

**Virtual Teams: The Impact of Advanced Knowledge and Process Collaboration
Technologies on Software Development Team Processes and Performance**

Alistair McKinnon

2003

Copyright Reserved

Abstract

Virtual teams are becoming an important aspect of organisational life in many industries. In software development, virtual teams can be particularly useful as team members with the right skill set can generate code for a project no matter where they are in the world. However, this virtual setting often has limitations in terms of communication abilities, conflict, trust and performance. To improve the process of virtual software development, advanced knowledge and process collaboration technologies can be used. This proposal outlines an experiment to test the impact of using an advanced collaboration technology named MILOS in a virtual software development environment. It is hypothesised that by using MILOS as opposed to a less advanced collaboration technology, the measures of team trust, interaction quality and performance will be improved, with team conflict reduced.

Introduction

Virtual teams provide organisations with a valuable tool to achieve project success. There are a number of limitations to traditional co-located teams, including near-proximity requirements, limits on team sizes, friction between team members and outsiders, and security problems (Eom and Lee, 1999). By allowing team members to span national and organisational boundaries, many of these limitations can be removed. Virtual teams can provide access to necessary skills otherwise unavailable due to the ability to source from a wider labour pool (Cascio, 2000). A virtual team configuration can improve the balance between personal and professional lives for team members, and improve the flow and availability of information due to online collaboration technologies (Cascio, 2000). This powerful information processing ability can enable the creation of a highly flexible work unit. Given this utility of virtual teams in today's rapidly changing globalised marketplace, it is of interest to understand how different variables can impact on a team's process and performance.

In the creation of information systems (IS), virtual teams are becoming increasingly common place. This is particularly the case in the open source community. The Linux operating system¹ is an example, where programmers and end users from all over the world contribute software code and bug reports. In the business world, organisations that make use of virtual software development teams need to have a comprehensive collaboration system that allows members to communicate and exchange software source code and executable files. Current industry practice is to use relatively simple technologies for this task, including CVS² (an internet based online file and document distribution system), newsgroups and email (Maurer and Holz, 2002).

Maurer and Holz (2002) provide an improved software development collaboration system, named MILOS (Minimally Invasive Long-term Organizational Support). This system is currently freely available to use, although it is still under development. It significantly adds to what is current industry practice by integrating Microsoft Netmeeting functionality (process collaboration) and improved knowledge management. This research proposal puts forward an experiment to measure the performance level, level of trust and conflict, interaction quality and satisfaction in virtual teams. The manipulation to be performed is the type of software development collaboration system in use by the teams, either CVS or MILOS.

¹<http://www.linux.org>

²<http://www.cvshome.org/>

Literature Review

Face-to-Face and Virtual teams

There has been considerable research into the effects of virtuality over co-location in the study of teams. Andres (2002) found that software development teams in a face-to-face setting had higher productivity and improved interaction quality than those using videoconferencing to collaborate. Chidambaram and Jones (1993) studied dispersed and face-to-face teams and discovered that social presence, communication effectiveness and the communications interface were superior in the face-to-face instances. Domino, Hevner and Collins (2002) propose hypotheses that relate software quality, productivity and developer satisfaction to developer setting. The hypotheses given by Domino et al. (2002) are based on communication effectiveness and media richness theories, and theorise that co-location is superior to dispersed teams in the development of software.

Media richness in communication technologies

Kydd and Ferry (1994, cited in Andres, 2002, pg 41) provide a method to rate the richness of a communications medium, this includes interpersonal interaction, verbal and non-verbal cues, socio-emotional communication and the level of feedback and back-channelling cues. The more able a form of media to transmit these communication characteristics, the higher the level of media richness it holds. Plowman (1995, cited in Domino et al., 2002) and Sillince (1996, cited in Domino et al., 2002) show that limited timing information and modalities (low richness) in a medium is linked with poor communication effectiveness. Karolak (1998) discusses the most common software development communication technologies, along with their content and timeliness ratings. Face-to-face communication is seen as having the greatest content and timeliness, whereas postal mail, voice mail and email are low on content and timeliness (Karolak, 1998).

Trust in virtual teams

Cramton and Webber (1999, cited in Susman, Gray, Blair and Perry, 2002) found evidence that dispersion and limited face-to-face interaction reduced the trust level in teams. Trust is harder to build, and is built differently in virtual teams than it is in face-to-face teams. This is partly due to the differences in communication mechanisms. Kirkman, Rosen, Gibson, Tesluk and McPherson (2002) suggest that rapid response to electronic communication is vital to build this trust in virtual teams. Kirkman et al. (2002) go on to suggest that a team charter should be developed to guide behaviours that foster trust. The rapid response policy is supported by Coppola, Hiltz and Rotter (2001), who found that high levels of trust are maintained with the continuous and frequent interaction of team members in a virtual team. Coppola et al. (2001) found that by establishing early communications among team members, a number of benefits arose. This included the development of a positive social atmosphere, the reinforcement of predictable communications and the involvement of team members in tasks. To overcome limited trust in a virtual team, it may be possible to select certain features in a collaboration system to make communications routine (Susman et al. 2002).

Conflict in virtual teams

Conflict in teams can generate problems. Cox (2003) found that intrapersonal and intragroup conflict are associated with poor work satisfaction levels. In relation to virtual teams, Cramton (2000, cited in Hinds and Bailey, 2000) discovered that geographic dispersion amongst team members increased conflict, due in part because of an imbalance of information at the dispersed sites. Members at one site assumed that the members at the other sites were as knowledgeable as themselves, leading to negative perceptions of the dispersed members due to unanticipated behaviour.

Dispersion in geography is often accompanied with dispersion in time in virtual teams. Barley (1988, cited in Hinds and Bailey, 2000) observed that conflict can occur between two groups that are unable to communicate properly due to temporal asymmetry. Technologists that could not talk to radiologists in one hospital due to the radiologist's unpredictability of work hours sometimes had negative perceptions of them, therefore increasing the level of conflict between the two groups.

Team collaboration systems

By supporting the interaction process between members of a team, it is possible to improve many aspects of the team, including performance and member satisfaction. These supporting systems (or collaboration technologies) are often a mix of traditional communication methods and computer based information systems. Easley, Devaraj and Crant (2003) found that team work quality (as developed by Hoegl and Gemuenden, 2001, to measure a number of team characteristics, including cohesion and communication) is positively associated with the use of a collaborative system. The use of this system also resulted in increased team performance in tasks that were supported by the system.

Majchrzak, Rice, Malhotra, King and Ba (2000) examined a computer-supported virtual team, and found that technology structures supporting the team were eventually modified in response to misalignments among team, organisational and technology structures. This indicates that the use of collaborative systems should not be rigid, otherwise the benefits usually associated with them may not eventuate (Majchrzak et al., 2000).

Ocker, Hiltz, Turoff and Fjermestad (1996) found that when using an asynchronous mode of computer conferencing, the software quality amongst teams of students was greater than those not using the system. The creativity of the software solutions was also found to be considerably improved, due to the greater number of alternative solutions proposed in the conferencing setting (Ocker et al., 1996). In a review of collaboration technologies, Dustdar and Gall (2002) suggests that peer-to-peer software, in combination with traditional server technologies provide an ideal communications solution. Dustdar and Gall (2002) also point to the utility of SMS cellular messaging services for real-time queries of information between team members.

The focus of this research is to test the use an advanced form of software development collaboration technology in a virtual team setting. As described earlier, Maurer and Holz (2002) provide a system named MILOS that integrates knowledge management

and advanced process support in the development of software solutions. As described by Maurer and Holz (2002), process support in MILOS is provided with a number of functions. The first is project coordination, where task assignments, deadlines and project overviews can be viewed, edited or added by team members and managers. Following this is document routing and active notifications, where team members and managers are able to access the workflow engine to complete the tasks they have been assigned and to receive notification when events occur that require attention. Finally, synchronous communication is provided in MILOS, with the integration of Microsoft Netmeeting. Netmeeting allows real-time audio, video and text to be shared between two or more dispersed developers.

Maurer and Holz (2002) go on to detail the knowledge management functionality of MILOS. MILOS maintains an experience base of reusable process descriptions that are commonly used by the organisation's software development process. This base can be added to and contains information about "tasks, potential task decompositions, information flow, and background knowledge" (Maurer and Holz, 2002, pg 6). Therefore, knowledge is captured and maintained, rather than being lost and reinvented, as is often the case in less sophisticated collaborative software support environments.

Conceptual Framework

In general, the literature on virtual teams and collaboration technologies does not look specifically at the software development process. Andres (2002), Ocker et al. (1996) and Domino et al. (2002) do look at the effect of collaboration technologies on teams involved in software development, but do not consider explicitly a collaboration system designed for software development. Rather, these studies focus on the existence or non-existence of a generic collaboration system such as videoconferencing or computer conferencing. A gap therefore exists in the literature in relation to specific research on the usefulness of advanced software development collaboration technologies when used in a virtual team setting.

This research proposes to find the relationship between virtual team performance and the use of two forms of software development collaboration technologies. It is hypothesised that the use of MILOS will improve the performance of virtual teams over those teams using CVS, the traditional collaboration technology used extensively in the industry:

H1: The performance of a virtual software development team using advanced knowledge and process collaboration technologies will be greater than that of a virtual software development team using only basic collaboration technologies.

The performance measure can be broken down into two constructs, IS quality and process efficiency. IS quality is the level of functionality and usability in relation to the original system requirements. Process efficiency examines the time efficiency and process completeness of an IS system. When using MILOS, the quality should be higher than when using CVS due to the improved level of understanding of the software requirements by team members. This is based on the ability of the knowledge management component of MILOS to inform team members, and to make it easier for new team members to receive all of the relevant information. The process efficiency of the virtual team should be greater when using MILOS, due to the advanced process support component. Team members should require less time in dealing with coordination tasks, therefore allowing more time to program lines of code.

H1a: The IS quality of a virtual software development team using advanced knowledge and process collaboration technologies will be higher than that of a virtual software development team using only basic collaboration technologies.

H1b: The process efficiency of a virtual software development team using advanced knowledge and process collaboration technologies will be greater than that of a virtual software development team using only basic collaboration technologies.

Andres (2002) measured team member interaction quality and process satisfaction when comparing face-to-face and virtual software development teams. For interaction quality the measure included the level of perceived negative opinions made and received by members and the level of frustration about other members' behaviour (Andres, 2002). Process satisfaction described the level of satisfaction, fairness and understandability of the task execution method (Andres, 2002).

By using MILOS, the interaction quality should be increased because a richer communications mechanism is being used, as Kydd and Ferry (1994, cited in Andres, 2002), Plowman (1995, cited in Domino et al., 2002) and Sillince (1996 cited in Domino et al., 2002) suggest. Through Netmeeting, MILOS provides synchronous communication with a range of rich media. In comparison, traditional methods using CVS, email and newsgroups are limited to asynchronous text based communications.

H2: The interaction quality of team members in a virtual software development team using advanced knowledge and process collaboration technologies will be higher than that of a virtual software development team using only basic collaboration technologies.

Although Andres (2002) found no significant effect of communication medium on group process satisfaction, it is predicted that by using MILOS, team members will have higher satisfaction levels. MILOS provides knowledge management that allows for the rapid dissemination of relevant information to team members. This should give team members greater confidence in the tasks they carry out, and ultimately impact positively on the quality of the final solution (therefore giving a high level of satisfaction). Teams using CVS may find it more difficult to get up to speed in a new software development due to the limited availability of information and the difficulty in finding relevant information. This may result in a lower level of team member satisfaction:

H3: The level of satisfaction of team members in a virtual software development team using advanced knowledge and process collaboration technologies will be greater than that of a virtual software development team using only basic collaboration technologies.

Barley (1988, cited in Hinds and Bailey, 2000) found that temporal asymmetries lead to conflict in teams. By reducing this asymmetry with improved communications that are internet-based, the use of MILOS should reduce conflict:

H4: The level of conflict in a virtual software development team using advanced knowledge and process collaboration technologies will be lower than that of a virtual software development team using only basic collaboration technologies.

Pearson, Allison, Ensley, Michael, Amason and Allen (2002) provide two constructs to measure conflict, relationship and task conflict. Therefore, hypothesis four is broken down into two parts:

H4a: The level of task conflict in a virtual software development team using advanced knowledge and process collaboration technologies will be lower than that of a virtual software development team using only basic collaboration technologies.

H4b: The level of relationship conflict in a virtual software development team using advanced knowledge and process collaboration technologies will be lower than that of a virtual software development team using only basic collaboration technologies.

Kirkman et al. (2002) and Coppola et al. (2001) note the importance of early and frequent communications in a virtual setting to maintain a high level of trust. The use

of MILOS provides greater incentive for communications due to the increased richness and synchronicity provided, and this may result in a higher level of team member trust:

H5: The level of trust in a virtual software development team using advanced knowledge and process collaboration technologies will be greater than that of a virtual software development team using only basic collaboration technologies.

Research methodology

Research design

Each of the dependent variables (information systems quality, process efficiency, task conflict, relationship conflict, satisfaction and trust) will be analysed separately by using a one-way analysis of variance (ANOVA) for each dependent variable. ANOVA compares the variability within the samples to the variability between the samples (College of Sciences, Institute of Technology and Engineering, 2000). If the means of the samples differ significantly from each other, then the independent variable (collaboration technology) can be said to have an effect on the dependent variable under examination. This will allow the hypotheses to be tested with an F test. A one way ANOVA is chosen because only one factor is to be varied in this study, the factor of collaboration technology. Table 1 below shows the ANOVA table that will be generated for each dependent variable.

Table 1: One-way ANOVA table for each dependent variable (taken from College of Sciences, Institute of Technology and Engineering, 2000, pg 33)

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square
Between group	SSB	K - 1	MSB = SSB/(K-1)
Within group	SSE	n - K	MSE = SSE/(n-K)
Total	SST	n - 1	
			F = MSB/MSE

SSB = sum of squares between

SSE = sum of squares within

SST = total sum of squares

K = number of populations (2 in this study)

n = number of samples (40 in this study)

Kozlowski and Klein (2000, cited in Colquitt, Noe and Jackson, 2002) recommend that researchers explicitly define the level of theory that they are working at. This research is targeted at the team level, but gathers most of its data from the individual team member. This data will be aggregated to the team level.

Subjects

The subjects that will participate in this experiment will be undergraduate computer science and computer systems engineering students at Massey University in Palmerston North, New Zealand. These students will be enrolled in a 2nd year object-oriented programming paper. Approximately 200 students will participate in the experiment. Each student will be placed into a team with four other members, giving a total of 40 teams. The students will be placed randomly into teams.

Experimental manipulation

20 teams will make use of the MILOS collaboration technology, and 20 will use CVS with email and newsgroup support. In the process of completing the experimental

task, the team members will only collaborate with each other through the assigned collaboration technology. Source code, documentation, and executable files will be stored and transferred either by MILOS or CVS. Training of both systems will be provided to the students in the first week of the semester.

Experimental task

All of the teams will be required to design and implement an object-oriented online registration system for university admissions. The programming language that the teams will use is C++, along with the use of an object-oriented software development lifecycle. The task requires that the teams provide a completed system that conforms to a set of requirements initially provided. These requirements have been chosen to allow the students to be able to complete the task successfully given their limited understanding of the subject matter. However, the task provides a real-world type problem, and will allow some generalisation of the results of the experiment into the software development industry.

The experiment will last the entire semester (13 weeks). Students will receive credit based on the quality of the final product which will be expertly assessed. Therefore the students have motivation to do well. Teams will have access to computing facilities on campus. MILOS and CVS will be installed on these computers, but students will be encouraged to use only the collaboration technologies that their team have been assigned to use. If a student wishes to work from a home computer, the required instructions and software will be provided. While in the labs or in other situations, students will be encouraged not to collaborate in a face-to-face manner with other participants on the experimental task.

The task is the same for all participants in the experiment. A reasonable amount of control exists in setting the parameters of the experiment which allows the level of complexity to be managed.

Dependent measures

Each team member will be asked to complete a questionnaire once the experimental task has been completed. In addition to this, the assessment graders (expert judges) will be asked to complete a questionnaire on the information system quality provided by each team. The dependent measures are broken into performance related team outcomes and psychosocial outcomes. Detailed construct questionnaire measures can be found in the appendix. Performance related outcomes include the quality of the information system and the efficiency of the software development process, as developed by Ravichandran and Rai (2000). The quality of the system elicits perceptions of functional completeness, usability and overall quality from the end users of the system (in this study the user is an expert judge). Process efficiency measures perceptions of schedule adherence and process correctness from the developers (the students taking part). Table 1 gives the validity results below for these constructs. All values are above the recommended thresholds, except the convergent validity of information system quality (however 0.87 is close to the 0.90 threshold).

Table 2: Validity of quality and efficiency constructs (adapted from Ravichandran and Rai, 2000, pg 136)

Construct	Unidimensionality: Goodness of fit	Reliability: Cronbach's alpha	Reliability: Werts Linn Jorsekog P _c	Convergent validity: Bentler- Bonnet
Information system quality	0.90	0.82	0.83	0.87
Process efficiency	0.96	0.78	0.79	0.95

The level of intragroup conflict in the virtual team is measured with the questionnaire items relationship conflict and task conflict as developed by Pearson et al. (2002). Pearson et al. (2002) calculated the goodness of fit index to be 0.91 for these items indicating high unidimensionality.

Levels of group process satisfaction and team member interaction quality are to be measured using questionnaire items developed by Andres (2002). Interaction quality elicits perceived team member behaviour, whereas group process satisfaction measures perceptions of fairness, understandability and overall satisfaction of the development process. Andres (2002) provides the reliabilities for these two measures, with interaction quality and process satisfaction having Cronbach's alpha levels of 0.65 and 0.67 respectively. These levels are very close to the threshold of 0.70, but caution will need to be taken in analysing the results from these two constructs.

Finally, the level of trust in the virtual team is examined using the questionnaire items used by Jarvenpaa and Leidner (1998). The items are based on the work of Pearce, Sommer, Morris and Frideger (1992, cited in Jarvenpaa and Leidner, 1998). The Cronbach's alpha level for these items is 0.92, indicating a high level of reliability (Jarvenpaa and Leidner, 1998).

Results

Table 3 below is a dummy table of how the results from the experiment will be presented. The F-statistic at the required significance value will be used to determine if a hypothesis can be accepted or rejected. Each hypothesis will be examined in turn, with comments made to the significance of the effect of collaboration system on each dependent variable.

Table 3: Collaboration system ANOVA results (dummy table)

Dependent variable	Degrees of freedom	F-statistic	Mean square	Significance level	Observed power
Team trust					
Relationship conflict					
Task conflict					
Interaction quality					
Group process satisfaction					
Information system quality					
Process efficiency					

Discussion

As the process of globalisation continues, the power of computing technology multiplies, and the speed and availability of the internet increases, the use of virtual teams and virtual organisations in all industries will also increase. There is a need to access the highly developed but narrow skill sets of individuals in complex industries. Breaking down traditional barriers at the organisational and national level allows the project manager access to a wider labour pool.

The use of an advanced software development collaborative technology gives managers and project leaders the possibility of achieving high levels of trust, low levels of conflict, high satisfaction, good interaction quality and high performance in a software development team. The answers to the proposed research questions will aid in the understanding of how collaborative technologies designed specifically for the software industry impact these variables. MILOS is an emerging piece of software. This research may illuminate ways to improve it and others like it, towards a comprehensive solution to the many difficulties faced by virtual software development teams.

A limitation of the proposed research is that it is very specific to the software industry. Caution needs to be made when generalising to teams not involved in software development. Collaboration systems involving knowledge management and process support are not however limited to the software industry. The basic concept behind MILOS is widely applicable, especially where there is a large amount of knowledge in an organisation or industry that needs to be maintained and distributed to team members.

References

- Andres, H. P. (2002). A comparison of face-to-face and virtual software development teams. *Team Performance Management*, 8, 2, 39-48.
- Cascio, W. F. (2000). Managing a virtual workplace. *Academy of Management Executive*, 14, 3, 81-91.
- Chidambaram, L. & Jones, B. (1993). Impact of communication medium and computer support on group perceptions and performance: a comparison of face-to-face and dispersed meetings. *MIS Quarterly*, 17, 4, 465-491.
- College of Sciences, Institute of Technology and Engineering. (2000). *143.340 Industrial research techniques: multivariate analysis*. Palmerston North: Massey University.
- Colquitt, J. A., Noe, R. A. & Jackson, C. L. (2002). Justice in teams: antecedents and consequences of procedural justice climate. *Personnel Psychology*, 55, 83-109.
- Coppola, N. W., Hiltz, S. R. & Rotter, N. (2001). Building trust in virtual teams. *Proceedings of IEEE Professional Communication Society International Professional Communication Conference, 2001*. pp. 353-366.
- Cox, K. B. (2003). The effects of intrapersonal, intragroup, and intergroup conflict on team performance effectiveness and work satisfaction. *Nursing Administration Quarterly*, 27, 2, 153-163.
- Domino, M., Hevner, A. R. & Collins, R. W. (2002). Applying agile software development processes to global virtual teams: a study of communication modalities. *Proceedings of the ACM SIGCPR Conference, 2002*. pp. 76-78.
- Dustdar, S. & Gall, H. (2002). Process awareness for distributed software development in virtual teams. *Proceedings of the 28th Euromicro Conference*, 244-251.
- Easley, R. F., Devaraj, S. & Crant J. M. (2003). Relating collaborative technology use to teamwork quality and performance: an empirical analysis. *Journal of Management Information Systems*, 19, 4, 247-268.
- Eom, S. B. & Lee, C. K. (1999). Virtual teams: an information age opportunity for mobilizing hidden manpower. *SAM Advanced Management Journal*, 64, 2, 12-16.
- Hinds, P. J. & Bailey, D. E. (2000). Virtual teams: anticipating the impact of virtuality on team process and performance. *Paper presented at the Academy of Management annual meeting, August, Toronto*.

- Hoegl, M. & Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: a theoretical concept and empirical evidence. *Organizational Science*, 12, 4, 435-449.
- Jarvenpaa, S. L. & Leidner, D. E. (1998). Communication and trust in global virtual teams. *Journal of Computer Mediated Communication*, 3, 4, June.
- Karolak, D. W. (1998). *Global software development: managing virtual teams and environments*. IEEE Computer Society.
- Kirkman, B. L., Rosen, B., Gibson, C. B., Tesluk, P. E. & McPherson, S. O. (2002). Five challenges to virtual team success: lessons from Sabre, Inc. *Academy of Management Executive*, 16, 3, 67-72.
- Majchrzak, A., Rice, R. E., Malhotra, A., King, N. & Ba, S. (2000). Technology adaption: the case of a computer-supported inter-organizational virtual team. *MIS Quarterly*, 24, 4, 569-600.
- Maurer, F. & Holz, H. (2002). Integrating process support and knowledge management for virtual software development teams. *Annals of Software Engineering, Special Volume on Process-Based Software Engineering*, 14, 145-168. Kluwer Academic Publishers.
- Ocker, R., Hiltz, S. R., Turoff, M. & Fjermestad, J. (1996). The effects of distributed group support and process structuring on software requirements development teams: results on creativity and quality. *Journal of Management Information Systems*, 12, 3, 127-153.
- Pearson, A. W., Ensley, M. D. & Amason, A. C. (2002). An assessment and refinement of Jehn's intragroup conflict scale. *International Journal of Conflict Management*, 13, 2, 110-127.
- Ravichandran, T. & Rai, A. (2000). Total quality management in information systems development: key constructs and relationships. *Journal of Management Information Systems*, 16, 3, 119-155.
- Susman, G. I., Gray, B. L., Blair, C. E. & Perry, J. (2002). Recognition and reconciliation of differences in interpretation of misalignments when collaborative technologies are introduced into new product development teams. *Proceedings of the Academy of Management Meeting, Denver, CO*.

Appendix

Measures of team trust

These measurement items use a five-point scale (1 – 5) and are taken from Jarvenpaa and Leidner (1998).

Members of my work group show a great deal of integrity.
I can rely on those with whom I work in this group.
Overall, the people in my group are very trustworthy.
We are usually considerate of one another's feelings in this work group.
The people in my group are friendly.
There is no "team spirit" in my group.
There is a noticeable lack of confidence among those with whom I work.
We have confidence in one another in this group.

Measures of intragroup conflict

The measures below make use of an eight-point scale (1 – 8).

Relationship conflict (taken from Pearson et al., 2002, pg 113).

How much emotional conflict was there among the members of your group?
How much anger was there among the members of the group?
How much personal friction was there in the group during decisions?
How much were personality clashes between members of the group evident?
How much tension was there in the group during decisions?

Task conflict (taken from Pearson et al., 2002, pg 113).

How much disagreement was there among the members of your group over their opinions?
How many disagreements over different ideas were there?
How many differences about the content of decisions did the group have to work through?
How many differences of opinion were there within the group?

Measures of team member interaction quality

The following measures are taken from Andres (2002, pg 48) and are rated using a five-point scale (1 – 5).

When working on this project, to what extent did you:

- Feel frustrated or tense about the other team members' behaviour;
- Express negative opinions about any project team member's behaviour;
- Observe others express a negative opinion about your behaviour.

Measures of group process satisfaction

The following measures are taken from Andres (2002, pg 48) and are rated using a five-step semantic differential scale.

How would you describe your team's software development process?

- Fair ... unfair;
- Confusing ... understandable;
- Satisfying ... unsatisfying.

Measures of information system quality

The following measures are taken from Ravichandran and Rai (2000, pg 154) and are rated using a seven-point scale (1 – 7).

Users perceive that the system meets intended functional requirements.

The information provided by the system meets user expectations.

The system meets user expectations with respect to response time, flexibility and ease of use.

Users are satisfied with the overall quality of the system.

Measures of process efficiency

The following measures are adapted from Ravichandran and Rai (2000, pg 154) and are rated using a seven-point scale (1 – 7) with items being reverse coded. Changes made have been to focus the items on a single project, rather than a continuous development process.

This project came in over schedule.

Fixing bugs and other types of rework account for a significant proportion of systems development effort.