Assessing Software Review Meetings: Results of a Comparative Analysis of Two Experimental Studies

Adam A. Porter* aporter@cs.umd.edu

Philip M. Johnson† johnson@hawaii.edu

February 22, 1997

Abstract

Software review is a fundamental tool for software quality assurance. Nevertheless, there are significant controversies as to the most efficient and effective review method. One of the most important questions currently being debated is the utility of meetings. Although almost all industrial review methods are centered around the inspection meeting, recent findings call their value into question. To gain insight into these issues, the two authors of this paper separately and independently conducted controlled experimental studies.

This paper discusses a joint effort to understand the broader implications of these two studies. To do this, we designed and carried out a process of "reconciliation" in which we established a common framework for the comparison of the two experimental studies, re-analyzed the experimental data with respect to this common framework, and compared the results. Through this process we found many striking similarities between the results of the two studies, strengthening their individual conclusions. It also revealed interesting differences between the two experiments, suggesting important avenues for future research.

1 Introduction

The value of software review as a mechanism for software quality improvement has been demonstrated repeatedly for over twenty years. Beginning with the landmark work of Michael Fagan at IBM in 1976, structured review mechanisms such as inspection have been shown repeatedly to be an extremely effective means to find work product defects early in the software development process.

As the benefits of such structured review processes (typically referred to as "Formal Technical Review" or FTR) became more visible, researchers and practitioners began to devise variations on Fagan's original method. For example, Tom Gilb developed a comprehensive inspection method with precisely defined phases, metrics, and suggested process rates for optimum defect removal effectiveness [10].

With few exceptions, these variations never challenged a fundamental premise of Fagan's original method: that a face-to-face meeting of the entire review team is essential to the review's success. While researchers have proposed changing the manner in which reviewers prepared for the meeting, or even the manner in which the meeting was conducted, the need for a meeting was never questioned. Fagan, Gilb, and others have argued that meetings enable a kind of "synergy" between participants, in which defects not found by reviewers working individually suddenly come to light. They also argue that meetings educate the participants, clarify requirements, and provide milestones that facilitate progress.

Unfortunately, meetings have demonstrated and substantial costs. They require the simultaneous attendance of all participants. Their effectiveness depends on satisfying many conditions, such as adequate preparation, readiness of the work product for review, high quality moderation, and cooperative interpersonal relationships. Meeting-based review appears to add 15-20% new overhead onto development costs [20], and simple scheduling issues have been shown to lengthen the start-to-finish interval for review by almost one third [22].

The costs of meeting-based review have stimulated more recent research designed to investigate whether new review methods can be devised that minimize or eliminate the cost of meetings while preserving the remaining benefits of review. Such research has ranged from the design of computer-supported cooperative work systems that implement an asynchronous, non-meeting-based review procedure [14], to alternative manual methods that also shift the process away from reliance on meetings [17].

*This work is supported in part by a National Science Foundation Faculty Early Career Development Award CCR–9501354.

†This work is supported in part by a grant from the National Science Foundation (CCR-9403475).

In addition to this research, a few studies have tried to directly assess the value of inspection meetings. These experimental studies attempt to quantify the actual benefits (and in some cases, the costs) associated with software review meetings. Each provides some insight into the question of meeting-based review, but each also suffers from the inevitable limitations of controlled experimental studies. First, it is difficult to assess the generality of the results, and whether they would apply to industrial practice. Second, it is unclear what they indicate in aggregate about the appropriate future directions for research on this important issue.

This paper describes a detailed comparative analysis of two such experiments, one performed at the University of Maryland and one performed at the University of Hawaii. The experiments had a similar motivation: to assess the true contributions of meetings to software review. They differed in their designs, instruments, review documents, procedures, and measurements. Despite these differences, a comparative analysis revealed striking similarities in the outcomes. After reanalyzing both studies according to a common experimental design framework, five hypotheses were developed and tested against both experimental data sets. In each case, both studies confirmed or rejected the hypothesis in the same way, which increases our confidence in the generality of the findings. Furthermore, the comparative analysis of the methods employed provides insights into the design of useful experimental research on software review, and suggests the most important next steps for research.

In the next section we review prior research on meeting-based review and comparative experimental analysis techniques. We then summarize the two studies chosen for this analysis. After that we present the comparative analysis, and, finally, we discuss our conclusions and recommendations for future research.

2 RELATED WORK

Reviewing software is as old as programming itself. However, the first structured, measurement-based review process was Michael Fagan's five-step Inspection method [7].

- *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.

2.1 Review Meetings

Two aspects of Fagan Inspection are especially relevant to determining meeting effectiveness in formal technical review. First, the goal of the preparation phase is to thoroughly understand the work product's "intent and logic", *not* to identify defects. Only during the Inspection meeting does defect identification become 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]" [7]. 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. In this case, reviewers have two preparation goals: comprehending the work product and detecting defects.

A second modification is to change the meeting technique from paraphrasing to defect collection [10, 13]. This shifts the focus of the meeting away from the work product and onto the issues raised during preparation.

The Active Design Review technique [16] invented by David Parnas and David Weiss makes even more radical modifications to the preparation goals and meeting technique. Active Design Reviews were designed to address three perceived weaknesses in existing 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 and Weiss 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 Lucent (Formerly AT&T) developers [22]. 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. 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 [5]. These results appear to support Parnas and Weiss' claim that whole group meetings are unnecessary for defect collection.

Parnas and Weiss' claim and Votta's results stand in direct contradiction to Fagan's. While Fagan observed that many more errors are found at the inspection meeting, Votta and his colleagues observed the opposite. This conflict might be caused by 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 Lucent, the objective of preparation is both comprehension and defect discovery. Defect collection, not discovery, is the primary goal of the Inspection meeting. Given these different goals, it is not surprising that Fagan found the Inspection meeting so productive for defect discovery, while Votta et al. did not.

Thus, while Votta's work does provide evidence that whole group meetings may not be necessary for an Inspection method whose meeting goal involves defect collection, it does not provide evidence that meetings are not necessary for an Inspection method whose meeting 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 mixing comprehension and defect discovery during preparation by Lucent developers decreases review effectiveness.

To provide insight into this issue, two research groups, one at the University of Hawaii and one and the University of Maryland, independently designed and conducted controlled experimental studies. Their goal was to assess the effect of team meetings on defect detection effectiveness and to determine whether superior alternatives exist. This article resulted from our recognition that the two studies were similar enough in motivation to deserve comparative analysis.

2.2 Comparative Analysis

We believe that no single study gives unequivocal results. Therefore, it is imperative that we try to integrate and compare studies that address a common hypothesis. This is the only way to gain confidence that empirical results are real and not just due to random variation. However, integrating multiple studies in a credible way isn't simple. In this case, the two studies address the same issue, but they were conceived and executed independently. Thus, direct comparison of the results is impossible because the studies differ considerably in their designs, instrumentation, subject population, and analysis methods.

The classic approach to understanding what several studies say about some phenomenon is to conduct a literature review, qualitatively summarize existing results, and manually synthesize them. The drawback of this approach is that it lacks precise methods for combining different results.

A statistical approach for integrating multiple studies is called meta-analysis [11]. This approach has two steps. First, the experimenters attempt to reconcile the primary experiments – i.e define a common framework with which to compare different studies. This involves defining common terms, hypotheses, and metrics, and characterizing key differences. Next the data from the primary experiments are transformed or recalculated according to agreed upon definitions. In the second step the transformed primary data is combined and reanalyzed. Unfortunately, it is not always clear when meta-analysis is appropriate, what statistical models should be used, or when it is acceptable to combine data from disparate sources.

Our approach for comparing these two experiments falls between the classic approach and meta-analysis. As with meta-analysis, we reconciled the two experiments, but we did **not** combine any of their data. We found that the reconciliation process highlighted many of the similarities and differences between the two experiments, allowing us to better understand what data was comparable and what was not. We discuss the specific steps used in this reconciliation process in Section 5.

The following two sections describe both experiments; their designs, analysis, and results. Subsequent sections describe the reconciliation of the experiments and the comparative analysis of their results.

3 Experiment 1: University of Hawaii

3.1 Motivation

This experiment compared the performance of real group and nominal group reviews. A "real" group is one in which the participants meet face-to-face and interact with each other to accomplish the group task. In this experiment, real groups model the standard, meeting-based review method. 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. By comparing the performance of real groups with nominal groups, the experiment attempts to tease out the effects that the meeting alone has on overall review performance. Although there is prior research on real vs. nominal group performance, these studies have focussed on idea generation, not software review [4, 15].

3.2 Hypotheses

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)?". The hypothesis was that there would indeed be significant differences in both detection effectiveness and detection cost. If Fagan's and other meeting advocate's assumptions underlying the group meeting held, then it was expected that real groups would find significantly more defects due to the advantages (synergy, etc.) of a group meeting. If these assumptions did not hold, then it was hypothesized that nominal groups would outperform the real groups with respect to defect detection. In either case, it was expected that groups would cost significantly more (i.e. require significantly more effort) that nominal groups.

Besides this primary hypothesis concerning cost and effectiveness, the experiment pursued research questions concerning the ability of the two methods to detect certain classes of defects and to detect "false positives" (issues raised that are not actual defects). Space constraints preclude a complete discussion of all the research questions any hypotheses considered in this experiment; for complete details, see the dissertation[1] by Danu Tjahjono [21].

3.3 Experimental Design

3.3.1 Subjects

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.

3.3.2 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.

The experiment was 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

[1] ftp://ftp.ics.hawaii.edu/pub/tr/ics-tr-95-08.ps.Z

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 portions 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.

3.3.3 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. Other research questions required 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.

3.3.4 Threats

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

The order in which the two review methods were presented to groups was randomized in order 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, differences between the two documents inspected were minimized in both rounds by ensuring that the two documents had approximately the same numbers of defects of the same types. Instrumentation effects were also minimized 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[2]. To minimize the third threat, the experimental review methods were based on descriptions of industrial practice of software review.

3.3.5 Analysis Strategy

Most of the research questions were tested using the Wilcoxon signed rank test [8]. 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

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

3.3.6 Experimental Instrumentation

Two basic instruments were developed 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. Twenty defects were seeded in each of the two programs for the ICS-313 round, but by the end of the experiment, 23 defects were documented 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. Nineteen defects were seeded in each of the two programs for the ICS-411 round, but by the end of the experiment, one additional defect was documented in the Pass-1 source code.

[2]http://www.ics.hawaii.edu/ csdl/csrs.html

Experimental Apparatus. To help ensure that all groups carried out review the same way, and to facilitate data collection, the CSRS computer-mediated software review environment was used as the experimental apparatus for this study. The data and process modeling 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.

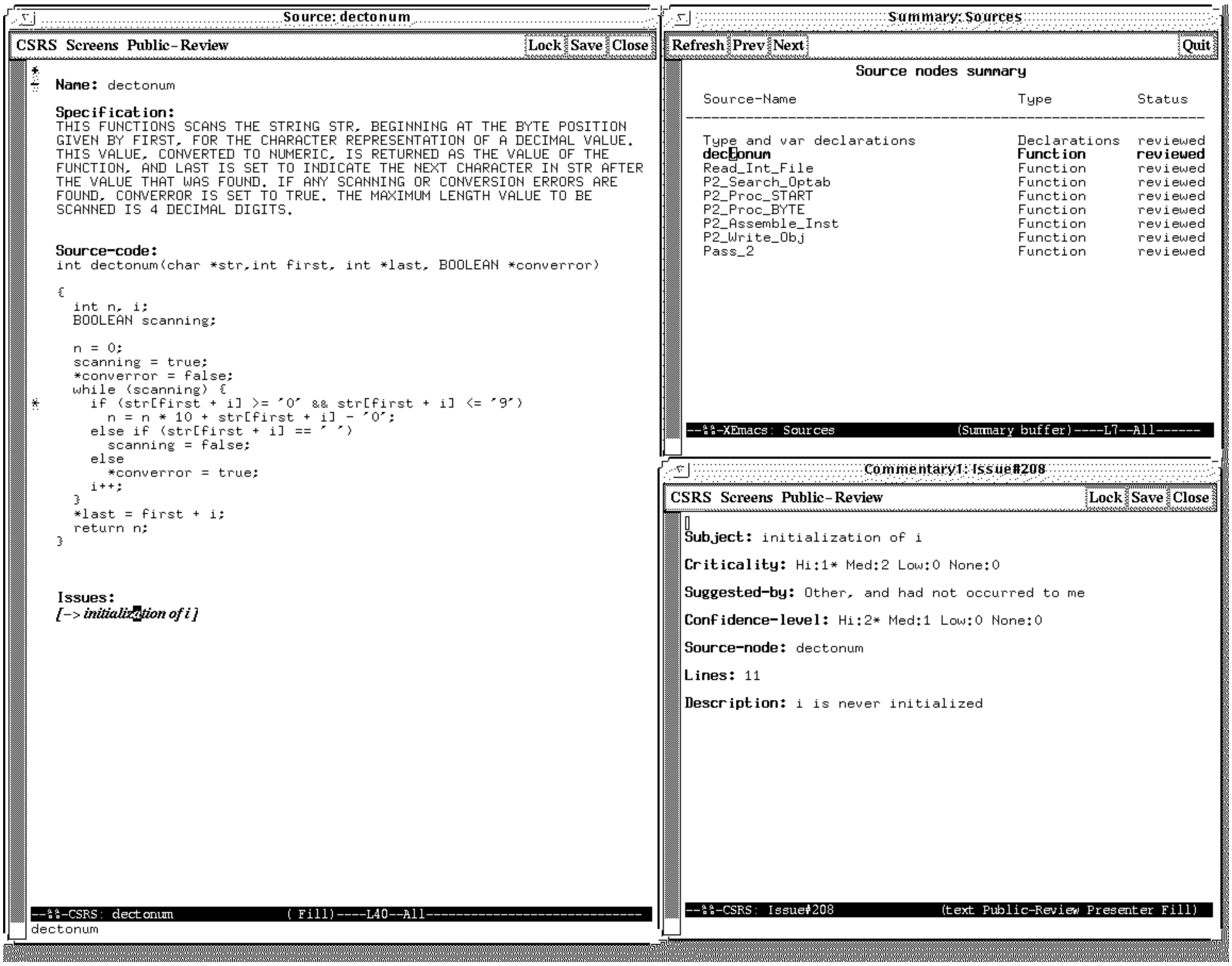

Figure 2: An EGSM screen image from the ICS-411 round.

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.

3.3.7 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 review 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.

An "external moderator" participated in all review sessions. His role was simply to guarantee correct execution of the process. For example, he ensured 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.

3.3.8 Data Collection

Data was collected 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 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 manually determined to be *false positives* as well as any defects 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 automatically recorded the time spent by each reviewer logged in to the system.

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.

3.4 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. The results from analyzing the data for each round separately and when grouped together are shown.

A "-" in a column indicates that a significant difference between real and nominal groups could not be detected 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, the data does not show 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 42.8% of all known defects, while nominal groups found an average of 46.4% of all known defects.

On the other hand, the data shows 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.

The experiment 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. 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

Round/Specification

		Round 1		Round 2	
		WLMS	CRUISE	WLMS	CRUISE
Review Method	PI	1G, 2A, 2B, 2C	1B, D		1E, 1F
	DC	1F, 2D, 2E, 2F		1C	1A
	DD	1A, 1E, 2G, 2H, 2I	1C	1D	1G

Table 1: This table shows the settings of the independent variables. The student teams are denoted 1A–1G. The professional teams are denoted 2A–2I. The student teams reviewed two documents, the WLMS and CRUISE, one per round, using one of the three review methods. The professionals only reviewed the WLMS.

4 EXPERIMENT 2: UNIVERSITY OF MARYLAND

4.1 Motivation

Many people are convinced that meetings are an essential part of successful reviews. The validity of this conviction depends on the argument that (1) many faults or many faults of specific types are found during meetings, and therefore they justify their cost; and (2) without these meetings the faults would not be found. Thus they believe that a group of reviewers is likely to be more effective working together than working separately.

To investigate these arguments, this experiment evaluated the performances of the following three review methods.

Preparation-Inspection (PI). Each reviewer individually analyzes the artifact in order to become familiar with it. The goal is not to discover faults but only to prepare for the review meeting. Reviewers are prohibited from taking any notes during Preparation. After all reviewers have completed Preparation, the team holds an Inspection meeting to find as many faults as possible.

Detection-Collection (DC). Each reviewer individually analyzes the artifact with the goal of Detecting as many faults as possible. The team then meets during the Collection phase to review the document. The meeting results will contain faults found during Detection, as well as new defects found during the meeting itself.

Detection-Detection (DD). Each reviewer individually analyzes the artifact with the goal of Detecting as many faults as possible. After all reviewers complete the Detection phase, they are asked to repeat Detection a second time, again individually, and again with the goal of detecting as many faults as possible. This approach does not involve a meeting. Instead the time is used by the reviewers to continue working individually.

These methods were chosen in order to answer two questions. The first is "Will more faults be found in reviews whose meetings have the primary responsibility for discovering faults, or will more be found in a review that involves a meeting, but in which faults may be found before the meeting starts?" The second question is, "How would review performance be affected if the time devoted to meetings were used instead for additional individual analysis?"

4.2 Hypotheses

One hypotheses of this experiment is that there is no difference in the performance of the two review methods that involve meetings. A second hypothesis is that review methods that eliminate meetings (the DD method) are at least as cost-effective as methods that rely heavily on them (the PI and DC methods) and probably more so. Moreover, this should be because the benefit of additional individual analysis (as provided by the DD method) is greater than or equal to holding review meetings.

4.3 Experimental Design

4.3.1 Subjects

The experiment was conducted twice. Once in the spring of 1995 with 21 graduate students in computer science as subjects, and once more in the fall of 1996 with 27 professional software developers as subjects. The subjects were assigned to groups of size 3, for a total of 16 different groups.

4.3.2 Design

This experiment compared the PI, DC, and DD methods for reviewing software requirements specifications (SRS). To limit its duration, the experiment used a partial factorial design in which each team participated in at most two reviews, using one of the three reviews methods in each round. Table 1 shows the settings of the independent variables.

Each complete run consisted of (1) a training phase in which the subjects were taught reviews methods and the experimental procedures, and in which they reviewed a sample SRS; and (2) an experimental phase in which the subjects conducted monitored reviews. During the experimental phase the graduate students conducted two reviews and the professionals conducted one.

4.3.3 Variables

The experiment manipulated four independent variables: (1) the review method used by each reviewer (PI, DC, or DD); (2) the review round (each reviewer participated in two reviews during the experiment); (3) the specification to be reviewed (two are used during the experiment); and (4) the order in which the specifications are reviewed (either specification can be reviewed first.)

The review method is the treatment variable. The other variables were used to assess several potential threats to the experiment's internal validity.

Due to time constraints, the professional population conducted only one review. Therefore, the only independent variable for that population is the review Method. For each review several dependent variables were measured: (1) the Individual Fault Detection Ratio, (2) the Team Fault Detection Ratio, [3] and (3) the Gain Ratio, i.e., the percentage of faults initially identified during the second phase of the review.(For the PI and DC methods the second phase is the team meeting; for the DD method it is the second individual detection activity.) These calculations are explained in detail in Section 4.3.9.

4.3.4 Threats

Four threats to internal validity were considered: selection effects, maturation effects, instrumentation effects, and presentation effects.

To minimize selection effects in both the graduate student and professional population the experimental design composed teams and assigned review methods on a completely random basis. This approach attempts to spread differences in natural ability across the review methods in an unbiased fashion. However, since each team uses only one or two of the three review methods, differences in the methods can't be completely separated from differences in reviewer ability.

Maturation effects are caused by subjects learning as the experiment proceeds. The review method used and the order in which the documents are reviewed were manipulated in order to assess this effect. These effects do not exist in the professional population since they only review one document.

Instrumentation effects result from differences in the specification documents. Such variation is impossible to avoid, but the experiments controlled for it by having each team review both documents. Again, these effects are not present in the professional population since they only review one document.

Finally, presentation effects can occur if reviewing one specification first makes it easier to review the remaining one. This possibility was controlled for by having half the teams review the documents in each of the two possible orders. Once again these effects are not present in the professional population since they only review one document.

Section 4.3.9 shows that the variation in the fault detection ratio is not explained by selection, maturation, or presentation effects.

[3] The Team and the Individual Fault Detection Ratios are the number of faults detected by a team or individual divided by the total number of faults known to be in the specification. The closer these values are to 1, the more effective the detection method. No faults were intentionally seeded into the specifications. All faults are naturally occurring.

The experiment considered three threats to external validity: subject representativeness, instrumentation representativeness, and process representativeness.

The graduate student subjects in our experiment may not be representative of software programming professionals. Although more than half of the subjects have 2 or more years of industrial experience, they are graduate students, not software professionals. Furthermore, as students they may have different motivations for participating in the experiment.

The specification documents may not be representative of real programming problems. The experimental specifications are atypical of industrial SRS in two ways. First, most of the experimental specification is written in a formal requirements notation (see Section 4.3.6). Although some industrial groups are experimenting with formal notations [2, 9], it is not the industry's standard practice. Second, the specifications used are considerably shorter than industrial specifications.

Finally, the review process in our experimental design may not be representative of software development practice. We have modeled our experiment's review process after the ones used in many development organizations, although each organization may adapt the process to fit its specific needs. Another difference is that the SRS authors are not present at our reviews, although in practice they normally would be. Finally, industrial reviewers may bring more domain knowledge to a review than our student subjects did.

4.3.5 Analysis Strategy

The analysis strategy had several steps. The first step was to find those independent variables that individually explained a significant amount of the variation in the team detection ratio. The second step was to evaluate the combined effect of the variables shown to be significant in the initial analysis. Both analyses used standard nonparametric analysis methods (see [3] or [12]). Once these relationships were discovered and their magnitude estimated, other data, such as the gain ratios, was analyzed to confirm or reject (if possible) a causal relationship between the review methods and review performance.

4.3.6 Experimental Instrumentation

Several instruments were used in this experiment: three small software requirements specifications (SRS), instructions for each review method, and a data collection form.

Software Requirements Specifications. The SRS used describe three event-driven process control systems: an elevator control system (ELEVATOR), a water level monitoring system (WLMS), and an automobile cruise control system (CRUISE). All faults present in these SRS appear in the original documents or were generated during adaptation; no faults were intentionally seeded into the document. The experiments discovered 42 faults in the WLMS SRS and 26 in the CRUISE SRS. The number of faults in the ELEVATOR SRS is unknown since it was used only for training exercises. (These specifications were originally developed for another experiment. See Porter et al.[17] for details.)

4.3.7 Fault Reporting Forms

We also developed a Fault Report Form. Whenever a potential fault was discovered – during either the fault detection or the collection activities – an entry was made on this form. The entry included four kinds of information: Review Phase (First or Second), Fault Location (Page and Line Numbers), Fault Disposition (Faults can be True Faults or False Positives), and Fault Description (in prose).

4.3.8 Experimental Procedures

The participants were given two lectures of 75 minutes each on software requirements specifications, the SCR tabular requirements notation, review procedures, the fault classification scheme, and on how to fill out of data collection forms. The references for these lectures were Fagan [6], Parnas [16], and the IEEE Guide to Software Requirements Specifications [1]. The participants were then divided into three-person teams – (see Section 4.3.4 for details.) Within each team, members were randomly assigned to act as the moderator, the recorder, or the reader during the collection meeting.

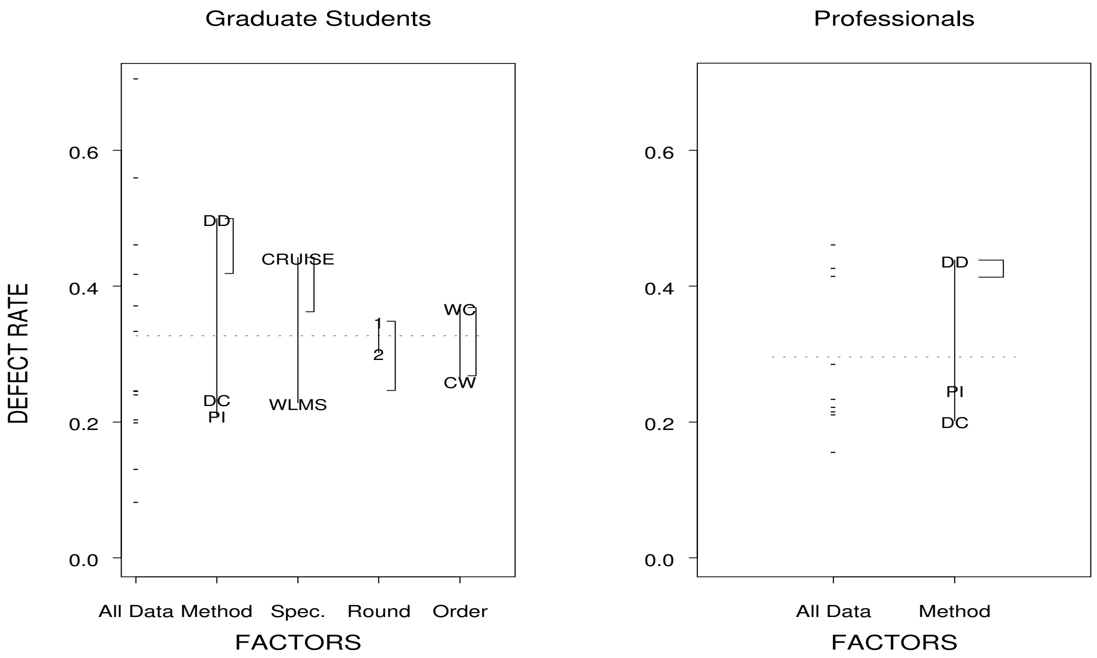

Figure 5: **Fault Detection Ratios by Independent Variable.** The dashes in the far left column of each panel shows each team's fault detection ratio. The horizontal line is the average fault detection ratio. The plot demonstrates the ability of each variable to explain variation in the fault detection ratios. For example, for the Method variable, the vertical location of DD symbol is determined by averaging the fault detection ratios for all teams reviewing with the DD method. The vertical bracket,], to the right of the variable shows one standard error of the difference between two settings of the variable.

Training Phase. For the training exercise, each team reviewed the ELEVATOR SRS. Individual team members read the specification and recorded all faults they found on a Fault Report Form. Their efforts were restricted to two hours. Later the experimenters met with the participants and answered questions about the experimental procedures. The ELEVATOR SRS was not used in the remainder of the experiment.

Experimental Phase. The experimental phase involved two review rounds for the graduate students and one for the professionals. The instruments used were the WLMS and CRUISE specifications discussed in Section 4.3.6, and the Fault Report Form.

During the first Round, three of the seven graduate student teams were asked to review the CRUISE specification; the remaining four teams reviewed the WLMS specification. Each review involved two phases that differed according to the method used. All the professional teams reviewed the WLMS in the first Round. The review methods used by each team are shown in Table 1.

The first phase for the Detection-Collection and Detection-Detection methods lasted up to 2.5 hours, and all potential faults were reported on the Fault Report Form. The first phase for the Preparation-Inspection method took the same amount of time, but the reviewers were not allowed to report faults or take any notes. After the first phase all materials were collected. For the student population 28, 2-hour time slots were set aside during which review tasks could be done. Participants performed each task within a single two-hour session and were not allowed to work at other times. For the professionals the review was conducted as part of a class session with each phase limited to 2 hours.

Once the team finished the first phase, the moderator arranged for the second phase which was also limited to 2.5 hours. This second phase involved a team meeting for the PI and DC methods during which the reader paraphrased each requirement, and the reviewers brought up any issues they had found earlier or had just discovered. The team recorder maintained the team's master Fault Report Form. The DD team did not hold a second meeting. Their second phase was exactly the same as the first – individual fault detection. The entire first Round was completed in one week. The second Round for the graduate students was similar to the first

Specification	Review Method		
	PI	**DC**	**DD**
WLMS	(.21) (.24) (.29) .36	(.14) (.21) (.21) .24	.33 .4 (.4) (.43) (.45) .55 .69
(average)	(.24) .36	(.21) .24	(.43) .46
Cruise	.08 .12 .19 .23	.19 .23	.46
(average)	.15	.22	.46

Figure 6: **Team Fault Detection Ratio.** This table shows the nominal and average fault detection ratios for all 16 teams (7 graduate student and 9 professional teams). There are only 22 observations, however, since one student team dropped out because of a team member's illness. Data from the professional population is enclosed in parentheses.

except that teams who had reviewed the WLMS during Round 1 review the CRUISE in Round 2 and vice versa.

4.3.9 Data Collection

Two sets of data are important to our study: the Individual Fault Summaries and the Team Fault Summaries.

An individual fault summary shows whether a reviewer discovered a particular fault. This data is gathered from the fault report forms the reviewers completed during fault detection.

A team fault summary shows whether a team discovered a particular fault. For the PI and DC methods this data is gathered from the fault report forms filled out at the collection meetings. For the DD method the team summary is just the set union of faults recorded by all reviewers. This data is used to assess the effectiveness of each fault detection method.

One problem with the team summaries is that the DC and PI methods involve meetings, but the DD method doesn't. Consequently, any meeting losses – faults discovered by individuals before the meeting that don't appear in the team fault report – will lower the fault detection ratio for these two methods but not, of course, for the DD method. However, since meeting losses average only about 5% for the DC method (meeting losses can't be measured them for the PI method) they were considered insignificant.

This experiment also compared the benefits of meetings (DC method) with the benefits of additional individual analysis (DD method). For the DC method meetings gains are calculated from the individual and the team summaries. For the DD method meeting gains are calculated by determining whether each fault was originally discovered during the first or the second detection activity.

4.4 Experimental Results

This analysis was done in three steps: (1) The team fault detection ratios were compared to ascertain whether the review methods had the same effectiveness. (2) The first and second round performance of individuals and teams were then compared to see how the different treatments performed. (3) Finally, individual fault summaries were analyzed to determine whether different treatments found different faults.

4.4.1 Analysis of Team Performance

If the hypothesis that reviews without meetings are no less effective than reviews with meetings is true, then the performances of the DD method should not be significantly lower than those of the other two methods. The analysis strategy was first to determine whether various threats to the experiment's internal validity could be seen and then to test our hypotheses.

The first analysis measures each independent variable's contribution to review performance. The Wilcoxon test showed that for the student population Review Method and Specification are significant, but the Round, Order, and Team Composition are not. Review Method was also significant for the professional population.

Figure 6 shows the input to this analysis. Six of the cells contain the average detection ratio for teams using each review method and specification (3 detection methods applied to 2 specifications).

Finally, the performances of each method were compared. Figures 5 and 6 summarize the defect detection data. As depicted, the DD review method resulted in the highest fault detection ratios (49% for graduate students, 43% for professionals). No difference could be detected between the DC review method (22% for

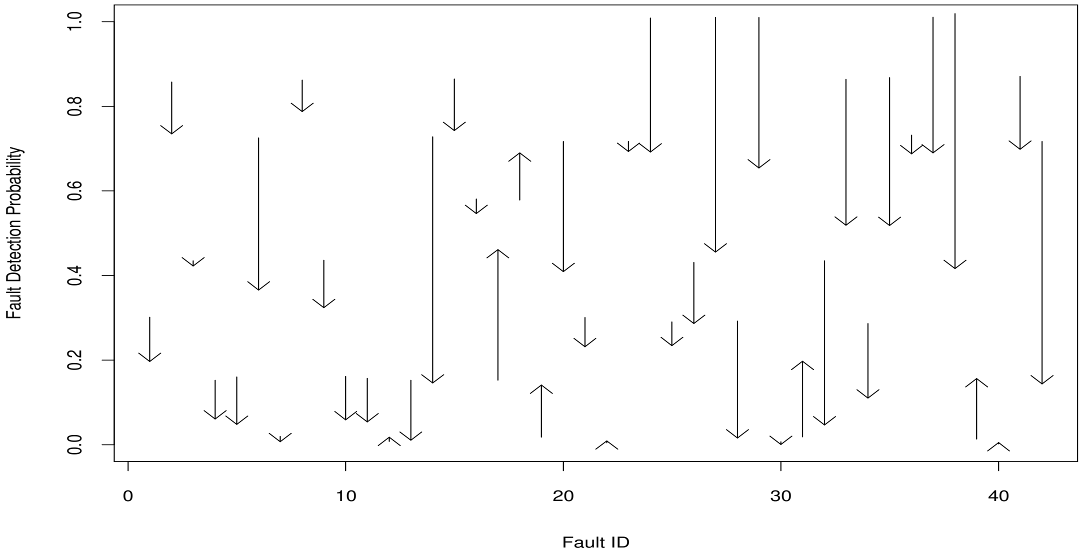

Figure 7: **Individual Defect Detection Probabilities for Reviews with and without Meetings.** Each defect is represented by an index along the horizontal axis. The with-meeting and without-meeting detection probabilities are represented by an arrow. That arrow's head indicates the with-meeting probability and its tail indicates the without-meeting probability. When a defects is found more often with a meeting than without, the arrow point up. The longer the arrow, the greater the difference between the two probabilities.

graduate students, 19% for professionals), and the PI detection method (19% for graduate students, 24% for professionals). These differences are consistent across both the graduate student and professional populations.

4.4.2 Analysis of Detection Ratios for Specific Faults

Even if reviews with meetings are less effective than reviews without meetings, team meetings might still be beneficial if they promote detection of classes of faults that individuals seldom or never find. If this hypothesis is true, then there should be a subset of faults for which the probability of their being found by the DC and PI methods is greater than by the DD method.

Figure 7 compares probability of detecting each WLMS defect when reviewing with meetings (DC and PI) and without (DD). The probability of detecting a defect with the DD method is defined as the number of teams that used the DD method and discovered the defect divided by the total number of teams that used the DD method. The detection probabilities for the other two methods are calculated similarly.

For each defect the following test is used to determine whether its detection probability is significantly higher when inspecting with a meeting than when inspecting without one.

Assuming that the detection of each defect follows a binomial distribution, we first estimate the the detection probability for reviews without meetings (i.e., p in the binomial distribution). The estimate is the percentage of DD groups that found the defect.

Next, the actual number of teams that detected the defect during a meeting-based review is calculated. Finally, a standard significance test is used to determine whether the number of teams that found the defect during a meeting-based review is significantly greater than that expected given p.

In this study only 5 faults in the WLMS had a higher detection probability with meetings than without. Of these, three were rarely detected with either method, one was frequently detected with both, and one (defect 17) was found with substantially greater frequency by reviews with meetings. Based on this data there appears to be insufficient evidence to support the existence of "meeting sensitive" faults.

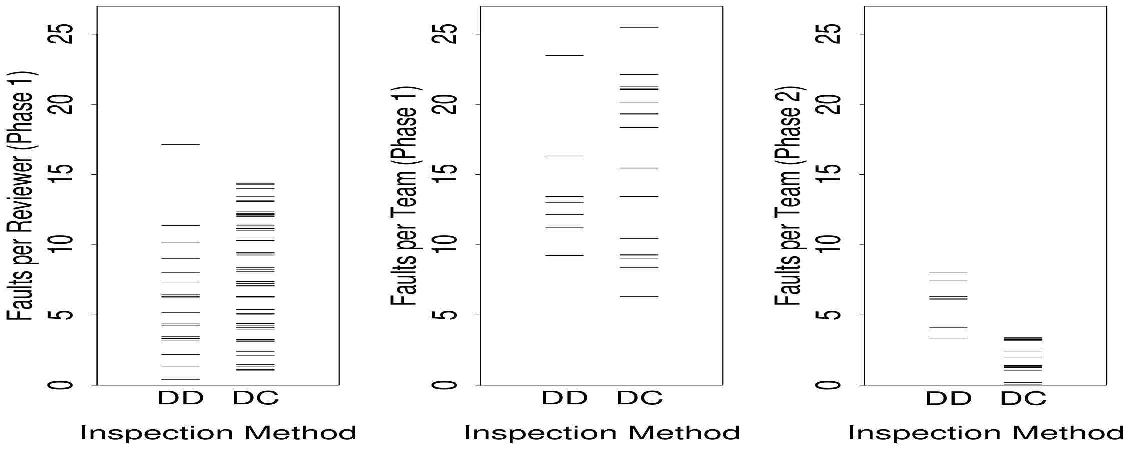

Figure 8: **Team and Individual Fault Detection Densities.** This figure shows the percentage of faults discovered by each individual during the first phase and by each team during the first and second phases of DD and DC review.

4.4.3 Analysis of Second Phase Performance

The previous analysis shows that in the current experiment, reviews without meetings seem more effective than reviews with meetings. This section examines whether the current data support the original hypothesis that meetingless reviews would be at least as effective because the benefits of having a meeting wouldn't outweigh the benefits of additional individual analysis.

If this hypothesis is true, then there should be little difference between the first phase performances of the DC and DD methods, but significant differences between the second phase performances. Therefore, the analysis strategy was to isolate the first and second phase performances to see how well they explain differences in total review performance.

This analysis is restricted to WLMS reviews only, and includes data from 14 WLMS reviews taken from two earlier review experiments [19]. These reviews involved both graduate student and professional populations and all used the DC method. This was done to increase our sample of DC reviews. This new data appears to be similar to that of the current experiment because the average defect detection ratio of the new reviews is 36% and that is nearly identical (34%) to the average for the current study. Furthermore, a Wilcoxon test was unable to reject the hypothesis that these two sets of data were drawn from the same distributions (p=.63).

Figure 8 contains a boxplot showing the number of faults found by each reviewer during the first phase of the DD and DC reviews. The ratios for both methods are statistically indistinguishable.

Figure 8 shows the number of unique faults discovered by each team during the first phase of a DD or DC review. This is the set union of faults found by all team members. Again the ratios appear to be similar (p=.61).

However, Figure 8 shows that the number of unique faults discovered during the second phase of a DD review are higher that those discovered during DC review (p=< .001).

Based on this analysis it appears that, in terms of number of defects found, the benefit of having a review meeting is less than that of performing additional individual analysis.

5 Comparative Analysis

As discussed in Section 2.2, one characteristic of empirical research is that the results of any single study must be viewed with suspicion. A single study's specific results become accepted over time through integration with additional experimental findings.

The simplest form of integration occurs when an experiment is replicated without modification. In this case, the structure of the studies is similar and the integration is relatively straightforward – however, mistakes or biases in the original design will permeate all the studies. Partial replication occurs in cases where an experimenter identifies a problem with the initial experimental design and modifies it before proceeding.

The integration described in this paper is performed on two studies that were designed and implemented independently. In this case, special attention must be paid to understanding the similarities and differences between the two studies so that their results can be compared. We call this process reconciliation. We used the following five step process to reconcile our experiments.

5.1 The Reconciliation Process

1. *Standardize independent variables.*

   The two experiments were initially designed with different independent variables. However, we found that that some of the UM independent variables could be made more similar to those in the UH study by holding out some data as well as regrouping some of the original UM independent variables.

2. *Standardize dependent variables.*

   The two experiments also initially differed with respect to the original dependent variables. Sometimes a dependent variable from one study was missing in the other. In some cases we could generate the missing dependent variable by re-analyzing the raw data. Other times, both experiments referred to the same dependent variable but defined it somewhat differently. In these cases, we attempted to reconcile the two studies by developing a common definition and then re-analyzing both sets of data.

3. *Develop common hypotheses.*

   Once we established a set of common independent and dependent variables, it was relatively straightforward to define a set of common hypotheses.

4. *Analyze data separately.*

   Using the same statistical techniques employed in the original studies, we then tested our new common hypotheses. Despite the common variables, we felt that the presence of significant differences in the artifacts inspected, the subject population, and the data collection instruments precluded the combination of the raw data into a single data set. Instead, we kept the two data sets separate and analyzed them independently.

5. *Compare results.*

   Based on the preceding analysis, we looked for similarities and differences in the results of the two experiments. When appropriate, we augmented these findings with information taken from the individual studies.

The next sections discuss the specific actions we took in each step of the reconciliation process.

5.2 Standardize Independent Variables

Both studies had a primary independent variable: review method. In the UH experiment, this variable had two values corresponding to two, one-phase review methods: EGSM (real groups) and EIAM (nominal groups). In the UM experiment, this variable had three values corresponding to three, two-phase review methods: detection followed by collection (DC), preparation followed by inspection (PI), and detection followed by more detection (DD).

We created a standard independent variable for both studies by re-characterizing the UM study in terms of the UH design's concept of real and nominal groups. As a result, the data from the first round of both the DD and DC methods maps to nominal groups, and the data from the second phase of the PI method maps to the real group.

5.3 Standardize Dependent Variables

The UH study originally measured dependent variables concerning total issues, total defects, effort, false positives, duplicates, and synergy. The original UM study measured total issues and defects in the same way as the UH study, but measured false positives differently and did not measure effort, duplicates, or synergy. In addition, the UM study measured gain rates (the percentage of new faults found during the second phase), which cannot occur in the UH experiment due to its one-phase design.

When we attempted to reconcile the false positive variable, we found that differences in the artifact type (code vs. requirements specification), made it hard to apply either of the original false positive definitions to the other experiment. Therefore, we reconciled this by redefining false positives in a way that would apply to both studies. This definition is “all non-true defect issues are false positives.”

The duplicates variable used in the UH study could be generated by reanalysis of the UM data. Finally, effort and synergy could not be reconstructed for the UM study, and gain rates could not be reconstructed for the UH study, so those variables are not considered in this comparative analysis.

5.4 Develop Common Hypotheses

Given the new common independent variable (review method, with two values: real groups and nominal groups) and the new common dependent variables (issues, defects, false positives, and duplicates), we generated the following common hypotheses. The combined hypotheses conform to the current “conventional wisdom”: that review meetings provide some sort of synergy or catalyst that significantly improves review outcome or its efficiency.

- C1: Real groups will find more significantly more defects than nominal groups.
- C2: Real groups will find significantly more issues than nominal groups.
- C3: Real groups will produce significantly less false positives than nominal groups.
- C4: Real groups will produce significantly less duplicates than nominal groups.
- C5: Real groups will find certain faults significantly more frequently than nominal groups.

5.5 Analyze Data Separately

Up to this point, the reconciliation process has followed the approach of meta-analysis. However, we felt that differences in the experimental design and instrumentation of the two studies prohibited the next step in meta-analysis, which would have been to combine the reconciled data values. Instead, we chose to test the hypotheses on each data set separately, and then present a ”side-by-side” comparison of the results.

Hypothesis C1: Real groups will find more significantly more defects than nominal groups. Figure 9 shows the observed defect density of real and nominal groups across both studies. In both studies the observed defect density for nominal groups was not statistically different from that of real groups. Therefore, we cannot reject the null hypothesis and cannot conclude that inspections with meetings find more defects than those without. As we mentioned earlier, this does not necessarily mean that there isn't any difference, but neither of these two studies can detect one.

Hypothesis C2: Real groups will find significantly more issues than nominal groups. The total number of issues raised by an inspection team is one measure of the number of questions or concerns noted by the reviewers. Since all defects were not found by any method, high numbers of issues might correlate to improved inspection effectiveness. Therefore, an important question is whether inspections with meetings raise more issues than inspections without them. Figure 10 shows the total number of issues raised by real and nominal groups across both studies. In both studies, nominal groups generated significantly more issues than real groups did. Therefore, we can reject hypothesis C2 and conclude instead that inspections without meetings appear to raise more issues than those with meetings.

Hypothesis C3: Real groups will produce significantly less false positives than nominal groups. Even if inspections with meetings do not find more defects or raise more issues than those without meetings, they may have other benefits. Hypothesis C3 tests whether meetings help to filter out incorrect issues (false positives). Since the author would otherwise have to resolve these issues, this filtering may be beneficial by reducing rework effort.

Figure 11 shows the proportion of false positive issues to total issues raised by nominal and real groups across both studies. We found that real groups had a significantly smaller percentage of false positives in both studies. Therefore, we confirm hypothesis C3 and conclude that inspections with meetings appear to do a better job at filtering out false positives.

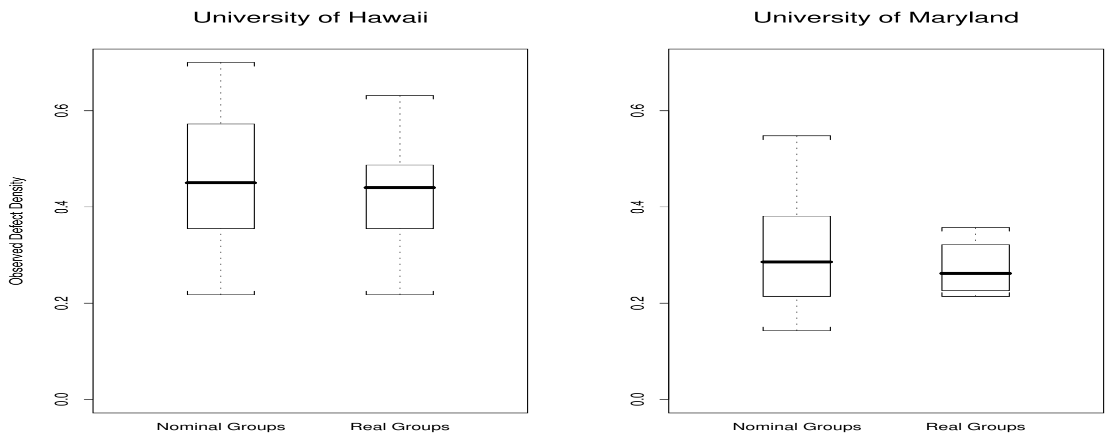

Figure 9: **Fault Detection Rates by Independent Variable.** This figure shows the performances of real and nominal groups across both studies. We found no statistically significant difference in either study (UH p=.34, UM p=.74).

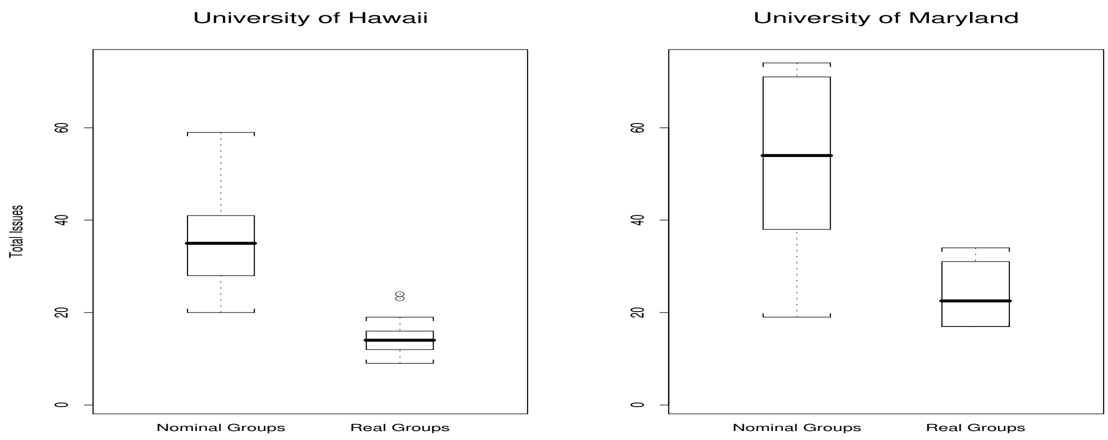

Figure 10: **Total Issues by Independent Variable.** This figure shows the number of issues raised by real and nominal groups across both studies. We found that nominal groups generated significantly more issues in both studies (UH p=< .001, UM p=.02).

Hypothesis C4: Real groups will produce significantly less duplicates than nominal groups. This hypothesis might seem tautological at first glance: since real groups cannot produce duplicates by definition, they must produce less duplicates than the nominal groups! However, this hypothesis is not guaranteed to be true: reviewers working independently in nominal groups might rarely overlap in the defects they discover. In this case, although real groups produce no duplicates by definition, the number of duplicates produced by nominal groups would be so few as to not differ significantly from real groups. Such a result would argue strongly in favor of nominal groups, since it would imply that little additional effort would be required in a nominal group-based process to weed out duplicates after combining the results.

However, the actual results show that nominal groups do produce significant numbers of duplicates: an average of 37% for the UH study, and 20% for the UM study. Thus, the hypothesis that real groups will produce significantly less duplicates than nominal groups (or, put another way, that nominal groups will produce significantly more than 0 duplicates) is confirmed.

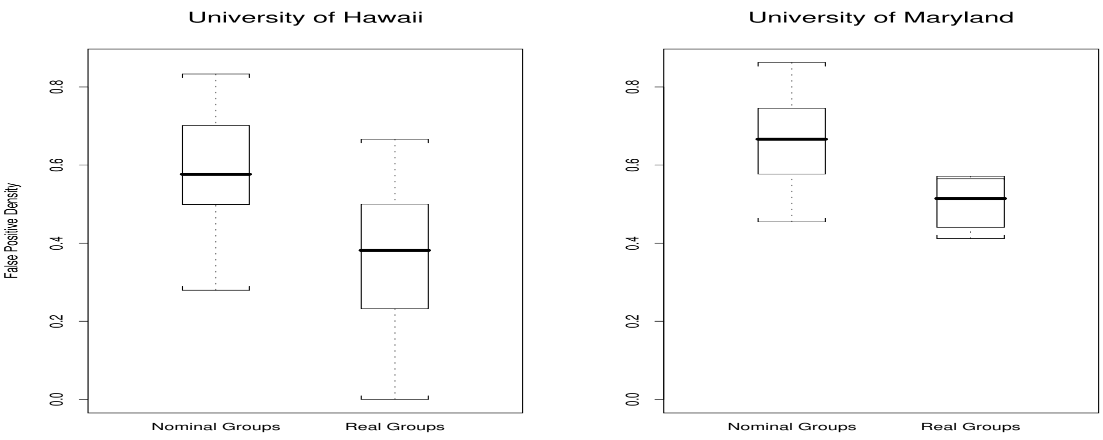

Figure 11: **False Positive Density by Independent Variable.** This figure shows the proportion of false positives to total issues for real and nominal groups across both studies. We found nominal groups generated significantly more false positives than real groups in both studies (UH p=< .001, UM p=.02).

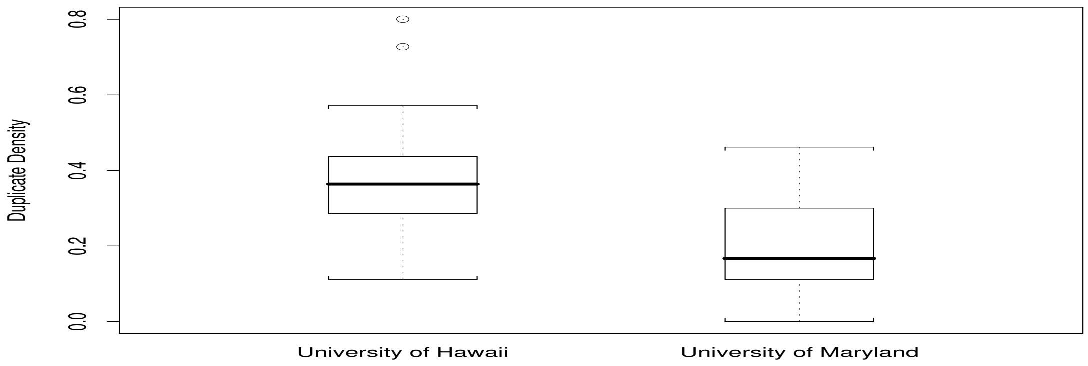

Figure 12: **Duplicate Density by Independent Variable.** This figure shows the proportion of defects found by two or more members of a nominal group to the total defects found by the group (real groups have no duplicates). We found that nominal groups in UH had significantly more duplicates than those in the UM study (p=< .001).

Hypothesis C5: Real groups will find certain faults significantly more frequently than nominal groups. As discussed in Section 4, even if inspections with meetings do not find more total defects those without meetings, the synergy of a meeting might potentially enable reviewers to find certain types of defects more easily than they could working alone. If these "meeting-sensitive" defects are quite expensive to detect otherwise, then this capability alone might justify the use of review meetings.

To examine this hypothesis, we compared the detection probability for each defect during inspections with and without meetings. Figure 13 shows the results for the UM and UH studies. In the UM study, 23 of 68 defects (33%) were found more frequently using real groups than nominal groups, but only 9 of these (13%) were found significantly more often using meetings. In the UH study, 35 of 87 defects (40%) were found more frequently with real groups, but only 12 of these (13%) were found significantly more often using meetings.

These findings confirm the hypothesis: in both studies, real groups did appear able to find a small number of faults more frequently than the nominal groups. However, these results should be viewed with some caution.

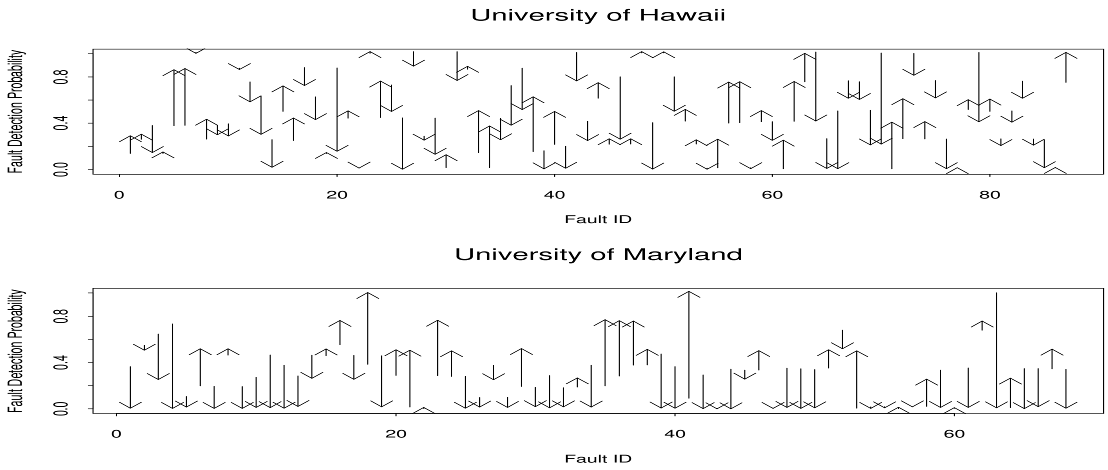

Figure 13: **Individual Defect Detection Probabilities for Inspections with and without Meetings.** This figure shows the differences in defect detection probability for all 87 defects in the UH study and for all 68 defects in the UM study. The interpretation of this diagram follows the explanation provided for Figure 7.

The significance test depends on the assumption that identifying each defect can be modeled as a Bernoulli trial and that our observations give us a reasonable estimate of the detection probability for each fault. Since both these assumptions are questionable, we use these results only as an indicator that some defects are found more or less often than others. It remains an open question as to whether classes of defects exist whose detection costs and importance alone justify the use of meetings.

5.6 Comparison of Results

The two studies, once reconciled and compared, reveal strikingly similar findings for all of the major hypotheses, as well as some interesting differences.

5.6.1 Defect detection rates

First, neither study showed that real groups find significantly more defects than nominal groups (and neither study showed the converse: that nominal groups find significantly more defects than real groups). Instead, Figure 9 shows that in each study, the two methods had very similar defect detection rates. This does not imply that the two methods are *exactly* equal with respect to their defect detection capabilities: it is quite possible that a well designed experiment with a sufficiently large number of trials could detect a statistically significant difference between the two methods. However, the results of these two studies cast strong doubt upon the prospect a *substantial* difference between inspections with and without meetings due to this factor alone. In other words, it is extremely unlikely that (for the defect detection mechanisms used in these studies) meeting-based review methods actually detect, for example, twice as many defects as nominal group methods (or vice-versa). Instead, it seems likely that the difference in effectiveness that can be attributed to meetings vs. non-meetings is quite small, and that other factors are far more important in determining the defect detection effectiveness of review meetings.

Figure 9 also reveals an interesting difference between the two studies: the average defect detection rate for the UH study was approximately 45%, while the average defect detection rate for the UM study was around 30%. We conjecture that this difference in magnitude results from two differences between the design of the two studies. In the UH study, students reviewed source code for the implementation of a program they had all just completed, and so they were intimately familiar with both the problem domain (Employee databases or Two pass assemblers) and the representation (C or C++). In the UM study, students reviewed two formal requirements specification documents. Compared to the UH students, the UM students were neither as familiar with the problem domain (automobile cruise control and water level management) nor the representation (SCR tabular requirements notation). Perhaps the greater defect detection rate observed in the UH study results from the increased familiarity of the UH students with the problem domain and representation.

Finally, the average defect detection rates for both studies fall substantially below the rates often provided for review techniques in the literature (such as 60-80% defect detection effectiveness). There are many possible explanations for this difference, but the existence of this difference suggests that more research on the actual defect detection effectiveness of review should be conducted.

5.6.2 True and False Issue Generation

Both studies found that nominal groups generated significantly more issues than real groups. In fact, Figure 10 shows that nominal groups generated about twice as many issues as real groups.

Since the defect detection effectiveness of real and nominal groups was almost equal, it follows that differences in issue generation were offset by inverse differences in false positive density. Figure 11 illustrates that in both the UM and UH studies nominal groups generated substantially more false positives than real groups.

Once again, however, the two studies differed in the magnitudes of these two variables. For example, the UH nominal groups generated an average of around 35 defects, while the UM nominal groups generated an average of over 50 defects. The UH real groups generated an average of around 15 defects, while the UM real groups averaged over 20. In other words, the UM groups generally outperformed the UH groups with respect to issue generation. This trend continues with respect to false positive generation, where the UM groups again "outperformed" the UH groups with higher levels of false positives.

We conjecture that these differences in magnitudes may be attributable to differences in the artifact type used in the two studies. The UM students reviewed a requirements document, for which the concept of a defect is more nebulous than for source code, the material reviewed by the UH students. Thus, the greater ambiguity in what constitutes a defect for requirements documents could explain why the UM students generated both more issues (it is easier to generate plausible defects when the nature of a defect can be more broadly construed) and more false positives (it is easier to generate non-defects, for exactly the same reason) than the UH students.

5.6.3 Duplicate Density

Both studies again agreed that nominal groups generate significant amounts of duplicate defects, as shown in Figure 12.

As might be expected by now, the two studies showed significant differences with respect to the magnitude of duplication. Interestingly, this is consistent with capture-recapture techniques [5] for estimating the number of defects remaining in a previously inspected artifact. That research says that, in general, the amount of overlap in N independent reviews will increase as the observed defect density increases. The teams in the UH study found a greater percentage of defects than did the teams in the UM study and, as predicted by the capture-recapture research, the amount of duplication increased as well.

5.6.4 Meeting-sensitive defects

Finally, both studies found instances of defects that were found significantly more frequently by real groups than nominal groups. Although this confirms the hypothesis, both studies only detected a small number of these defects, and the implications of this finding are unclear. These findings may be simply an experimental artifact. Even if they do represent true evidence of one or more classes of meeting-sensitive defects, then additional research needs to be performed to determine the precise nature of these classes. Preliminary examination of the meeting-sensitive defect instances observed in these studies do not immediately suggest the defining characteristics of such a meeting-sensitive defect class. The findings do, however, suggest an important direction for future research.

6 Conclusions

6.1 Concerning review meetings

The UH and UM experiments shared a common motivation: to assess the contributions of meetings to software review. However, their design and enactment differed in substantial ways. The UH subjects were junior-level undergraduates; the UM subjects were graduate students and professionals. The UH artifacts were source code; the UM artifacts were requirements documents. The UH environment was entirely on-line, using a state-of-the-art computer-supported cooperative work environment; the UM environment was entirely manual, using paper and pencil. The UH design used two treatments, while the UM design used three.

Despite these differences, we were able to find five hypotheses that could be tested on both data sets, and both data sets confirmed or denied the hypotheses in the same way. This striking convergence of findings leads us to the following two conjectures about the *typical* outcome of meetings in software review:

- With respect to *total* defect detection effectiveness, typical meeting-based software review methods are neither substantially more effective nor less effective than non-meeting-based software review alternatives.
- Individual defect detection typically generates far more issues than group-based defect detection, yet has the cost of higher false positive rates and (when review roles are not specialized) higher issue duplication.

In addition to these conjectures, both studies also found evidence that meeting-based review may be potentially useful for detecting certain classes of defects. We are far less confident of this finding than of the preceding two, because the precision of our analysis method is limited, by the number of data points we have.

While the comparative analysis of these two studies and their subsequent close agreement in findings suggests that our two conjectures are generally true in traditional situations, it is still quite possible that different outcomes would result in non-traditional situations. For example, the Cleanroom Development method has achieved extremely high defect removal rates through a rigorous, formal process, of which meeting-based review is an integral component. It is possible that removing review meetings in the context of Cleanroom development could have an unanticipated "ripple effect" through other aspects of the process, leading to a significant decrease in defect detection effectiveness. On the other hand, a geographically distributed development organization may find that non-meeting-based review substantially outperforms meeting-based review, since it enables greater participation and interaction between developers who are not physically co-located.

6.2 Concerning comparative analysis

We are highly satisfied with the comparative analysis process we designed and its results. The reconciliation process strikes a good balance between the limitations of a literature-based comparison and the constraints of full meta-analysis.

Unlike a literature-based comparison, reconciliation forced us to reanalyze the original raw data for both studies. The search for common variables and hypotheses led us to new analyses of both data sets that were not performed during the original experiments. These new analyses significantly strengthened the results beyond what would have been possible had we attempted to compare the studies based purely upon the original published findings.

Unlike full meta-analysis, reconciliation did not require us to ensure that our studies were so similar in design that merging of raw data was possible. In fact, our studies differed in so many ways that we are skeptical that a full meta-analysis could be applied successfully to these studies. Instead, reconciliation enabled us to perform a "side-by-side" comparison of the new hypotheses and results, and to observe clear similarities between the reconciled outcomes.

6.3 Concerning future research

This study suggests useful future directions for research in experimental software engineering in general, and formal technical review in particular.

First, we hope that this example of comparative analysis using the reconciliation approach will spur other experimental software engineering researchers to consider its application to their own research areas. Many experimental studies could be, but haven't been compared in this way. Since controlled experiments are time-consuming and expensive, comparative analysis may provide a way to obtain higher scientific "return-on-investment" from this valuable raw data.

Second, although the results from this comparative analysis increase our confidence in our two conjectures concerning typical review practice, we believe substantial value can still accrue from replication of this comparative analysis on prior or future review experiments. It is important to determine if these two conjectures continue to hold widely for industrial practitioners, and across other artifact types, and with other variations on meeting and non-meeting-based review methods. Such analyses will help us to precisely distinguish the "typical" from "atypical" review situations, and design and apply methods appropriately.

Third, this analysis suggests that reviewer specialization should be further investigated as a way to increase the effectiveness of (at least) non-meeting-based review methods. Review specialization can increase overall defect detection effectiveness by reducing the issue duplication rate. More research needs to be done concerning the most effective ways to specialize the tasks of reviewers for particular domains (See Porter et al. [18]).

Finally, this analysis provides no definitive answer to the tantalizing question of whether or not there are classes of defects that are predictably better found using meetings (or, conversely, better found *not* using meetings). The analysis does show that certain individual defects appeared to be found more readily in the specific meeting-based reviews of the specific artifacts of these studies. However, we were unable to abstract away from these isolated instances to a class of defects for which this property of meeting sensitivity might be expected to hold. Resolving this question more conclusively is an important task for the experimental review community. If we determine that some defects can be found significantly more easily using meetings than via other approaches, then we could design special-purpose review meetings for detection of just those issues, and use non-meeting-based approaches for others.

References

[1] *IEEE Guide to Software Requirements Specifications.* Soft. Eng. Tech. Comm. of the IEEE Computer Society, 1984. IEEE Std 830-1984.

[2] Mark A. Ardis. Lessons from using basic lotos. In *Sixteenth International Conference on Software Engineering*, pages 5–14, Sorrento, Italy, May 1994.

[3] G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters.* John Wiley & Sons, New York, 1978.

[4] 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.

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

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

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

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

[9] S. Gerhart, D. Craigen, and T. Ralston. Experience with formal methods in critical systems. *IEEE Software*, 11(1):21–28, January 1994.

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

[11] G.V. Glass, B. McGaw, and M.L Smith. *Meta-analysis in Social Research.* SAGE, Beverly Hills, CA, 1981.

[12] R. M. Heiberger. *Computation for the Analysis of Designed Experiments.* Wiley & Sons, New York, New York, 1989.

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

[14] 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.

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

[16] Dave L. Parnas and David M. Weiss. Active design reviews: principles and practices. In *Proceedings of the 8th International Conference on Software Engineering*, pages 215–222, Aug. 1985.

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

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

[19] Adam A. Porter and Lawrence G. Votta. An experiment to assess different defect detection methods for software requriements inspections. Technical Report CS-TR-3130, University of Maryland, College Park, MA, September 1993.

[20] Glen W. Russel. Experience with inspections in ultralarge-scale developments. *IEEE Software*, 8(1):25–31, January 1991.

[21] 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.

[22] Lawrence G. Votta. Does every inspection need a meeting? In *Proceedings of ACM SIGSOFT '93 Symposium on Foundations of Software Engineering*. Association for Computing Machinery, December 1993.