Groupware Support for Software Inspections: The Impact of Group Interaction and Interface on Performance

Craig K. Tyran
Oregon State University, tyran@bus.orst.edu

Groupware Support for Software Inspections: The Impact of Group Interaction and Interface on Performance

Craig K. Tyran, Dept. of Accounting, Finance and Information Management, College of Business, Oregon State University, tyran@bus.orst.edu

Abstract

The software inspection meeting is one of the best-known techniques for quality assurance in software development and has become a standard practice in many software development groups (Ebenau and Strauss, 1994). The application of groupware has been suggested as a particularly promising way to improve the inspection process (Johnson, 1998). This paper discusses a "research in progress" study that concerns the application of groupware to software inspection meetings. A controlled experimental study involving eighty teams is presently being conducted to address two fundamental research questions relating to the use of groupware to support software inspections: 1) Is it more effective to use groupware in an interactive or nominal group mode?; and 2) Is it worthwhile to incorporate task structure into the design of the groupware interface? The research design is a 2 X 2 factorial design using the inspection team as the unit of analysis. One independent variable manipulates the type of group interaction (nominal vs. interactive) and the second independent variable manipulates task structure (a partitioned groupware interface vs. an unpartitioned interface). The dependent variable will be team performance.

Introduction

The software inspection meeting is one of the best-known techniques for quality assurance in software development and has become a standard practice in many software development groups (Ebenau and Strauss, 1994). The primary purpose of an inspection meeting is to detect the defects in a work product (e.g., program code, system specification). Recent studies have questioned the value of the inspection meeting, as face-to-face inspection meetings have been found to be no more effective at finding defects than "nominal" groups comprised of inspectors working alone without interaction with others (Porter, et al., 1996). Based on the findings from these studies, practitioners and researchers have called for the development of innovative alternatives to the traditional face-to-face inspection meeting (e.g., Glass (1999)). The application of groupware has been suggested as a particularly promising way to improve the inspection process (Johnson, 1998). While the potential for using groupware to support inspections appears bright, relatively little research has been done in this area. To aid in the development and refinement of groupware applications for software inspections, it is important to gain more insight into how to best use groupware to enhance the performance of inspection teams. The purpose of this study will be to explore two fundamental research questions relating to the use of groupware for inspections: 1) Is it more effective to use groupware in an interactive or nominal group mode?; and 2) Is it worthwhile to incorporate task structure into the design of the groupware interface?

Literature review

As noted earlier, research by Porter, et al. (1996) indicates that face-to-face meetings may not be the most effective way to conduct inspections. This finding is consistent with past empirical research in the field of group behavior that has found that nominal groups often perform better than face-to-face groups for tasks such as brainstorming. Studies suggest that one explanation for this is that the desired process "gains" (e.g., synergy) of a meeting are counterbalanced by undesirable meeting process "losses" associated with domination, production blocking, and evaluation apprehension (e.g., Diehl and Stroebe, 1987). Recently, researchers have explored the application of groupware to the inspection task as a means to improve group performance by reducing process losses, while maintaining process gains. The early results have been favorable, as studies in the laboratory (Tyran and George, 1999) and the field (van Genuchten, et al., 1997-98) suggest that groupware supported inspection meetings can be more effective than face-to-face meetings.

However, several questions regarding the application of groupware to software inspections remain unanswered. One of the key questions centers on whether groupware inspections should be conducted in an "interactive" mode in which team members communicate simultaneously using the electronic channel of the groupware, or whether groupware inspections may be more effectively conducted in a nominal mode, where the group members work alone and do not interact with other members. Groupware theory suggests that an interactive mode may have the potential to be more fruitful due to efficiency advantages and stimulation advantages (e.g., Pinsonneault, et al., 1999). This question has been explored for brainstorming types of tasks. While the laboratory findings for brainstorming have been somewhat mixed (e.g., Gallupe, et al., 1991, Pinsonneault, et al., 1999, Valacich, et al., 1994), this study hypothesizes that the efficiency and stimulation advantages of groupware will result in enhanced inspection team performance. Hence, the first hypothesis of this study is:

> *Hypothesis 1:* Interactive inspection teams using groupware will perform better than nominal inspection teams.

Another important question regarding the application of groupware to inspections concerns how the task is structured. In this context, "task structure," relates to the way that a problem is decomposed. Empirical evidence from individual problem solving research suggests that individuals presented with an all-encompassing task tend to focus on a small subset of the potential solution space and often miss important aspects of the problem (e.g., Connolly, et al., 1993). One way to address this type of problem is to structure a task so that the individual is required to focus on a broader set of issues (Armstrong, et al., 1975). So far, no research has explored the impact of task structure on the inspection task. However, two groupware studies have explored the issue of using task structure to promote the performance of brainstorming groups. In each study, task structure (which was provided by partitioning a brainstorming problem into smaller pieces) was found to have a significant beneficial impact on group performance (Dennis, et al., 1996; Dennis, et al., 1999). Based on the favorable impact of task structure in these studies it is hypothesized that providing a task structure mechanism that partitions the solution space of the inspection task will yield performance benefits. Hence, the second hypothesis of this study is:

> *Hypothesis 2:* Inspection teams that use a task structuring mechanism will perform better than teams that do not use task structure.

Research method

The objective of the research study will be to evaluate the impact of group interaction and task structure on the performance of software inspection teams using groupware support. A controlled experimental study is presently being conducted to address the research questions. The research design is a 2 X 2 factorial design using the inspection team as the unit of analysis. One independent variable manipulates the type of group interaction (nominal vs. interactive) and the second independent variable manipulates task structure (a partitioned groupware interface vs. an unpartitioned interface). The dependent variable will be team defect detection performance. Following is a brief discussion of the research subjects, task, procedure and measures.

- **Subjects**: Approximately two hundred and fifty undergraduate students majoring in information systems will participate in the study. Subjects will be randomly assigned into teams of three people each, resulting in approximately eighty teams. Teams will be randomly assigned to one of the four treatment groups. Team sizes of three people will be used based on recommendations regarding the optimal size of software inspection teams (Ebenau and Strauss, 1994).

- **Task**: The task for this study will require the subjects to identify defects existing within a software specification document. This document has been pretested in an earlier study (Tyran and George, 1999). The specification document describes the user requirements for a subsystem of a fictional real estate company and includes three subsections: a two-page long descriptive narrative section, a data dictionary, and three data flow diagrams (DFDs). The document is planted with a variety of defects. A "master list" of the defects existing in the specification has been prepared to assess the performance of the inspection teams.

- **Procedure:** Prior to the experiment, all participants in the study will be provided with a reading and an in-class overview regarding the objectives and procedures of the software inspection meeting process. Additionally, upon reporting to the experimental site, the subjects will be given a brief scripted oral review of software inspection principles to ensure that all are familiar with the inspection procedures. The inspection procedures to be followed for this study will be based on the industry standards of the Fagan's inspection approach (Fagan, 1976) and the IEEE standard (1989). Each participant will be provided with a checklist of the generic types of defects found in a software specification document. Specific procedures for each experimental treatment are described below:

  Group Interaction Variable (Nominal vs. Interactive Groups): The participants in the nominal treatment will use the groupware interface to record all defects that they identify into their own personal groupware window. Participants in the nominal groups will only be able to view their own findings and will not be able to see the contributions made by others in their group. The participants in the interactive groups, on the other hand, will share a common groupware window and will be able to view the findings submitted by their teammates. For each of these treatment groups, half of the teams will have a computer interface offering minimal task structure, and half will have an interface offering a basic form of task structure (see below).

Task Structure Variable (Partitioned vs. Unpartitioned Interface): In this study, task structure will be manipulated by means of the groupware interface. Half of the subjects will be provided with task structure consisting of an interface that explicitly partitions the inspection task into subparts based on the four different categories of defects defined in the checklist (e.g., completeness, consistency). The other half of the participants will be provided with a groupware window that is unpartitioned. No checklist categories are included in the groupware window.

- **Measures**: The dependent variable for the study will be group defect detection performance. Group performance will be measured by counting the number of distinct defects identified correctly by the members of a group. Defects identified by more than one member of the team will only be counted once. The master list of defects for the specification document will be used as the "answer key." To supplement the performance data, each subject's perceptions regarding the inspection exercise will be measured using a questionnaire. The questionnaire items and scales will be aimed at assessing various aspects of the subject's perceptions of the group process including confound checks and issues related to learning and the effectiveness of the inspection process.

Conclusion

This study will build on previous leadership research by extending our understanding of the application of groupware for the process of software inspection meetings. This research is expected to have implications for both researchers and practitioners. In addition to addressing the research questions discussed above, the study may identify fruitful areas for future research. Also, findings from this study may help to inform practitioners who wish to improve the effectiveness of their software inspection teams.

References

Armstrong, J.S., Denniston, W.B. and Gordon, M.M. "The Use of the Decomposition Principle in Making Judgments," *Organizational Behavior and Human Performance* (14:2), 1975, pp. 257-263.

Connolly, T.R., Routhieaux, R.L., and Schneider, S.K. "On the Effectiveness of Group Brainstorming: Test of One Underlying Cognitive Mechanism," *Small Group Research* (24:4), 1993, pp. 490-503.

Dennis, A.R., Valacich, J.S., Connolly, T., and Wynne, B.E. "Process Structuring in Electronic Brainstorming," *Information Systems Research* (7:2), 1996, pp. 268-277.

Dennis, A.R., Aronson, J.E., Heninger, W.G., and Walker, E.D. "Structuring Time and Task in Electronic Brainstorming," *MIS Quarterly* (23:1), 1999, pp. 95-108.

Diehl, M. and Stroebe, W., "Productivity Loss in Brainstorming Groups: Toward the Solution of a Riddle," *Journal of Personality and Social Psychology* (53:3), 1987, pp. 497-509.

Ebenau, R.G. and Strauss, S.H. Software Inspection Process. McGraw-Hill, New York, NY, 1994.

Fagan, M.E. "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems Journal* (15:3), 1976, pp. 182-211.

Gallupe, R.B., Dennis, A.R., Cooper, W.H., Valacich, J.S., Bastianutti, L.M., and Nunamaker, J.F. "Electronic Brainstorming and Group Size," *Academy of Management Journal* (35:2), 1992, pp. 350-369.

Glass, R.L., "Inspections – Some Interesting Findings," *Communications of the ACM* (42:4), 1999, pp. 17-19.

IEEE Computer Society, IEEE Standard for Software Reviews and Audits (IEEE Std. 1028-1988). IEEE, New York, NY, 1989.

Johnson, P.M, "Reengineering Inspection," *Communications of the ACM* (41:2), 1998, pp. 49-52.

Pinsonneault, A., Barki, H., Gallupe, R.B., and Hoppen, N. "Electronic Brainstorming: The Illusion of Productivity," *Information Systems Research* (10:2), 1999, pp. 110-133.

Porter, A.A., Siy, H., and Votta, L.G., "A Review of Software Inspections," Advances in Computers (42), Academic Press, 1996, pp. 39-76.

Tyran, C.K. and George, J.F. "Augmenting Software Inspection Meetings with Group Support Technology," Working Paper, Oregon State University, 2000.

Valacich, J.S., Dennis, A.R., Connolly, T.R. "Idea Generation in Computer-Based Groups: A New Ending to an Old Story," *Organizational Behavior and Human Decision Process* (57:3), pp. 448-467.

van Genuchten, M., Cornelissen, W., and van Dijk, C. "Supporting Inspections with an Electronic Meeting System", *Journal of Management Information Systems* (14:3), 1997-98, pp.165-178.