has been used. The organization has implemented DSDM gradually into the organization. As a consequence, Method/1 and DSDM were used next to each other for a period of over two years, the period to which our data refer. All DSDM projects using facilitated workshops have been compared to Method/1 projects that used the traditional one-on-one interview technique to gather project requirements. Since the terminology and phase names in these two methods are not equal, the

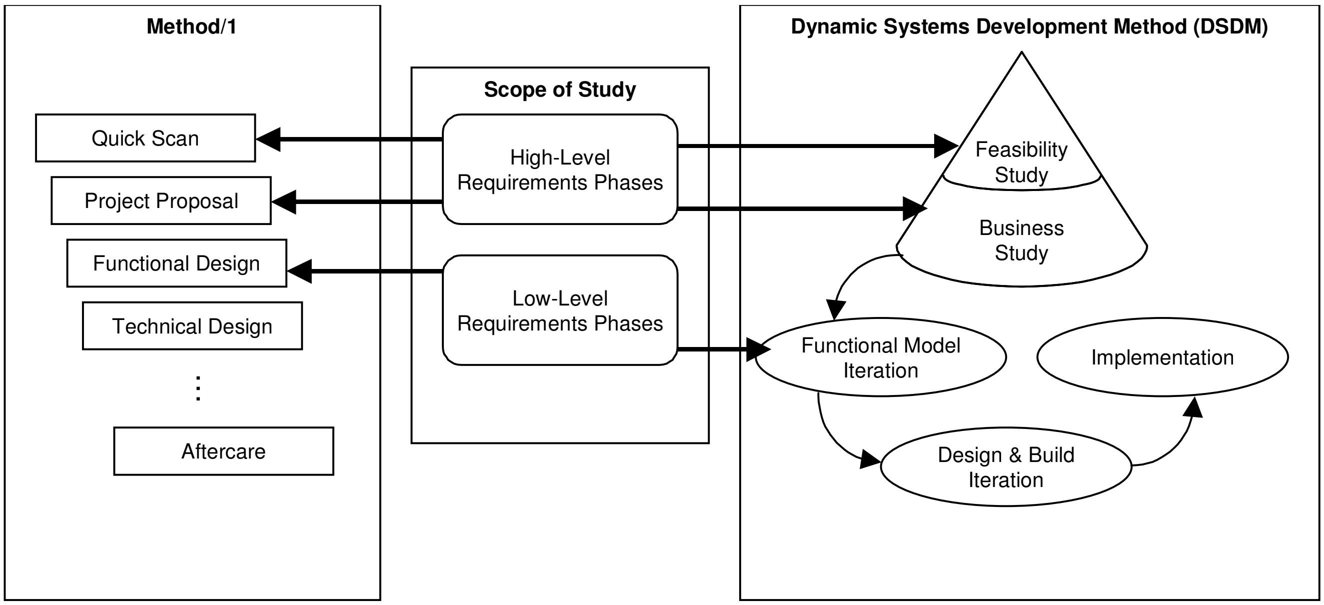

Figure 1. Mapping Method/1 phases to DSDM phases

mapping, outlined in section 3.2, has been used to compare DSDM projects with Method/1 projects. A comparison of both high-level and low-level requirement specification phases was planned. Upon examining the project database comparing high-level requirements specification phases proved infeasible, as a large proportion of projects at the organization share the high level requirements phase between multiple small implementation projects. Attributing the effort of the collective high level requirements engineering phase to the individual development projects would require a lot of speculation. Therefore only attention is paid to the low-level requirement specification phase.

The only reliable effort data in the project database concerns that of IT personnel. So we only take those into account in our effort equations below. In particular, effort spent on requirements engineering by non-IT personnel, such as business representatives, is not taken into consideration.

In order to gain insight into the subjective opinions regarding the use of the DSDM method as compared to Method/1 projects, we also used a database with the evaluations and lessons learned about past projects. At the end of each project a mandatory evaluation of the project is conducted using an electronic tool. The evaluation consists of both a list of closed questions that are rated on a scale from 1 to 10 and of a list of open questions regarding the project. The project manager, the customer and the IT employees all participate in the evaluation of projects. For this research closed questions from the evaluation form have been selected that deal with subjective ratings of the requirements clarity and with subjective ratings that deal with overall project satisfaction. For the latter category only the project managers and customers rated the project.

The data from the project administration database and the evaluation database have been extracted and converted into a format suitable for statistical analysis using S-Plus version 6.0.

4. Discussion of results

In this section we discuss the analysis of effort, duration and satisfaction in both types of projects. Effort concerns the number of hours spent by IT personnel on requirements activities. Duration concerns the lead time of the requirements phases, as recorded in the project database. Satisfaction concerns the subjective ratings of the stakeholders regarding project results and requirements clarity.

4.1. Effort

Comparing the hours of effort spent in low-level requirements analysis for both DSDM projects with facilitated workshops and Method/1 projects that did not use facilitated workshops at first does not yield much difference. In figure 2 a scatter plot is given of hours spent on requirements engineering per function point. Although some of the Method/1 projects perform considerably worse than the DSDM projects, no obvious pattern can be discerned.

If we however plot the productivity in requirements engineering, in terms of hours spent by ICT personnel per func-

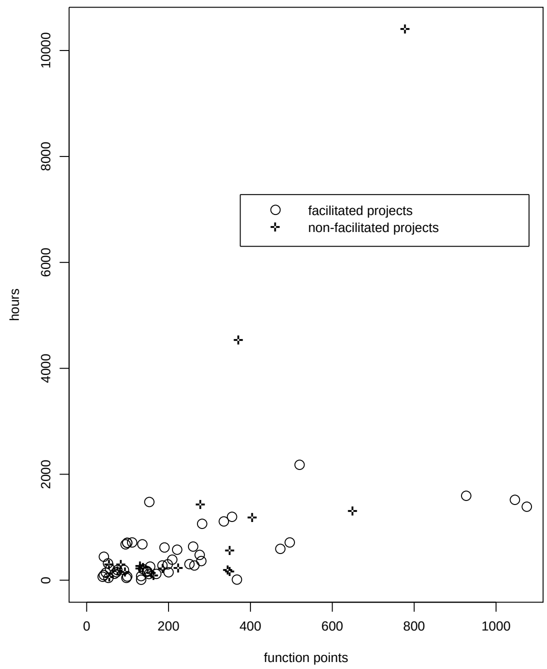

Figure 2. Effect of technique and size on the required effort of requirements engineering

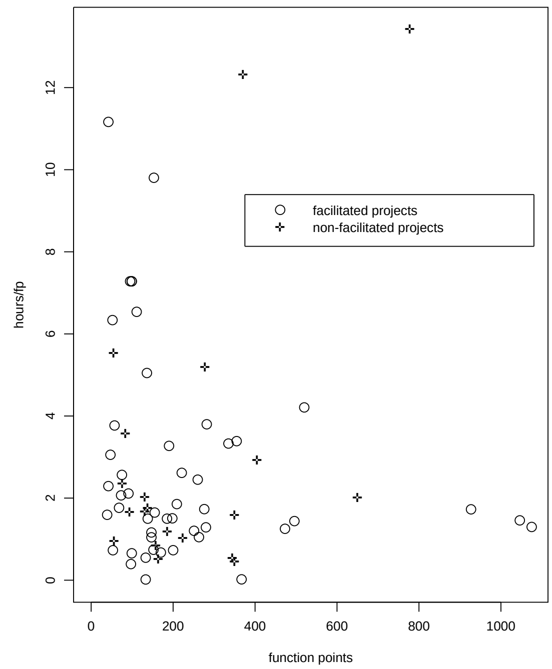

Figure 3. Effect of technique and size on the effectiveness of requirements engineering

tion point against the size of the projects, a pattern does appear. When we examine only the small software development projects some projects using facilitated workshops perform worse than their traditional Method/1 counterparts. When we examine large projects the opposite can be observed: here Method/1 projects under-perform while facilitated workshops are more productive.

To determine whether small projects are really better off without facilitated workshops and large project are really better off with facilitated workshops, the statistical significance of the pattern is tested with an ANOVA analysis. If the effect caused by the project size (in function points) on the effectiveness of the requirements engineering technique (facilitated workshops or one-on-one interviews) cannot be explained by chance alone, the interaction between the factors technique and size should be significant.

The dependent variable productivity is severely left-skewed. To obtain an approximately normally distributed variable, the dependent variable has been transformed with an square root transformation. The transformed variable has been used in the ANOVA analysis.

When testing the interaction effect between requirements engineering technique and project size with an ANOVA test, the interaction effect (technique:size in table 1) turns out to be significant well beyond the 0.05 α-level. The first line (labeled 'technique'), reflects the influence of the requirements engineering method on total effort. The second line (labeled 'technique:size') reflects the influence of project size on the above relation.

Using regression analysis, the turning point where Method/1 projects become less productive than DSDM projects with facilitated workshops can be determined. For our set of projects, the regression lines intersect at 171.4 function points. This means that requirements engineering using Method/1 is more productive for projects smaller than the threshold of 171.4 function points.

Table 1. ANOVA analysis of the effects of technique and size on requirements engineering effort per function point

	df	*SSE*	*MSE*	*F*	*p(F)*
technique	1	0.06	0.06	0.11	0.7444
technique:size	1	3.25	3.25	6.19	0.0154
residuals	64	33.58	0.52		

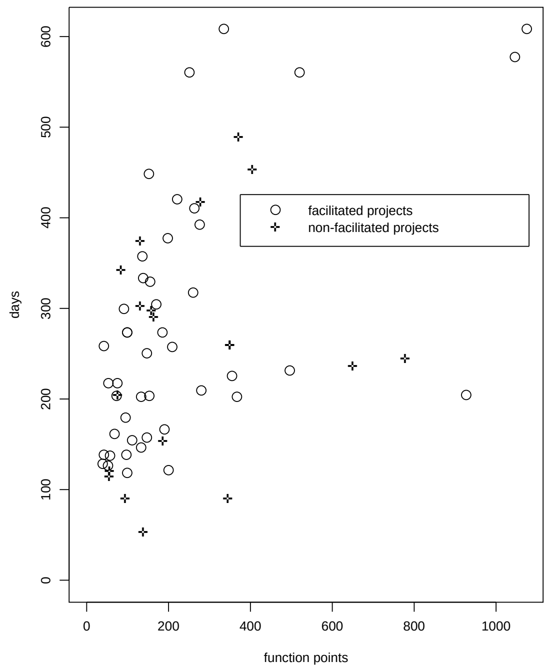

Figure 4. Effect of technique and size on the required calendar time of requirements engineering

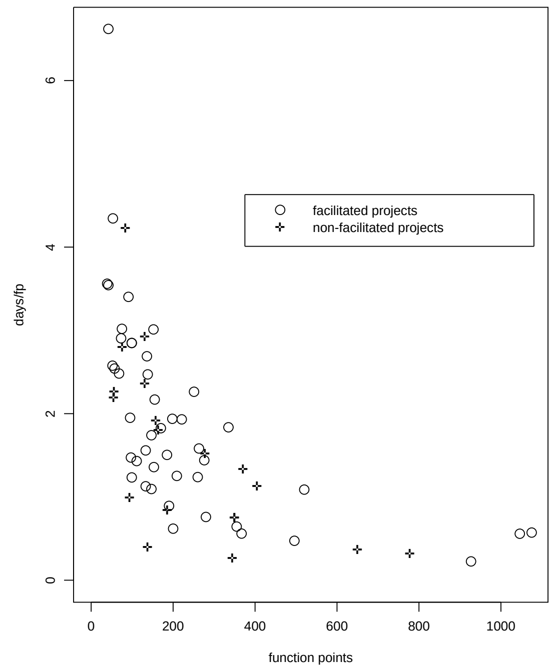

Figure 5. Effect of technique and size on the required calendar time of requirements engineering per function point

4.2. Duration

To compare the difference in duration in low-level requirements analysis for both DSDM projects with facilitated workshops and Method/1 projects that did not use facilitated workshops the date of the earliest activity in the low-level requirements engineering phase has been subtracted from the date on which the last activity in the low-level requirements engineering phase took place. This yields the duration of the low-level requirements engineering phase in days.

Examining the scatter plots in figure 4 and 5 reveals that no significant differences occur between the duration of projects that used facilitated workshops and those that did not.

An ANOVA analysis of the data (see table 2) confirms that the factor technique does not have a significant influence on the duration of requirements engineering and the interaction between requirements engineering technique and size (as we did see for effort in section 4.1) is also absent. As the dependent variable duration per function point is somewhat left-skewed, the variable has been transformed with an square root transformation before it has been used in the ANOVA analysis. An ANOVA analysis on the non-transformed variable yields similar results.

Table 2. ANOVA analysis of the effects of technique and size on requirements engineering duration per function point

	df	*SSE*	*MSE*	*F*	*p(F)*
technique	1	0.37	0.37	3.55	0.0645
technique:size	1	0.07	0.07	0.65	0.4216
residuals	60	6.19	0.10		

4.3. Satisfaction

To determine if the requirements engineering technique (one-on-one interviews or facilitated workshops) influences the satisfaction of either the customer, project manager or IT employee, information from the project evaluation database has been collected for the projects that have also been used

Table 3. Project evaluation questions

Question	*Project manager*	*Customer*	*IT personnel*
Q1: Functionality is delivered according to agreements	X	X	
Q2: The delivered results are according to my expectations	X	X	
Q3: The delivered results meet the quality requirementsas agreed beforehand	X	X	
Q4: The requirements were clear	X	X	X

in the above analyses.

The evaluation database contains both quantitative and qualitative information about the perceived success of the project, from the viewpoint of each of the stakeholders. At the end of a project, the project manager and the customer are allowed to rate, amongst others, their satisfaction about the results of the project and about the requirements clarity. IT employees rate their satisfaction about the requirements clarity only. The set of questions asked and the categories of persons concerned are listed in table 3. Unfortunately not for all projects an evaluation has been conducted and in some of the projects not the full set of questions has been answered.

The results of the MANOVA analysis of the (combined) set of questions relating to satisfaction are given in table 5. In the same vein, the results with respect to requirements clarity are listed in table 4.

Table 4. ANOVA analysis of satisfaction about requirements clarity

	df	*SSE*	*MSE*	*F*	*p(F)*
viewpoint	2	3.76	1.88	1.83	0.1659
technique	1	0.02	0.02	0.02	0.8902
viewpoint:technique	2	1.51	0.75	0.74	0.4813
residuals	94	96.52	1.03		

It appears stakeholders are more satisfied about the results of Method/1 projects than they are about the results of DSDM projects. This is contrary to what we expected. A closer inspection of the individual questions as listed in table 3 gives a potential explanation. For both Q1 and Q3, Method/1 projects score higher than DSDM projects, and the project leader gives higher rates than the customer. For question Q2, the project leader and customer rate Method/1 projects equally high. However, project leaders rate DSDM projects (slightly) higher than Method/1 projects on this question, while customers rate DSDM projects quite a bit lower on this question. We hypothesize the following:

- project leaders do not expect too much from yet another method imposed by senior management. They've seen it all before.
- customers on the other hand *do* expect a positive effect from their involvement in the workshops.
- the net effect of both quality and functionality of DSDM projects is perceived to be a bit lower than for Method/1 projects.
- this mismatch between expectations and actual achievements results in lower scores for Q2, especially so for the customer.

Table 4 clearly reveals that different stakeholders do not rate requirements clarity differently for Method/1 and DSDM projects.

4.4. Threats to validity

In this section we will discuss the potential threats to the validity of our study. For this discussing, we use the framework presented by Cook & Campbell [5, p. 37–39] which distinguishes four categories of validity: *internal validity*, *statistical validity*, *external validity* and *construct validity*. As the statistical validity of the study has been discussed in sections 4.1, 4.2 and 4.3 this issue will not be repeated over here.

Threats to the internal validity of a study are the uncontrolled nuisance or background factors that could have an effect on the outcomes of the study and therefore present an alternative explication to the observed effect. In our study the internal validity of the study can be jeopardized by factors that influence the the required effort or duration of the project or that influence the satisfaction of the employees.

The amount of experience in the requirements gathering technique (facilitated workshops or one-on-one interviews) might be a confounding factor, as the IT personnel had more experience in one-on-one interviews than they had in facilitated workshops. To minimize this effect the organization

Table 5. MANOVA analysis of stakeholder satisfaction regarding project results

	df	*Pillai Trace*	*approx. F*	*num df*	*den df*	*p(F)*
viewpoint	1	0.19	4.18	3	53	0.0099
technique	1	0.07	1.30	3	53	0.2847
viewpoint:technique	1	0.32	8.30	3	53	0.0001
residuals	55					

provided a three day course to all IT personnel to learn to use the new project methodology and facilitated workshops. Besides training, all new projects have been supported by consultants that had extensive prior experience using facilitated workshops. Although this might not have eliminated the total impact of difference in experience using a certain requirements gathering technique, these measures did reduce their impact.

To prevent negative effects on project effort, duration or satisfaction caused by incapable or inexperienced workshop facilitators, the organization created a pool of workshop facilitators. Before employees were allowed to enter the facilitators pool, they were either to possess previous facilitation experience or to have followed a standard course for workshop facilitators [3]. This reduced the effect of the facilitor's experience on the outcome of the facilitated workshop.

After a small number of trial projects, which have been excluded from this study, the organization has implemented the DSDM methodology in a department by department way. The project employees could not choose to use either DSDM and facilitated workshops or Method/1 and one-on-one interviews. As soon as a department implemented DSDM all new projects were to use DSDM. This measure makes it unlikely that motivation of project staff, the suitability of facilitated workshops for the project or other project characteristics had a significant influence on the relation between requirements gathering technique and project outcomes (in terms of required effort, duration and satisfaction).

Threats to the external validity of a study reduce the generalizability of the study's outcomes to domains outside the scope of study. This study has been performed in just a single organization. The organization develops applications in a very wide range of programming languages, platforms and operating systems. As different programming languages, platforms and operating systems have been used during the study, the results should be applicable to different technological environments.

The organization is however a very large internal IT department that is part of an even larger overall organization. Both the IT department and the surrounding business environment can be characterized as being a professional bureaucracies. Further studies are needed to determine if the effects of the requirements gathering technique found in this study will also be found in medium to small IT departments, in less bureaucratic environments and in companies for whom IT is not a service but a core business.

A last threat to the validly of the study is posed by the question whether efficiency is the right measure to determine the benefits of a certain requirements gathering technique. This issue touches the construct validity of the study. One could argue that a good requirements gathering technique should not strive for efficiency, but instead should strive for effectiveness. Instead of eliciting the most requirements per unit of effort, one should perhaps aim to elicit the most requirements of high value or high quality per unit of effort.

Unfortunately it is extremely hard to measure the quality of a requirement. To judge the value or quality of a requirement one would need to measure the benefits of a system and the evolution of that system during its use. This would require an extensive, longitudinal study of system development projects.

5. Conclusion

Care must be taken not to draw too bold conclusions. The data analysis shows that neither of the two requirements engineering techniques is more effective in every situation. The significant interaction effect between the requirements engineering technique used and the size of the software project indicates that for large projects facilitated workshops offer greater productivity, whereas for smaller projects the one-on-one requirements engineering interviews appear to be more productive. This need not come as a big surprise. Facilitated workshops incur some overhead (employing a facilitator and organizing the workshops) that do not outweigh their benefits for small projects.

One reason for not finding significant positive effects may be due to the fact that we only have data on the short-term effects of facilitated workshops, as perceived by various stakeholders. The effect of getting a better hold of the *real* requirements only shows after, and sometimes long after, the system has been made operational. It is only then that we see a plethora of change requests for ill-conceived systems. Next to that, facilitated workshops may have additional benefits that we did not measure, such as an improved mutual understanding between stakeholders. This is similar to what can be observed for other participatory activities, such as design reviews.

We have refined our model for factors that influence the benefits of different requirements engineering techniques to include such things as:

- the number of stakeholders involved.
- the number of requirements changed during requirements engineering.
- the number of conflicting requirements.

We are currently gathering data on a number of projects to quantify these effects.

Acknowledgments

This research is mainly financed by ABN AMRO Bank N.V. We thank the ABN AMRO Bank for her cooperation. We are especially grateful to Jean Kleijnen (ABN AMRO Bank) and Frank Harmsen (Cap Gemini Ernst & Young) for their valuable comments and input.

References

[1] Arthur Andersen. *Method/1 User Manual*. Arthur Andersen & Co., 1988.

[2] Arthur Andersen. *Foundation, Method/1: Quick Reference*. Arthur Andersen & Co., 1990.

[3] Cap Gemini Ernst & Young. *Facilitated Session Leader Training*. Cap Gemini Ernst & Youngi Academy, 2002.

[4] E. Carmel, R. D. Whitaker, and J. F. George. PD and joint application design: a transatlantic comparison. *Communications of the ACM*, 36(6):40–48, June 1993.

[5] T. D. Cook and D. T. Campbell. *Quasi-Experimentation: Design & Analysis Issues for Field Settings*. Rand McNally College Publishing Company, Chicago, IL, USA, July 1979.

[6] D. E. H. Damian, A. Eberlein, B. Woordward, M. L. G. Shaw, and B. R. Gaines. An emperical study of facilitation of computer-mediated distributed requirements negotiations. In *Fifth International Symposium on Requirements Engineering (RE '01)*, pages 128–135, Washington, DC, USA, Aug. 2001. IEEE Computer Society Press.

[7] E. J. Davidson. Joint application design (JAD) in practice. *Journal of Systems and Software*, 45(3):215–223, Mar. 1999.

[8] M. Doyle and D. Straus. *How to Make Meetings Work*. Penguin Putnam, Inc., New York, NY, USA, 1976.

[9] R. Hubbard, C. N. Schroeder, and M. N. R. An assessment of the relative efficiency of a facilitator-driven requirements collection process with respect to the conventional interview method. In B. H. Cheng, D. Weiss, and P. Hsia, editors, *4th International Conference on Requirements Engineering (ICRE'00)*, pages 175–188, Washington, DC, USA, June 2000. IEEE Computer Society Press.

[10] L. A. Macaulay. The role of the facilitator in requirements engineering. In *1998 Third International Conference on Requirements Engineering (ICRE '98)*, pages 244–245, Washington, DC, USA, 1998. IEEE Computer Society Press.

[11] J. Martin. Joint application design (JAD). In *Rapid Application Development*, chapter 8, pages 156–171. Macmillan Publishing Company, New York, NY, USA, 1991.

[12] J. Martin. The requirements planning phase (JRP). In *Rapid Application Development*, chapter 7, pages 138–155. Macmillan Publishing Company, New York, NY, USA, 1991.

[13] S. McConnell. Joint application development (JAD). In *Rapid Development: Taming Wild Software Schedules*, chapter 24, pages 449–463. Microsoft Press, Redmond , WA , USA, 1996.

[14] E. Mumford. The participation of users in systems design: An account of the origin, evolution and use of the ethics method. In D. Schuler and A. Namioka, editors, *Participatory Design: Principles and Practices*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, Apr. 1993.

[15] Nesma. *Definitions and counting guidelines for the application of function point analysis: A practical manual*. NESMA, Amsterdam, NL, 2.0 edition, Nov. 1997.

[16] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber. Capability maturity model for software, version 1.1. Technical Report CMU/SEI-93-TR-24, DTIC ADA263403, Software Engineering Institute, Feb. 1993. Available from: http://www.sei.cmu.edu/.

[17] M. G. Rush. Facilitation in the 90's. Newsletter 10, MG Rush Systems, Inc., Barrington, IL, USA, Feb. 1997.

[18] J. Stapleton. *Framework for Business Centred Development: DSDM Manual version 4.1*. DSDM Consortium, Ltd., Kent, United Kingdom, 2002.

[19] B. W. Unger and S. Walker. Improving team productivity in system software development. In *Proceedings of the fifteenth annual SIGCPR conference*, pages 104–115, New York, NY, USA, Aug. 1977. ACM Press.

[20] P. van Krugten-Elgersma and M. Hoogenboom. *Workgroups and Soft Skills*. Academic Service, 2000.