A Controlled Experiment on the Effects of Synchronicity in Remote Inspection Meetings

Fabio Calefato, Filippo Lanubile, Teresa Mallardo
University of Bari, Italy, Dipartimento di Informatica
{calefato, lanubile, mallardo}@di.uniba.it

Abstract

Traditionally, software inspection has largely relied on collocated interaction of inspectors. As companies have begun to turn to distributed software development, meeting in a room has become impractical. In this paper we report on controlled experiment to assess the effect of synchronous and asynchronous communication in remote inspection meetings.

1. Introduction

Software inspection is a type of formal peer review for the static verification of software artifacts, including source code and documents. Traditionally, it has largely relied on collocated interaction, by getting inspectors together in a meeting room [2], although this represented a serious bottleneck in the software development process [10]. In addition, over the last years collocated meetings have become problematic due to geographical distance, as companies turn to distributed software development as an organizational model. A number of studies have been reported on assessing alternatives to the face-to-face inspection meetings [8, 4].

IBIS (Internet-Based Inspection System) is a web-based tool that aims to support geographically dispersed inspection teams [6], on the basis of a reengineered inspection process [9]. In IBIS inspectors individually take note of defects (discovery stage), which are then put together to remove duplicates (collection stage). Finally, inspectors meet and discuss on all collated defects identified, so as to accept true defects and remove false positives (discrimination stage). A previous controlled experiment has shown that asynchronous discussions within distributed software inspections can replace face-to-face meetings for the purpose of discriminating between true defects and false positives [5]. However, while asynchronous discrimination meetings ensure that inspectors commit to the task at convenience, they have the flipside of requiring a longer time to be completed, as compared to synchronous meetings. Thus, even adopting the reengineered inspection process, asynchronous inspection meetings may represent the same bottleneck problem encountered in classic collocated inspection process.

The theory of Media Synchronicity [1], a prominent theory on computer-mediated communication, suggests that this occurs because synchronous communication better supports the process of convergence, i.e., the development of shared views and understanding between communication participants. Indeed, the goal of the discrimination stage is that inspectors reach agreements (i.e., converge) on both the true defects to be accepted and the false positives to be removed.

The goal of this paper is to empirically investigate the use of synchronous and asynchronous text-based communication in remote inspection meetings. Thus, according to the theory reported above, we defined the following research question.

RQ: *What is the impact of synchronicity for achieving a mutual agreement between participants in remote inspection meetings?*

Because of the varied nature of the software artifacts, we also want to assess the effects of discriminating the defects found in both requirements and design documents.

2. The Experiment

The study involved thirty-six graduate students in computer science, attending a web engineering course at the University of Bari, two researchers, and four research assistants. As a final course assignment, students were required to work in groups of three to five people and develop a web application, including both the requirements and design document. Students formed 11 developer groups. Each inspection team was composed of the authors of the inspected artifact (i.e., the developer group who created the artifact), the inspection leader (i.e., one of the two researchers), the inspection expert (i.e., one of the four research assistants), and the domain expert(s) (i.e., one or two students from another developer group). Each of the two artifacts produced by developer groups was submitted to an inspection process. Thus, we performed 22 distributed inspections.

The distributed inspections process was entirely supported by the IBIS tool. To simulate the geographical dispersion of the inspection teams, students were allowed to use the tool from home, as well as from the laboratories in our department. Firstly, inspectors conducted the discovery stage in parallel and then the inspection leader performed the collection stage. In the discrimination stage the entire inspection team interacted in a meeting performed either synchronously or asynchronously. During an asynchronous meeting each defect was mapped to a threaded discussion and participants had maximum three days to add messages and vote by rating any potential defect as true defect or false positive. Instead a synchronous meeting took place through a 2-hour long structured chat.

2.1. Design

The independent variables are the following:

- Interaction type: asynchronous vs. synchronous interaction.
- Software artifact: requirements document vs. design document.

Table 1 shows the experimental design which corresponds to a 2x2 factorial design with 22 different inspection teams. We tested the following null hypotheses:

H_{01}: No interaction between Interaction type and Software artifact.
H_{02}: No main effect for Interaction type
H_{03}: No main effect for Software artifact.

Table 1. Experimental Design

		Software artifact	
		requirements	Design
Interaction type	async	5	6
	sync	5	6

2.2. Dependent variables

Meeting Effectiveness

Because the goal of the inspection meeting is to reach mutual agreement in discriminating between false positives and true defects, we defined the construct of the meeting effectiveness using the following dependent variables:

- Collated defects: the number of defects merged from individual findings to be discussed during the meeting.
- True defects: the number of defects for which consensus was reached during the meeting in considering them as true defects.
- Removed false positives: the number of defects for which consensus was reached d during the meeting in considering them as not true defects, thus as false positives.
- Agreements: the sum of true defects and removed false positives.
- % of agreements: the ratio of agreements to collated defects.

Meeting Efficiency

The meeting efficiency was defined considering the following variables:

- Elapsed time (in hour): the time elapsed from the first and the last message exchanged during the meeting (both asynchronous and synchronous).
- Agreements per hour: the ratio of agreements to elapsed time.

2.3. Data Analysis

We performed a two-way ANOVA, with two between-groups factors (i.e., *interaction type* and the *software artifact*) on the two dependent variables: *% of agreements* and *agreements per hour*.

With regard the meeting effectiveness, the analysis failed to reveal significant interaction between the two independent variables ($F=0.17$; $p=0.68$). While the main effect for Interaction type was not significant ($F=0.13$; $p=0.72$), the main effect for Software artifact was significant ($F=6.62$; $p=0.01$). Then subjects who discussed design defects were more effective than subjects who discussed requirements defects, as shown in Figure 1.

As regards meeting efficiency, the analysis also failed to reveal a significant interaction ($F=0.39$; $p=0.53$). The main effect for Software artifact was not significant ($F=0.28$; $p=0.59$), whereas the main effect for Interaction type was significant ($F=21.74$; $p=0.0001$). Thus, as shown in Figure 2, subjects who synchronously discussed spent fewer hours than subjects who asynchronously discussed to reach mutual agreement on requirements and design defects, although they achieved the same level of the effectiveness.

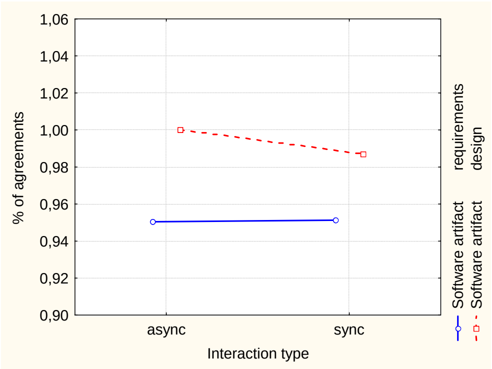

Figure 1. Significant main effect for Software artifact

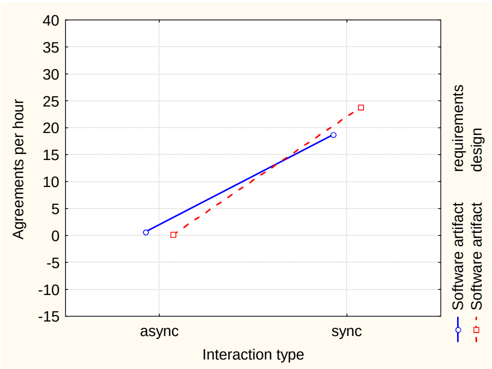

Figure 2. Significant main effect for Interaction type

3. Conclusions

In this paper we have presented an experiment on the effects of synchronous and asynchronous interaction to reach mutual agreements in remote inspection meetings. Results have shown that synchronous inspection meetings are more efficient than asynchronous meetings, while being equal in terms of effectiveness. Hence, this finding suggests that in distributed context, when business hours overlap, synchronous meetings can be preferred to shorten the overall length of the inspection process. In addition, results show that the type of artifacts to inspect (i.e., requirements vs. design documents) makes a difference on the effectiveness of such meetings, as a larger number of mutual agreements were reached by participants in the inspections of design documents. This can be explained because the higher the abstraction level of the artifact, the more difficult to express what a fault consists of [7].

Our conclusions are subjected to some threats to validity. In particular, threats of external validity exist because of the academic laboratory setting of the experiment and the simulated geographical dispersion of the subjects. A threat to internal validity is caused by the limited size of the experimental data, subjected to parametric statistics. Final, because of the different time limit for meetings, a construct validity threat exists on the efficiency construct, i.e., the number of agreements per hour.

These threats notwithstanding, both theory and practice confirm our conclusions. The theory of Media Synchronicity posits that synchronous communication is better suited to support tasks of decision making, such as inspection meetings, where the convergence of participants is needed. In addition, several open-source software projects, like Gnome [3], other than relying upon asynchronous communication tools (e.g., email and web forums), use synchronous text-based chat sessions for bug triage of issue repositories and rapid decision-making about project evolution.

References

[1] A.R. Dennis, J.S. Valacich, "Rethinking Media Richness: Towards a Theory of Media Synchronicity", *Proc. Hawaii Int'l Conf. on System Sciences (HICSS-32)*, Vol. 1, 1999.

[2] M.E. Fagan, "Design and Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, Vol. 15, No. 3, pp. 182-211, 1976.

[3] GNOME http://live.gnome.org/Bugsquad/BugDays

[4] P. Grunbacher, M. Halling, S. Biffl, "An Empirical Study on Groupware Support for Software Inspection Meetings", *Proc. Int'l Conf. on Automated Software Engineering (ASE'03)*, Montreal, Canada, pp. 4-11, 2003.

[5] F. Lanubile, T. Mallardo, "A Preliminary Study on Asynchronous Discussions for Distributed Software Inspections", *Proc. Workshop on Cooperative Support for Distributed Software Engineering Processes (CSSE'04)*, Linz, Austria, 2004.

[6] F. Lanubile, T. Mallardo, F. Calefato, "Tool Support for Geographically Dispersed Inspection Teams", *Software Process: Improvement and Practice*, Vol. 8, No. 4, Wiley InterScience, pp. 217-231, 2003.

[7] F. Lanubile, F. Shull, V. Basili, "Experimenting with error abstraction in requirements documents", *Proc. Int'l Symposium on Software Metrics (METRICS'98)*, Bethesda, Maryland, pp.114-121, 1998.

[8] D.E. Perry, A. Porter, M.W. Wade, L.G. Votta, and J. Perpich, "Reducing Inspection Interval in Large-Scale Software Development", *IEEE Transaction on Software Engineering*, Vol. 28, No. 7, pp. 695-705, 2002.

[9] C. Sauer, D.R. Jeffery, L. Land, and P. Yetton, "The effectiveness of software development technical reviews: A behaviorally motivated program of research", *IEEE Transactions on Software Engineering*, Vol. 26, No. 1, pp. 1-14, 2000.

[10] L.G. Votta, "Does Every Inspection Need a Meeting?" *ACM Software Engineering Notes*, Vol. 18, No. 5, pp. 107-114, 1993.