Assessing software review meetings: A controlled experimental study using CSRS

Philip M. Johnson*
Danu Tjahjono
Department of Information and Computer Sciences
University of Hawaii
Honolulu, HI, 96822 USA
johnson@hawaii.edu

ABSTRACT
Software review is a fundamental component of the software quality assurance process, yet significant controversies exist concerning efficiency and effectiveness of various review methods. A central question surrounds the use of meetings: traditional review practice views them as essential, while more recent findings question their utility.

We conducted a controlled experimental study to assess several measures of cost and effectiveness for a meeting and non-meeting-based review method. The experiment used CSRS, a computer mediated collaborative software review environment, and 24 three person groups. Some of the data we collected included: the numbers of defects discovered, the effort required, the presence of synergy in the meeting-based groups, the occurrence of false positives in the non-meeting-based groups, and qualitative questionnaire responses.

This paper presents the motivation for this experiment, its design and implementation, our empirical findings, conclusions, and future directions.

Keywords
Formal technical review, inspection, experimental study, CSRS.

INTRODUCTION
Formal technical review is an umbrella term for a variety of structured group processes designed to assess and improve the quality of a software work product. While the value of formal technical review (and its most popular form, inspection) to software quality improvement is undisputed, debate about the most effective review procedure is increasing. Such controversy is recent; until a few years ago, structured group review of software work products was virtually equated with the inspection method invented by Michael Fagan [3, 4]. As the potential of formal technical review has become better understood, a plethora of alternatives have been proposed by researchers and practitioners.

Perhaps the most fundamental procedural constant of Fagan inspection and its many variants is the review meeting, where the review team, after some preparation, discusses the work product in a face-to-face manner and notes as many defects as possible. Review meetings are often considered essential to the effectiveness of formal technical review, primarily because they make possible a "synergy" among the review team that can lead to the discovery of defects not found by the participants working individually. Fagan refers to this as the "Phantom Inspector", and some review forms actually provide a checkbox to indicate whether or not the "Phantom" attended the meeting. Other reasons for holding a review meeting include education, clarification, and an imposed deadline [6].

However, meetings are the most costly component of an already costly process which has been shown to add 15-20% new overhead onto development [12]. Meetings are costly because they require the simultaneous attendance of all team members, and their effectiveness requires satisfaction of many conditions, including adequate preparation, efficient moderation, readiness of the work product for review, and cooperation among group members. Furthermore, simply scheduling a time for a meeting of the review team has been shown to lengthen the start-to-finish time for inspection by almost a third in one development group [14]. Presumably, long inspection intervals lead to longer overall development intervals, with potentially enormous costs when time-to-market for a product is a critical factor.

As a result, some researchers and practitioners have proposed a fundamental change to the formal technical review process: the radical restructuring or elimination of meetings altogether. The proponents of this position claim that the benefits of meetings have been exaggerated and that alternatives (such as two person "depositions") are more cost-effective. Although the evidence presented is substantial, far more research is required to truly understand the implications of such a recommendation. Specifically, the case studies published so far present only half the story: discussing either the strengths of a non-meeting-based method [10] or the weaknesses of meeting-based methods [14]. Until now, no study has done a

*This research was supported in part by a grant from the National Science Foundation (CCR-9403475).

ICSE 97 Boston MA USA

side-by-side comparison designed to compare meeting-based and non-meeting-based software review and assess their relative costs and benefits directly.

This paper presents the results of a controlled experimental study designed to shed additional light on the strengths and weaknesses of meetings for formal technical review. The study compared various measures of cost and effectiveness among a meeting-based review method and a non-meeting-based review method. Analysis of the data indicated that the meeting-based method was significantly more costly, and was significantly more effective in filtering out false positives. However, we were unable to observe a significant difference in defect detection effectiveness between the two methods. Finally, participants strongly preferred the meeting-based method and believed it led to higher review quality, even though the empirical data did not support this conclusion. These findings suggest that the decision to use or discard meetings is more complicated than either traditional proponents, who find it universally necessary, or the recent review revisionists, who would eliminate it altogether.

The remainder of this paper is organized as follows. The next section describes motivates our study with a discussion of prior research related to the costs and benefits of review meetings. The following section presents the experiment, including the design, instrumentation, and procedures. The next sections present the results and our conclusions. The final section present our proposals for future research in this area.

RELATED WORK

Although reviewing software is as old as programming itself, the first structured, measurement-based group process for software review was Michael Fagan's Inspection method [3]. Fagan Inspection essentially consists of a five step process:

- *Overview:* the author presents an overview of the scope and purpose of the work product.
- *Preparation:* reviewers analyze the work product with the goal of understanding it thoroughly.
- *Inspection meeting:* the inspection team assembles and the reader paraphrases the work product. Reviewers raise issues that are subsequently recorded by the scribe.
- *Rework:* the author revises the work product.
- *Followup:* The moderator verifies the quality of rework and decides if reinspection is required.

Two aspects of Fagan Inspection are especially relevant to the question of the effectiveness of meetings in formal technical review. First, the goal of the preparation phase is to gain a thorough understanding of "intent and logic" the work product, *not* to identify defects. It is only during the Inspection meeting phase that defect identification becomes an explicit goal. Fagan notes that "sometimes flagrant errors are found during [preparation], but in general, the number of errors found is not nearly as high as in the [inspection meeting]" [3]. Second, the Inspection meeting involves a specific technique, paraphrasing, which generates an in-depth analysis of the entire document in real-time by the review team during the meeting.

These two factors, the preparation goal and the meeting technique, have been manipulated extensively in the design of new FTR methods by other researchers and practitioners. One common modification is to introduce defect detection as an explicit goal in preparation. In these cases, the reviewers note defects on a preparation form or on the work product itself prior to the inspection meeting. Thus, reviewers now have two goals to satisfy during preparation: comprehension of the work product and detection of defects.

A second modification is to change the meeting technique from paraphrasing to defect collection [6, 7]. This shifts the focus of the meeting away from the work product and onto the issues raised during preparation.

The Active Design Review technique [10] invented by David Parnas makes even more radical modifications to the preparation goals and meeting technique. Active Design Reviews were designed to address three perceived weaknesses of inspection methods:

- If reviewers do not adequately comprehend the document, then they are unlikely to discover the important defects.
- Each reviewer should have a specialized area of concern to minimize overlap and maximize coverage of the work product.
- A meeting of the whole group is unnecessary for defect collection.

Active Design Reviews address these issues by requiring reviewers to fill out individually customized questionnaires during preparation that assess their comprehension of the work product and point them toward areas prone to defects. The group meeting is eliminated. Instead, the author meets with each reviewer individually to go over their questionnaires and gather feedback on the work product. Parnas deployed this method for the design of a military flight navigation system with favorable results, although he did not report any quantitative measures of effectiveness.

Larry Votta built upon Parnas' research in a study of development groups at Bell Labs of Lucent Technologies Inc. (formerly AT&T Bell Laboratories) [14]. He collected data on the perceived utility of meetings by developers as well as several statistics on their outcome. His data showed that within the development environment studied, scheduling conflicts appeared to lengthen the total time of an inspection by approximately 30%. Furthermore, he was unable to demonstrate the presence of "synergy" within the inspection meetings—the number of new issues raised during the meeting appeared to balance out the number of issues found during preparation that were "lost" (i.e. not recorded) during

the subsequent meeting. A related study by Votta and others found that 90% of the defects were found during the preparation phase, leaving only 10% discovered during the meeting [2]. These results appear to support Parnas' claim that whole group meetings are unnecessary for defect collection.

Parnas' claim and Votta's results stand in direct contradiction to those of Fagan. While Fagan observed that many more errors are found at the inspection meeting, Votta and his colleagues observed the opposite. This conflict in findings can be attributed to differences in the goals and techniques for preparation and meeting between the two methods. In Fagan Inspection, the objective of preparation is comprehension, and defect discovery does not become an explicit goal until the Inspection meeting. In both Active Design Reviews and the Inspection method as practiced by development groups at Bell Labs, the objective of preparation is both comprehension and defect discovery. Defect collection, not discovery, is the primary goal of the Inspection meeting. Given this difference in objectives, it is not surprising that Fagan found the Inspection meeting so productive for defect discovery, while Votta et al. did not.

Thus, Votta's work provides evidence that group meetings may not be necessary for an Inspection method whose meeting goal focusses on defect collection. However, it does not show that meetings are not useful when their goal is defect discovery through paraphrasing. Furthermore, Fagan asserts that "a team is most effective if it operates with only one objective at a time." If Fagan is right, then perhaps the mixing of comprehension and defect discovery during preparation by Bell Labs developers leads to decreased review effectiveness.

If Votta's study indicates that whole group meetings do not contribute significantly to review effectiveness when the meeting goal is defect *collection*, the next step is to determine if whole group meetings contribute significantly to review effectiveness when the meeting goal is defect *detection*. To provide insight into this issue, we designed and carried out a controlled experimental study to assess whether or not "real" group meetings using paraphrasing for the purpose of defect detection can outperform "nominal" group meetings also using paraphrasing for the purpose of defect detection. A "real" group is one in which the participants meet face-to-face and interact with each other to accomplish the group task. A "nominal" group is one in which the participants work individually without interacting with each other, and their individual results are pooled together to accomplish the group task. Although there is prior research on real vs. nominal group performance, these studies have focussed on idea generation, not software review [1, 9].

If Fagan's results generalize, then real groups using meetings should outperform the nominal groups. If Votta's results generalize, then the individuals in the nominal groups should outperform the real groups. This experiment, and our results, are presented next.

THE EXPERIMENT

Our experiment compared the performance of real group and nominal group reviews. The main research question was, "Are there differences in detection effectiveness (the number of program defects detected) and detection cost (the amount of effort/time to find a defect) when subjects review source code using a synchronous, same-place same-time interaction (real groups) versus an asynchronous, same-place same-time interaction (nominal groups)?". We also explored several other questions, including: whether the two methods differ in their ability to detect certain classes of defects; whether the two methods differ in their ability to detect "false positives" (issues raised that are not actual defects); the level of defect discovery duplication in nominal groups; the level of synergy in real groups; the levels of reviewer satisfaction with each method; and the levels of reviewer confidence in the outcome with each method. (Additional research questions considered in this experiment are discussed in Danu Tjahjono's Ph.D. thesis [13].)

The subjects were 27 undergraduate students enrolled in ICS-313 (Programming Language Theory) and 45 undergraduate students enrolled in ICS-411 (System Programming) classes at the University of Hawaii in the Spring of 1995. The subjects were assigned to groups of size 3, for a total of 24 different groups. Each group performed two reviews, once using a real group review method and once using a nominal group review method.

To help obtain a controlled experiment, we implemented the real group and nominal group review methods using CSRS, the Collaborative Software Review System [8]. CSRS helped us to both standardize the review process for each of the two review methods, and also minimize the process differences between the two review methods apart from the use of real versus nominal groups. The real group method and corresponding software support was called EGSM (Experimental Group Synchronous Method) and the nominal group method and software was called EIAM (Experimental Individual Asynchronous Method).

Experimental Design

The experimental design involved one factor (group interaction) with two treatments: real group interaction and nominal group interaction. The experimental design was a balanced design in which each group reviewed two sets of source code using two different group interactions. Both the source code and synchronicity were assigned to the groups randomly.

We carried out the experiment in two rounds, the first round using the ICS-313 students and the second round using the ICS-411 students. The source code reviewed by the students was based upon recently completed exercises in the two classes, so that the students were very familiar with the ideas underlying the review materials. The ICS-313 groups reviewed two portions of an Employee database application written in C++, and the ICS-411 groups reviewed two por-

tions of a two-pass assembler written in C.

Figure 1 shows the experimental design for each of the two rounds.

Round 1: ICS-313 Groups

	Employee1	Employee2
EGSM	$G1^1, G6^1, G8^1, G9^1$	$G2^2, G3^2, G4^2, G5^2, G7^2$
EIAM	$G2^1, G3^1, G4^1, G5^1, G7^1$	$G1^2, G6^2, G8^2, G9^2$

Round 2: ICS-411 Groups

	Pass1	Pass2
EGSM	$G3^2,G4^2,G9^2,G10^2, G11^1,G12^2,G13^1$	$G1^2,G2^2,G5^2,G6^1, G7^2,G8^1,G14^1,G15^2$
EIAM	$G1^1,G2^1,G5^1,G6^2, G7^1,G8^2,G14^2,G15^1$	$G3^1,G4^1,G9^1,G10^1, G11^2,G12^1,G13^2$

Figure 1: Source code and group assignments for the two rounds. "G" indicates a group. The superscript indicates the order in which the source code was reviewed.

Variables

The experiment manipulated the independent variable, *group interaction*, with two treatments, real group and nominal group.

For the main experimental question, the experiment manipulated two dependent variables, or review measures: *defects*, the total number of distinct, valid defects detected by the group, and *effort*, the total review time spent by the group. For the other research questions, we manipulated several additional dependent variables, including the review measures: *false positives*, the number of invalid defects recorded by the group; *duplicates*, the number of duplicate defects found during nominal group review; and *synergy*, the number of defects found through interaction of two or more people during real group review.

Threats to internal validity are those factors that may affect the values of the dependent variables apart from the setting of the independent variable.

To minimize selection effects in the ICS-313 round, we rated each individual's skill as low, medium, or high, based upon their grades in prior assignments. We then selected a member at random from each category to form groups of three. To minimize selection effects in the ICS-411 round, we chose individuals at random to form groups of three.

We randomized the order in which the two review methods were presented to groups to minimize training effects. These effects were also reduced through a training session prior to the experiment in which subjects practiced the use of CSRS and the software review methods.

Finally, we minimized the differences between the two documents inspected in both rounds by ensuring that the two documents had approximately the same numbers of defects of the same types. We also minimized such instrumentation effects by having all groups inspect both documents.

Threats to external validity are those factors that limit the applicability of the experimental results to industry practice. Such threats include: the student reviewers may not be representative of professional programmers, the software reviewed may not be representative of professional software, and the inspection process may not be representative of industrial practice.

These threats are real. Overcoming the first two threats is best accomplished by replication of this study using industrial programmers with real work products. To support this replication, our experimental materials and apparatus are freely available via the Internet. To minimize the third threat, we based our experimental review methods on descriptions of industrial practice of software review, such as Gilb's Inspection [6].

Analysis Strategy

Most of the research questions were tested using the Wilcoxon signed rank test [5]. This non-parametric test of significance does not make any assumption regarding the underlying distribution of the data. It is based on the rank of differences between each pair of observations in the dataset.

The data analysis proceeds in the following way. Assume that the data are a set of N paired observations on X and Y. The difference, d, between each pair is calculated. If the two observations in a pair are the same, then d = 0 and the pair is deleted from the analysis. The d's are then ranked without regard to sign; that is, the absolute values $|X_i - Y_i|$ are ranked. A rank of 1 is assigned to the smallest d, of 2 to the next smallest, and so on. The sign of the difference d is then attached to each rank. Denote the sum of the positive ranks by W_+ and the sum of the negative ranks by W_-. The normal deviate z (z-value) is given by

$$z = \frac{W_- - \frac{N(N+1)/4}{4}}{\sqrt{\frac{N(N+1)(2N+1)}{24}}}, \text{ where } W = W_+ \text{ if } W_+ \leq W_- \text{ else } W_-.$$

The p-value of z is then used to test the null hypothesis concerning X and Y, that is, that there is no significant differences between X and Y. If the p-value is less than the significance level $\alpha = 0.05$, then we reject the null hypothesis, and can conclude that there is a significant difference between X and Y.

Experimental Instrumentation

We developed two basic instruments for this experiment:

the source code materials reviewed by the subjects, and the experimental apparatus using CSRS.

Source code review materials
The experimental review materials were based on programs recently implemented by the students themselves. Two sets of source code with approximately the same size were selected. The code was re-edited and re-compiled to ensure that it had no syntax errors. Natural language specifications for each procedure or function in the source code were provided.

In both rounds, the defects were mostly logic, computation, and data handling problems, such as missing or incorrect condition tests, forgotten cases or steps, and incorrect data access. Some of these defects were specific to the C/C++ languages, such as memory leaks. None of the defects, however, involved an incorrect specification. In fact, the participants were told beforehand that when the code did not conform to the specification, then the specification should be assumed correct, and the code was therefore incorrect.

For the ICS-313 round, the programs implemented an Employee database using the C++ programming language. One program used an array implementation of the database, and the other used a linked-list implementation. The source code was seeded with natural defects, in other words, defects made by the students themselves. To obtain these defects, the students were asked to submit the programs right after first successful compilation. We seeded 20 defects in each of the two programs for the ICS-313 round, but by the end of the experiment, we documented 23 defects in the array implementation and 25 defects in the linked list implementation.

For the ICS-411 round, the programs implemented a two-pass assembler using the C programming language. The two programs corresponded to Pass-1 and Pass-2 of the assembler. As in the ICS-313 round, the two programs had approximately equal numbers of defects and types of defects. Unlike the ICS-313 round, the defects were seeded in the same relative location. For example, when a defect of type uninitialized variable was seeded in the beginning of the function Pass-1, the same type of defect was also seeded in the beginning of the function Pass-2. We seeded 19 defects in each of the two programs for the ICS-411 round, but by the end of the experiment, we documented one additional defect in the Pass-1 source code.

Experimental Apparatus
To help ensure that all groups carried out review the same way, and to facilitiate data collection, we used the CSRS computer-mediated software review environment as the experimental apparatus for this study. The data and process modelling languages of CSRS were used to implement two different review methods that differed only with respect to group interaction.

Figure 2 shows a screen image from the EGSM review from the Pass2 assembler source. In both methods, CSRS presented subjects with this three window user interface, where the set of functions/procedures to be reviewed are shown in the upper right screen, the function or procedure currently reviewed is shown in the left screen, and defects raised by reviewers are entered in a commentary window in the lower right screen.

The EIAM interface differs only slightly from that shown in Figure 2. All issues in EIAM are private to each reviewer, but public in EGSM among all reviewers of a given group. Similarly, the criticality field value is private to each reviewer in EIAM, but public (all votes are shown) in EGSM. EGSM also includes a field called "Suggested-by" that allows each reviewer to indicate who suggested the issue, and is used to measure group synergy. This field is not included in EIAM, since synergy is not present by definition.

Experimental Procedures
Training
All subjects attended a set of lectures on formal technical review. This lecture explained the goals of formal technical review and the specific procedures to be used in this study. The training was based upon software review tutorial materials used by one of the authors (Philip Johnson) for industry.

The subjects were then assembled into three person teams according to the procedures specified above. They next attended a two hour training session to familiarize themselves with the CSRS review environment and the EGSM and EIAM review methods. During this session, they practiced review on sample source code implementing a "BigInteger" data abstraction. They practiced the use of paraphrasing as a mechanism to analyze software and discover defects.

General Review Procedures
Both EGSM and EIAM consisted of a single review phase, whose objective was defect detection. Subjects were told to not determine how to correct any defects they discovered, but to merely note their presence.

Since all subjects had recently completed the implementation of a program quite similar to the review materials, there was no need for a "preparation" phase with the objective of comprehension, or to mix comprehension with defect discovery. The subjects were already very familiar with the requirements, specifications, and design of the software under review.

Both methods used the paraphrasing method from Fagan Inspection as the analysis technique. For EGSM, one of the three subjects in each group was assigned to the role of Presenter, and he/she verbally summarized the source code in a line-by-line fashion. The presenter also acted as a reviewer and was free to discover defects. For EIAM, subjects paraphrased the source code silently.

In EGSM, the subjects collaborated fully with each other. As the Presenter paraphrased the code aloud, any of the re-

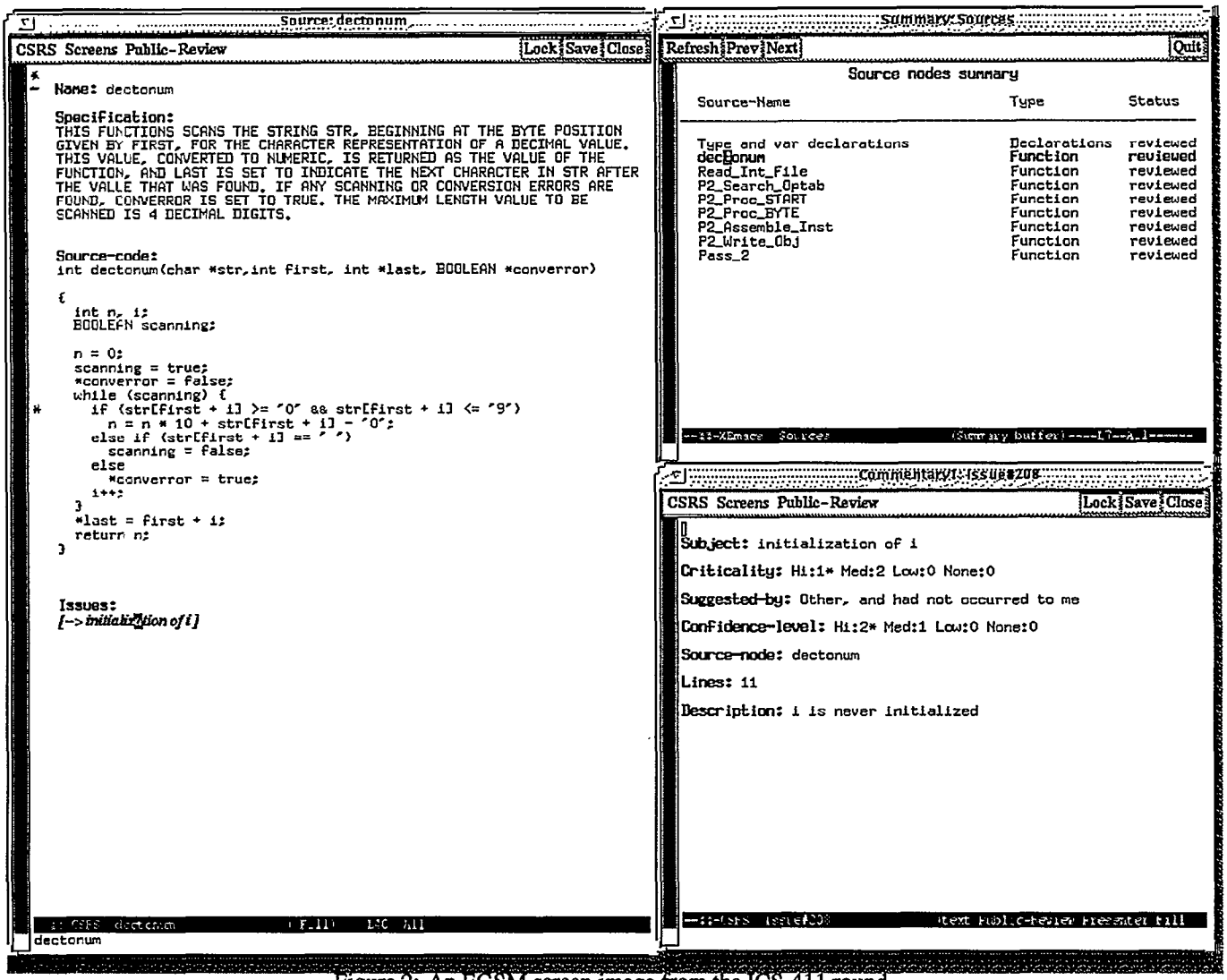

Figure 2: An EGSM screen image from the ICS-411 round.

view team members were free to interrupt with suggestions of potential defects. Others would then confirm or reject the suggestion. If disagreement continued, the team would vote on whether or not to include the issue as a defect. The Presenter was the only one who could enter issues, and all review screens were kept synchronized. This prevented reviewers from "wandering off" into the code and kept the reviewers together.

In EIAM, subjects worked entirely independently, raising issues and noting them by themselves. For administrative purposes, each EIAM team did meet in the same room at the same time, but all interaction between members was prevented.

In all review sessions, one of the authors (Danu Tjahjono) acted as an "external moderator". The purpose of this role was simply to guarantee correct execution of the process. The external moderator performed such tasks as ensuring that paraphrasing was used in EGSM, that discussion did not wander, and that any questions about the CSRS user interface could be answered quickly and correctly.

The review time for every session was limited to a maximum of three hours. However, no review team or reviewer used more than 2.5 hours to complete the review of each program.

Round 1: ICS-313

For the ICS-313 round, 27 students participated and were split into 9 groups. Four of the groups were randomly chosen to use the EGSM method to review the array implementation of the employee database, while the remaining five groups used EIAM to review the same source materials. All groups then switched methods and reviewed the linked list version of the employee database. The round was completed within two weeks.

Round 2: ICS-411

For the ICS-411 round, 45 students participated and were split into 15 groups. Seven of the groups were randomly chosen to

use the EGSM method first, while the other eight groups used the EIAM method first. The review material encountered first was also randomly assigned, with seven groups encountered the Pass1 source code for their first review, while the other eight groups encountered the Pass2 source code first. All groups then switched both review method and source material for their second review session. This round was also completed within two weeks.

Data Collection

We collected data through two mechanisms: CSRS and questionnaires filled out by all subjects at the end of each review session. CSRS stored all defects recorded by both real groups (using EGSM) and nominal groups (using EIAM) in an internal database for later analysis.

For each real group, the value of the dependent variable *defects* was calculated as the total number of defects entered into CSRS by the group, minus those that we manually determined to be *false positives*.

For each nominal group, the value of *defects* was calculated by summing all of the errors found by the individuals in a particular group, then subtracting both those defects that we manually determined to be *false positives* as well as any defects we determined to be *duplicates*, i.e. found by more than one member of the group.

For all groups, the value of *effort* was calculated as the sum of the total time spent on review by each member of the group. CSRS used a timestamp-based mechanism to automatically record the time spent using the system by each reviewer.

For each real group, the value of *synergy* was determined by analysis of the value of the "Suggested-by" field for each recorded defect. The Suggested-by field had four possible values: "Me", "Me, but inspired by others", "Other but also occurred to me", and "Other and had not occurred to me". If one or more of the reviewers recorded "Me" as the value of this field, then synergy was defined as not occurring during the discovery of this particular defect. The value of *synergy* for a real group was calculated as the total number of defects found, minus those for which synergy did not occur.

Finally, each subject filled out three questionnaires during the study. A questionnaire evaluating the subject's attitudes towards the EGSM method and the EGSM review group experience was administered after the EGSM review. A similar questionnaire on EIAM was administered after the EIAM review. A final questionnaire evaluating subject preference for EIAM or EGSM, and their satisfaction with CSRS was administered at the end of the study. Most of the questions required subjects to respond by circling one number on a five point scale, although a few questions were open ended and asked for explicit commentary.

EXPERIMENTAL RESULTS

Figure 3 summarizes the results of comparing the performance of real groups (EGSM) and nominal groups (EIAM) for certain review measures using the Wilcoxon signed rank test. We show the results from analyzing the data for each round separately and when grouped together.

A "-" in a column indicates that we were not able to detect a significant difference between real and nominal groups for the review measure. A ">" indicates that real group performance using EGSM was significantly higher ($p<.05$) than nominal groups using EIAM for the corresponding review metric, while a "<" indicates that real group performance was significantly lower ($p<.05$) than nominal group performance.

EGSM vs. EIAM

Review Measure	ICS-313	ICS-411	All
Defects	-	-	-
Cost/Defect	>	-	>
Effort	>	-	>
Issues	<	<	<
False positives	<	<	<

Figure 3: Selected Wilcoxon signed rank test results

As shown in Figure 3, we were unable to detect a significant difference between real and nominal groups with respect to the number of valid defects discovered. In other words, this study was unable to demonstrate that review meetings for the purpose of defect discovery outperformed individuals working independently. Real groups found an average of 43% of all known defects, while nominal groups found an average of 46% of all known defects.

On the other hand, we found that real groups using meetings were significantly more costly than individuals working independently. The average effort required per defect was 41 minutes for real groups and 34 minutes for nominal groups, and the average overall effort for a review session was 5:57 hours for real groups and 5:11 hours for nominal groups.

We also found that individuals working independently in nominal groups raised significantly more total issues, an average of 14 issues per nominal group session, than real groups which raised an average of 9 issues per session. However, nominal groups had a significantly greater percentage of false positives (22% of all raised issues, on average) than those working in real groups (5.3% of all raised issues, on average).

Figure 4 summarizes some of the major results from analysis of measures that apply only to one of the two review methods. *Synergy* indicates the percentage of defects in which synergy played a role for the set of EGSM review sessions. We found that synergy participated in the process of defect discovery about a third of the time overall.

Duplicates indicates the average percentage of defects that were discovered by more than one reviewer in a given EIAM

group for the set of review sessions. Again, about a third of the defects were discovered by more than one reviewer during individual review.

Data	ICS-313	ICS-411	All
Synergy (EGSM only)	42%	21%	29%
Duplicates (EIAM only)	29%	31%	30%

Figure 4: Selected method-specific measurements

Finally, Figure 5 summarizes some of the major results from analysis of the questionnaire data. The questionnaire results are shown pooled for all subjects. The numbers shown indicate the percentage of subjects agreeing with the focus of the question by responding with a 4 or 5 on a five point scale. Depending upon the way the question was worded, a response of 4 or 5 corresponded to "high" or "very high", "true" or "very true", or "somewhat EGSM" or "strongly EGSM". .

Question Focus	% Agreement
Confident in EGSM review quality	78%
Confident in EIAM review quality	42%
Satisfied with EGSM method	81%
Satisfied with EIAM method	61%
More productive using EGSM than EIAM	72%
Prefer EGSM over EIAM	63%

Figure 5: Selected questionnaire results

These results show that over three quarters (78%) of the participants felt confident or very confident in the quality of their review using EGSM, while less than half (42%) felt confident or very confident in the quality of their EAIM review. 81% were satisfied or very satisfied with the EGSM review method. This percentage drops to 61% for the EIAM review method. Finally, most participants felt they were more productive using EGSM than EIAM (72%) and most preferred to use EGSM over EIAM (63%).

CONCLUSIONS

In this study, we used a real group vs. nominal group approach to gaining insight into the contributions and importance of meetings in software review. Real groups reflect a meeting-based approach to defect discovery, closely similar to Fagan Inspection. Nominal groups reflect an individual-based approach to defect discovery, closely similar to review methods such as those studied by Votta et. al in which defect discovery is an objective of the individual preparation phase.

Our goal was to provide additional insight into the question of whether meetings provide an essential contribution to the review meeting process, as Fagan asserts, or whether meetings are simply costly without corresponding benefits, as Votta and his colleagues assert. Interestingly, our data appears to provide partial support for both of these assertions. It is important, when interpreting these conclusions, to remember that this data was collected using student programmers. Care should be taken when applying these results to professional programming groups.

Support for non-meeting-based review

We were unable to find any significant differences between meeting-based groups and individuals in the number of valid errors discovered[1]. However, we did find a significant difference in the cost of review: meeting-based review required more total effort and more effort per defect.

Furthermore, the data suggests that changes to the review method could potentially lead to individuals significantly outperforming groups in defect discovery. This is because the nominal groups in this study did generate significantly more *issues* than the real groups, but their overall defect discovery was reduced by the presence of duplicates. In research by Adam Porter and colleagues [11], they hypothesized that "systematic" defect discovery techniques, such as their scenario-based technique would outperform "ad-hoc" techniques (such as paraphrasing) in part because of "reduced reviewer overlap" (duplication of defects). In their experiments, systematic defect discovery techniques did indeed significantly outperform ad-hoc techniques. Therefore, it is plausible that individual reviewers could employ a systematic defect discovery technique that reduces defect duplication to the point where individuals would both outperform groups and have lower overall cost.

In summary, the fact that we were able to observe significantly more cost for meeting-based review, that we were unable to observe significantly more defects discovered by meeting-based review, and that related research suggests a mechanism for improving non-meeting-based review effectiveness all support the assertions of Votta and his colleagues.

Support for meeting-based review

Our data does not unequivocably support a non-meeting-based review style, however. Real groups using meetings generated significantly less false positives (issues that were not valid defects) than individuals in nominal groups. Meetings were more effective than individuals at filtering issues.

Real groups also displayed a substantial degree of synergy: over 40% of the issues detected by ICS-313 groups were a product of some level of synergy, and almost 30% of the issues overall resulted from synergy.

Most subjects indicated a personal preference for the meeting-based review method. Most felt more confident in the quality of meeting-based review, and more satisfied with it as a review process. Indeed, almost three quarters of the subjects believed that meeting-based review was more productive than individual review, even though the primary review measures suggest the opposite!

[1]It is important to recognize that this does not mean that we found the two techniques to be equally effective. There may be differences that we were unable to detect simply due to the design or conduct of the experiment.

Why did the subjects feel more confident in meeting-based review, and even believe it to be more productive? In the meeting-based review sessions, reviewers ended the session with an awareness of all the defects found by the group. In the nominal groups, they left the session knowing only about their own (potential) defects. (The average group found about 50% more defects during a session than the average individual.) Furthermore, in meeting-based review, all of the potential defects raised were subject to immediate verification by other members present, providing everyone with some confidence that the defects raised were valid. In the nominal groups, no such external validation took place. Finally, subjects felt the presence of synergy during meeting-based review: they clearly discovered defects through interaction with each other, even though this did not lead to a measurable difference in defects found between the two review methods.

Thus, meetings appear to serve an important function, but that function is not necessarily to increase the total number of defects discovered. Rather, the meeting serves as a mechanism to share the state of the review with all the participants, to enable them to evaluate its effectiveness, and to foster a sense of collective ownership and responsibility for its outcome.

FUTURE DIRECTIONS

Replication

An essential future direction is to assess the validity of these results, our conclusions, and the threats to external validity through replication. In particular, replication will help establish if meeting-based review for the purpose of defect detection is truly more costly than individual review without also detecting significantly more errors. (It is interesting to note that our findings do not confirm those of the behavioral science researchers, who found that nominal groups outperform real groups.) Also, replication can also provide more insight into the issue of synergy, in which our findings that substantial synergy occurred appear to contradict those of Votta.

To support this process, the CSRS system, the EGSM and EIAM methods, and the experimental procedures are all available on the World Wide Web. See the CSRS Home Page at:

`http://www.ics.hawaii.edu/~csdl/csrs/`

Whither review meetings?

This study was designed to help resolve a central question in modern software review practice: are meetings worth their cost? As a first step, we identified the importance of the objectives for preparation and the meeting. In Fagan Inspection, where defect discovery is restricted to the meeting, meetings obviously account for most defect discovery. In Bell Labs inspection, where defect discovery occurs during preparation and the primary objective of the meeting is defect collection, meetings do not account for much defect discovery.

Our study found that a meeting-based method is significantly more costly than its non-meeting-based alternative, but could not demonstrate that they find significantly more defects to offset this cost. However, meetings do provide other benefits, such as filtering false positives and improving group awareness of and confidence in the review. Meetings may also serve other social functions, such as team building and education. A useful future direction is to perform a longitudinal, case study experiment in industry that follows the transition from a traditional meeting-based review method to a non-meeting-based review method, and assesses the resulting costs and benefits of the change.

Computer-mediated software review

This study also demonstrates the potential of computer-mediated software review environments, both as an experimental platform and as practical support of software review. CSRS provided substantial help in supporting and standardizing the review practice by both real and nominal groups. Reviewer satisfaction with CSRS was very high; 72% of the participants indicated they would prefer to use CSRS over manual review.

Computer-mediated software review may also play a key role in overcoming the costs of software review meetings without losing their most important benefits. In prior research [8], we implemented a method called FTArm using CSRS that minimized or eliminated the need for face-to-face review meetings while preserving the ability of participants to filter false positives, retain group awareness of the state of review, enable educational opportunities, and so forth. We look forward to future research that assesses the costs and benefits of computer-mediated software review within an industrial setting.

ACKNOWLEDGEMENTS

We gratefully acknowledge the efforts of the ICS 313 and ICS 411 classes, who graciously served as subjects for this study. Richard Halverson, instructor of ICS 411, also helped with aspects of the ICS 411 round design and implementation. Finally, we would like to acknowledge the many contributions to this research from the other members of the Collaborative Software Development Laboratory, including: Cam Moore, Rosemary Andrada, Dadong Wan, Jennifer Geis, Julio Polo, and Russ Tokuyama.

REFERENCES

[1] M. Diehl and W. Stroebe. Productivity loss in brainstorming groups: Toward the solution of a riddle. *Journal of Personality and Social Psychology*, (53):497–509, 1987.

[2] Stephen G. Eick, Clive R. Loader, M. David Long, Scott A. Vander Wiel, and Lawrence G. Votta. Estimating software fault content before coding. *Proceedings of the 14th International Conference on Software Engineering*, pages 59–65, May 1992.

[3] Michael E. Fagan. Design and code inspections to reduce errors in program development. *IBM System Journal*, 15(3):182–211, 1976.

[4] Michael E. Fagan. Advances in software inspections. *IEEE Transactions on Software Engineering*, SE-12(7):744–751, July 1986.

[5] George A. Ferguson and Yoshio Takane. *Statistical Analysis In Psychology And Education.* McGraw-Hill Book Company, 6 edition, 1989.

[6] Tom Gilb and Dorothy Graham. *Software Inspection.* Addison-Wesley, 1993.

[7] Watts S. Humphrey. *Managing the Software Process.* Addison Wesley Publishing Company Inc., 1990.

[8] Philip M. Johnson. An instrumented approach to improving software quality through formal technical review. In *Proceedings of the 16th International Conference on Software Engineering*, Sorrento, Italy, May 1994.

[9] B. Mullen, C. Johnson, and E. Salas. Productivity loss in brainstorming groups. *Basic and Applied Social Psychology*, (12):3–24, 1991.

[10] David L. Parnas and David M. Weiss. Active design reviews: Principles and practices. *Proceedings of Eighth International Conference on Software Engineering, London, England*, pages 132–136, August 1985.

[11] Adam A. Porter, Lawrence G. Votta, and Victor R. Basili. Comparing detection methods for software requirements inspections: A replicated experiment. *IEEE Transactions on Software Engineering*, 21(6):563–575, June 1995.

[12] Glen W. Russell. Experience with inspection in ultralarge-scale developments. *IEEE Software*, January 1991.

[13] Danu Tjahjono. *Exploring the effectiveness of formal technical review factors with CSRS, a collaborative software review system.* PhD thesis, Department of Information and Computer Sciences, University of Hawaii, August 1996.

[14] Lawrence G. Votta. Does every inspection need a meeting? In *Proceedings of the ACM SIGSOFT 1993 Symposium on Foundations of Software Engineering*, December 1993.