Evaluating Heuristics for Planning Effective and Efficient Inspections

Forrest J. Shull, Carolyn B. Seaman, Madeline M. Diep, Raimund L. Feldmann, Sara H. Godfrey, and Myrna Regardie

Abstract — A significant body of knowledge concerning software inspection practice indicates that the value of inspections varies widely both within and across organizations. Inspection effectiveness and efficiency can be measured in numerous ways, and may be affected by a variety of factors such as inspection planning, the type of software, the developing organization, and many others. In the early 1990's, NASA formulated heuristics for inspection planning based on best practices and early NASA inspection data. Over the intervening years, the body of data from NASA inspections has grown. This paper describes a multi-faceted exploratory analysis performed on this data to elicit lessons learned in general about conducting inspections and to recommend improvements to the existing heuristics. The contributions of our results include support for modifying some of the original inspection heuristics (e.g. increasing the recommended page rate), evidence that inspection planners must choose between efficiency and effectiveness, as a good tradeoff between them may not exist, and identification of small subsets of inspections for which new inspection heuristics are needed. Most importantly, this work illustrates the value of collecting rich data on software inspections, and using it to gain insight into, and improve, inspection practice.

Index Terms — D.2.5 a Code inspections and walkthroughs, D.2.8 Metrics/Measurement, D.2.8.c Process metrics, D.2.9.k Project control & modeling

1 Introduction

A long history of experience and experimentation has produced a significant body of knowledge concerning the proven effectiveness of software inspections. Data and experience from many years and many types of organizations have shown that a properly conducted inspection can remove between 60% and 90% of the existing defects [1]. It is well established that inspections in the earliest phases of software development yield the most savings by avoiding downstream rework.

However, the value of inspections varies widely both within and across organizations. Inspection effectiveness and efficiency can be measured in numerous ways (defects found, defects slipped to testing, time spent, defects found per unit of effort, etc.), and may be affected by a variety of factors, some related to inspection planning (e.g. number of inspectors) and others related to the software and the developing organization (e.g. application domain). The work described here is based on an analysis of a large body of data collected from inspections at NASA, the US governmental space agency. NASA heuristics for inspection planning were formulated in the early 1990's based on best practices and data from early NASA inspections. Over the intervening years, the body of data from NASA inspections has grown, and recently the authors of this paper were given the opportunity to analyze it to gain new insights into inspection effectiveness and efficiency.

The original set of heuristics for planning inspections was formulated by Dr. John Kelly at NASA's Jet Propulsion Laboratory (JPL), based on metrics collected across hundreds of inspections [2]. The heuristics focused on parameters known as the moderator's three control metrics, that is, the three parameters over which the inspection planner has direct influence. Modifying the values of these parameters is the mechanism by which an inspection moderator can affect the outcome of a given inspection. Kelly's research examined data from many inspections at NASA to formulate optimal ranges for these values and to help guide inspection planners. These values were incorporated into the JPL robust formal inspection training, which has been widely disseminated across all the NASA Centers.

The specific heuristics resulting from Kelly's research are:

- **Team size**, the number of participants involved in the inspection, should be between four and six people, regardless of the type of document being inspected. These values reflect the fact that teams of less than four people are likely to lack important perspectives, while larger teams

- Forrest Shull, Madeline Diep, Raimund Feldmann, and Myrna Regardie are all with the Fraunhofer USA Center for Experimental Software Engineering, College Park, MD 20740. E-mail: {fshull, mdiep, rfeldmann, mregardie}@fc-md.umd.edu.
- Carolyn Seaman is with the Fraunhofer Center, as well as the University of Maryland Baltimore County, Department of Information Systems, Baltimore, MD 21250. E-mail: cseaman@umbc.edu.
- Sara Godfrey is with NASA Goddard Space Flight Center, Greenbelt, MD 20771. E-mail: Sara.H.Godfrey@nasa.gov.

Manuscript received (insert date of submission if desired). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

are more likely to experience dynamics that limit full participation.

- **Meeting length** should be less than two hours, regardless of the type of document being inspected. If meetings stretch on longer than two hours, members' energy is likely to flag and the results are likely to be less than optimal. It is recommended that meetings end after two hours and then additional meetings should be scheduled if warranted.
- **Page rate,** the number of document pages that the inspectors examine per hour of the meeting, will depend on the type of document. Inspections of requirements documents should examine less than 15 pages per hour; design and test documents less than 20 pages per hour; and code documents less than ten pages per hour. These recommendations reflect the fact that giving a team too much material to look through will invariably result in a more superficial inspection.

Many things about software development have changed since that time. Languages, design notations, even the scale and type of problems tackled on NASA projects are very different from what they would have been in the early 1990s. Inspections themselves remain an important part of development processes at NASA. For example, software inspections are included in the mandatory NASA Procedural Requirements for Software Engineering (NPR 7150.2), issued by the Office of the Chief Engineer [3]. As a result, one focus of our work has been to examine whether the recommended ranges of parameters still apply to contemporary NASA projects.

Our first step in analyzing the inspection data was to attempt to validate the NASA inspection planning heuristics. This analysis and its results are described in more detail in section 2, but the overall finding confirms that the heuristics continue to be generally effective in most circumstances (i.e. on average, inspections that comply with the heuristics result in more defects found than those inspections that did not comply), but there was evidence that they could be improved upon. In particular, we found four potential weaknesses:

1. The heuristics only apply to maximizing the total number of defects found (i.e. effectiveness), and don't address other potential outcomes of interest (e.g. effort spent), in particular related to efficiency;
2. The heuristics represent a one-size-fits-all approach to inspection planning, with no refinement for different project situations;
3. The heuristics are not as universally applicable as one would hope, i.e. some modification could yield a stronger relationship between compliance and inspection effectiveness and/or efficiency;
4. Compliance with the heuristics seemed to be decreasing, i.e. the heuristics may be out of date in relation to what is realistic in the contemporary NASA development environment.

Based on these initial findings, we conducted a series of exploratory analyses to better understand the effect of inspection planning parameters on inspection effectiveness and efficiency, and to ultimately refine and improve the inspection planning heuristics. In particular, our analysis is guided by the following research questions:

Q1. What effect do inspection parameters have on other inspection outcomes, besides total number of defects found?
Q2. Are there variations in the heuristics that are appropriate for different situations?
Q3. Are there variations in the heuristics that would better ensure that compliance would result in better inspection outcomes (i.e. that would improve the relationship between compliance and effectiveness/efficiency)?

We begin, in section 2, by describing the initial analysis showing the effectiveness of the original heuristics. We then describe the data that our analysis is based on and our methodology in terms of the various exploratory data analyses we performed in order to gain understanding of the relationships between variables (in section 3) and the results of those analyses (in section 4). A discussion of these results is presented in section 5, followed by a discussion of related literature that addresses the factors influencing inspection effectiveness and efficiency in section 6. We summarize our conclusions in section 7.

2 Background – Inspections at NASA

We obtained a large set of data from a variety of software inspections across NASA. It includes 2,528 inspections of requirements, design, code, test plans, and a small number of other unspecified artifacts. These inspections come from 81 projects across five NASA Centers. These data were self-reported by developers as part of the inspection process and were not reported to management, so we have some confidence that they accurately represent inspection practice and were not manipulated for any purpose.

We describe here our initial analyses, which motivated the investigation described in the rest of the paper. We first divided the data into two sets: an "historical" dataset that covered the period of the early 1990s when the original heuristics were formulated, and a "contemporary" dataset that covered the time period since then (up until 2006). As the dividing line, we used January 1, 1995. This was a rather arbitrary choice but had the advantage of dividing the entire set into two roughly equal parts (1041 contemporary inspections and 1487 historical ones). It should be noted that the relative sparseness of the contemporary set (1041 inspections over 11 years, as compared to 1487 inspections over 6 years) is an artifact of the data collection process (i.e. our

dataset cannot be considered to be exhaustive), and is not evidence that the prevalence of inspections at NASA has decreased.

For each inspection in the dataset, we determined if it conformed to the recommended values of each inspection heuristic. For each heuristic, we compared the mean number of defects reported for inspections that complied with that heuristic (in-compliance inspections) to the mean for inspections that did not (out-of-compliance inspections). Since none of our variables were normally distributed, we used a non-parametric statistic, the Mann-Whitney test, to identify significant differences in the means. We summarize these results in Table 1, dividing them according to the dataset (contemporary vs. historical), and observe whether the heuristics might have become more or less effective over time For each dataset, we report how many inspections actually provided the data required for this analysis and the average number of defects found by inspections meeting that criteria.

Table 1 highlights several observations about the dataset. In the contemporary data, it is much less likely that teams followed the suggested heuristics for inspection team size and page rate. For example, out of 229 projects providing "team size" data, only 23 (10%) fell into the suggested range; and only about 15% of the projects reporting "page rate" data conformed to the heuristics. In contrast, for the historical dataset, regarding the "team size" data, the number of inspections that followed the heuristics and those that did not is much closer to 50/50; namely, 253 were in the recommended range and 239 were not.

Our results show that projects that followed the heuristics for team size detected more defects on average for both contemporary and historical datasets. In fact, the difference is actually more pronounced in the contemporary data. From the statistical analysis, the p-values shown in the rightmost column are much less than our chosen alpha-level of 0.05. This means that there is less than a 5% chance that the perceived difference is actually due to chance, rather than a real effect of the parameter.

We also observe that the meeting length heuristic has an effect counter to expectations. That is, in both the contemporary and historical datasets, inspections that conformed to the meeting length heuristic found fewer defects than those that exceeded the heuristic. This result is curious, but is probably affected at least in part by the fact that relatively very few inspections did not conform to this heuristic.

The results pertaining to the page rate heuristic are also worthy of note. They indicate that the page rate heuristic is not only less effective, but also more difficult to comply with, in the contemporary dataset than historically. Historically, about 24% of the inspections were able to comply with the page rate heuristic (as compared to 15% of the contemporary inspections), and those inspections found significantly more defects. The contemporary inspections that conformed to the page rate heuristic found marginally more defects (4.4 vs. 4.1), but the difference is not significant. This finding points to the need to update this heuristic.

Table 1. Testing original heuristics on historical and contemporary data.

	In-compliance Inspections		Out-of-compliance Inspections		Inspections following heuristics significantly better?
	# of inspections	Avg. # of resulting defects	# of inspections	Avg. # of resulting defects	
CONTEMPORARY DATASET (1995 and later)					
Team size	23	38.7	206	6.5	YES ($p<0.0005$)
Meeting length	184	3.7	7	27.6	NO
Page rate	23	4.4	134	4.1	NO ($p=0.5$)
HISTORICAL DATASET (1994 and earlier)					
Team size	253	11.7	239	7.3	YES ($p<0.0001$)
Meeting length	460	8.5	29	22.7	NO
Page rate	115	15.6	355	7.4	YES ($p<0.0001$)

Table 2. Variables used in the analysis.

Independent Variables	
Team size	Number of people attending the inspection meeting and/or serving as inspectors
Meeting length	Length of inspection meeting, in hours
Page rate	Number of document pages inspected, divided by the meeting length
Dependent Variables	
Number of defects found	Total number of defects reported as an outcome of the inspection
Total effort	Total effort, in person-hours, including meeting effort and preparation effort, but not including rework
Defects per page	Number of defects found divided by the number of document pages inspected
Defects per hour	Number of defects found divided by total effort in person-hours
Intervening Variables	
Application domain	Attitude, Orbit, Flight software
Software type	NASA-defined software type codes (A to H, or unspecified)
Project size	Small or medium
Product type	Requirements, design, code, test documents
Center	NASA Center at which the inspection took place

As mentioned in the introduction, the analysis presented in the rest of this paper is motivated by our observations from the analysis presented above. In particular, it is clear that the inspection planning heuristics could be more uniformly effective for all inspection parameters. In addition, from Table 1, we can see that contemporary inspections were less likely to comply with the heuristics than older (historical) inspections. This may indicate that, due to changes in the development environment, the heuristics are becoming harder to apply, and projects are more often violating them.

3 Methodology

The methodology for our work was exploratory. We started with the research questions above, probed the data in various ways, and allowed the results of each analysis to guide the types of analysis to come next. We used both visual and statistical methods to gain insight into relationships between variables that might be interesting and/or to confirm or determine the strength of relationships we suspected might exist. Thus, we did not have a pre-defined sequence of steps that we followed for data analysis. Instead, in this section, we present the details of the dataset we used, including all the variables we investigated, and an overview of the variety of analysis techniques we employed throughout the paper. The sequences of analyses used are detailed in section 4, along with their findings.

3.1 Variables

The dataset we obtained has numerous fields, which we have designated for our purposes as independent, dependent, or intervening variables. These variables are listed in Table 2 and are described in the subsections below.

3.1.1 Independent variables

The independent variables for our analyses consist of the three inspection control metrics, i.e. team size (number of participants), meeting length, and page rate.

The meeting length is reported in hours. An anomaly that we noticed during inspection of the data in preparation for analysis (but after the analysis presented in section 2) was that a number of records reported very large meeting lengths, as high as 80 hours. In conversations with some of our contacts from whom our data were donated, we learned that some of these very long meetings were actually inspections that spanned multiple meetings, but for reporting purposes, the meeting lengths were summed. Since the meeting length heuristic is concerned with the length of contiguous time that the inspection team meets, using the summed meeting length data did not make sense. For our analyses, we ignored inspections for which the reported meeting length was greater than 4 hours. We chose 4 hours as a reasonable limit on the length that a single contiguous meeting was likely to last. Also, it constituted a logical break in the data where only 194 records (~8% of the total data set) were eliminated.

Page rate was a derived measure that normally did not appear in the raw data we received from Centers. Since some inspections reported the size of the inspected artifact in lines of code (LOC), and others in pages, we used a scaling factor of 30 LOC per page to convert between the two (this is the standard conversion factor used in planning inspections at NASA [2]). We then used the page measure (raw or derived) to calculate the page rate by dividing it by the meeting length.

3.1.2 Dependent variables

The dependent variables for our study corresponded to the inspection outcomes, i.e. those attributes that describe how successful the inspection was. Nearly all the inspections in our dataset reported the total number of defects found, which we designated as one of our dependent variables. However, in order to get a clearer picture of our analysis results, we also looked at various normalized outcome variables. In particular, we normalized the defect count for each inspection by number of pages inspected ("defects per page") and by total inspection effort ("defects per hour"). We also used total inspection effort in person-hours as a dependent variable.

3.1.3 Intervening variables

There is a very large number of attributes of inspections that could potentially affect the inspection outcome. Many, but not all, of these were represented in our dataset. There were several important attributes for which there was enough variety in the data that we could treat them as intervening variables in the analysis. For each of these variables, we partitioned the data into subsets based on the attribute values.

For application domain, we identified subsets of the inspection data that included inspections of artifacts from projects in the same domain. The domains for which we had sufficient data for analysis were attitude, orbit, and flight software. These domains are sub-domains of satellite control software, which is the larger domain in which NASA is working. Another way to categorize domain is through the NASA set of software type "codes", designated A through H. For example, type A software is software that controls spacecraft carrying humans. In our dataset, the only substantial subset based on these categories was type C software, which is software that supports non-life-critical aspects of a mission, e.g. processing of scientific data from instruments. Inspection records outside this subset were either of software of another category, or for which no category was reported. Finally, project size was determined by creating categories of projects based on total size in LOC (small, medium, etc.) based on the distribution of project size in the dataset (small was defined as 10-100KLOC, etc.). The only subsets of sufficient size for analysis were small projects and medium projects.

In addition to these intervening variables, subsets of the inspection data were also defined based on the work artifact being inspected (e.g. requirements, design documents, code, test documents) and on the NASA Center in which the project was performed. The NASA Center variable is important because it is a proxy for cultural and historical differences between Centers, as well as differences in inspection processes. While all Centers must adhere to a general agency-wide inspection process, there is considerable leeway in the details for Centers to tailor the process. These two intervening variables were available for all inspections in the dataset.

Not all of the inspections in the dataset had recorded values for all of the variables outlined above. Hence, we conducted our analyses separately to use the maximum possible amount of data in each analysis. For this reason, the number of data points reported in section 4 varies from one analysis to another.

3.2 Data analyses

As explained earlier, our analysis was exploratory in nature, so we did not follow a defined sequence of steps. In section 4, as we present results we will also report the tests and techniques that were used to generate each finding. In this section, we provide an overview of how the different tests were performed.

Comparison of means tests: Some of our analyses were similar to the analysis that motivated this investigation, described in section 2. For each inspection heuristic, we divided the data into an in-compliance set and an out-of-compliance set. We then performed a Mann-Whitney test to see if the outcomes of the two sets of inspections were significantly different. To address Q1, we expanded the set of outcome variables used, to include total effort expended, defects per page and defects per hour, in addition to the number of defects found (as in section 2). We were also interested in fine-tuning the inspection heuristics for particular contexts (Q2).For this purpose we performed similar analyses, except with subsets of the data, partitioned by values of various intervening variables. For some of the subsets and heuristics, the divisions between in-compliance and out-of-compliance inspections were too unbalanced to do a meaningful comparison, i.e. either nearly all inspections in the subset were in compliance, or nearly all were out of compliance. These cases had to be ignored during this analysis, but they were noted as examples of situations where the heuristics were either very easy or very difficult to comply with. We also used the Mann-Whitney test to check for significant differences between subsets of the data based on values of dependent variables (e.g. to characterize highly effective inspections) in an effort to identify better thresholds for the heuristics.

Visualizations: In order to determine if there were any obvious "natural" thresholds in the data that might serve as improved thresholds for the inspection heuristics (e.g. the most effective range for number of participants), we used scatter plots. The scatter plots were examined to see if, in any cases, it was visually possible to identify any ranges of independent variables associated with optimal values of the dependent variables.

Another visualization we found useful were boxplots, which show, within a given subset of the data, the distribution (including the mean, median, quartiles, and outliers) of a particular variable. We used boxplots to investigate possible relationships between variables. To do this, we segmented the data into subsets based on one variable, then generated boxplots

using another variable in each subset. A side by side comparison of these boxplots was used to intuit how different the distributions were, and thus the possibility of a relationship between the two variables.

Regression trees: Regression trees are one approach to modeling the relationship between inspection parameters and outcomes. Regression tree modeling applies an iterative partitioning algorithm to a set of data, resulting in a tree-like structure, where each node represents a subset of the data conforming to a conjunction of conditions based on the independent variables in the set. The conditions are chosen such that each resulting subset is as homogeneous as possible with respect to a chosen dependent variable. The "quality" of a regression tree (i.e. how useful it is in characterizing a dataset) is normally assessed through two metrics. The first is the correlation coefficient between the actual values of the dependent variable and the values predicted by the tree. The second is the relative error, which indicates how much of the dataset is correctly predicted using the tree. Regression trees give insight into those independent and intervening variables most likely to have an effect on the value of the dependent variable, and which combinations of the variables are likely to be significant within different subsets of the data. We used regression tress primarily as an investigative tool to help us narrow down our large set of independent variables and focus on those most likely to be significant.

Tests of correlation: We calculated correlations for all combinations of independent and dependent variables in Table 2, for the whole dataset and for all subsets (based on values of intervening variables) large enough to support the analysis. Because of a lack of normality, the non-parametric test, Spearman's rho, was used.

4 FINDINGS

To reiterate, our specific research questions for this analysis are:

- Q1. What effect do inspection parameters have on other inspection outcomes, besides total number of defects found?
- Q2. Are there variations in the heuristics that are appropriate for different situations?
- Q3. Are there variations in the heuristics that would better ensure that compliance would result in better inspection outcomes (i.e. that would improve the relationship between compliance and effectiveness/efficiency)?

As a baseline, we began with an overview of how the heuristics perform in the dataset as a whole. These results are shown in Table 3. Again, we used the Mann-Whitney test to compare the means between in-compliance and out-of-compliance inspections. These results are mostly consistent with those in Table 1, so from this point we will no longer distinguish between historical and contemporary inspections. However, it should be noted that older inspections, in general, consumed more resources and found more defects, but were less efficient, than more recent inspections.

Table 3. Testing original inspection heuristics on entire dataset.

COMBINED DATASET (both historical and contemporary)					
	In-compliance inspections		Out-of-compliance inspections		Inspections following heuristics significantly better?
	# of inspections	Avg. # of resulting defects	# of inspections	Avg. # of resulting defects	
Team size	276	14.0	445	7.0	YES ($p<0.0005$)
Meeting length	644	7.1	36	23.6	NO
Page rate	138	13.8	489	6.5	YES ($p<0.0005$)

Table 4. Testing original inspection heuristics with respect to different dependent variables.

COMBINED DATASET (effect of heuristics on four outcome variables)				
	Do inspections conforming to the heuristics perform better or worse than those that don't with respect to:			
	Total effort?	Total defects?	Defects per page?	Defects per hour?
Team size	worse	better	worse	worse
Meeting length	better	worse	worse	better
Page rate	worse	better	better	worse

Next, we repeated this analysis for all the dependent variables. The results (comparison of means, shown in Table 4) show a much more complicated picture than that presented by the results on total defects alone (from Table 3). Columns 2 and 3 of Table 4 show that compliance to team size and page rate heuristics result in more effective but more expensive inspections. The heuristic for meeting length has an opposite effect. When one examines the outcomes of an inspection more closely (i.e. by normalizing the number of defects found, as in the last two columns of Table 4), the picture becomes yet more cloudy. For example, what appears to be a benefit of complying with the team size heuristic (more total defects found) evaporates when we consider defects per page or defects per hour.

The results from this initial analysis imply that optimizing the effectiveness of an inspection (i.e. maximizing the number of defects found) is at odds with optimizing its efficiency (i.e. minimizing total effort spent). This led us to focus on one of our dependent variables, defects per hour (total number of defects found divided by total effort spent), which reflects the tradeoff between defect detection and effort. Maximizing defects per hour would likely be a concern of many inspections, particularly in projects where managers want to get the most out of the resources they consume, and where there is not a willingness to spend a premium to ensure that every single defect is found. However, in other types of projects, particularly safety- or mission-critical software, the concern may be in fact to maximize the number of defects found, despite the cost. Thus, in the presentation below, we have gathered two sets of evidence that our analysis yielded. First, we present findings that lead to guidance for projects concerned with maximizing the number of defects found (i.e. effectiveness). Then, in section 4.2, we present findings that point to heuristics for maximizing defects per hour (i.e. efficiency).

4.1 Maximizing total defects found

Table 3 shows that more defects are found in inspections that are in conformance with the heuristics related to team size and page rate, than in those out of conformance. In an effort to understand this phenomenon better, we did a deeper analysis of the inspection dataset from the perspective of maximizing the number of defects found in an inspection. This led to the following finding:

Finding 1: Inspections in which large numbers of defects were reported involved higher levels of effort and more participants.

While rather straightforward and seemingly obvious, this finding implies that neither higher levels of effort nor more participants alone will result in higher numbers of defects. It also implies that there is no "shortcut" to finding more defects other than working hard at it. It should be noted that similar findings were also found with respect to the number of defects found per page, although for simplicity we focus simply on total defects found in this discussion. This finding is supported principally by tests of correlation, but visual examinations of distributions, regression trees, and examination of outliers also contributed to understanding. Significant correlations were found between the number of defects found and all variables tested (number of inspection participants, total effort, meeting effort, and preparation effort – more about page rate later). Correlation coefficients ranged from 0.49 (for number of participants) to 0.79 (for total effort). Scatter plots showed little evidence of an upper bound on the benefits of adding resources (time and people) to inspections. For example, Figure 1 shows a scatter plot relating the mean number of defects found and the meeting length of an inspection (i.e. each dot corresponds to the mean number of defects found for all inspections having the corresponding meeting length). There were only two inspections with meeting length of 3.5 hours, so the point corresponding to that value can be considered an anomaly and ignored for the moment. The scatter plot shows an increasing relationship that does not level off, at least within the range of this dataset.

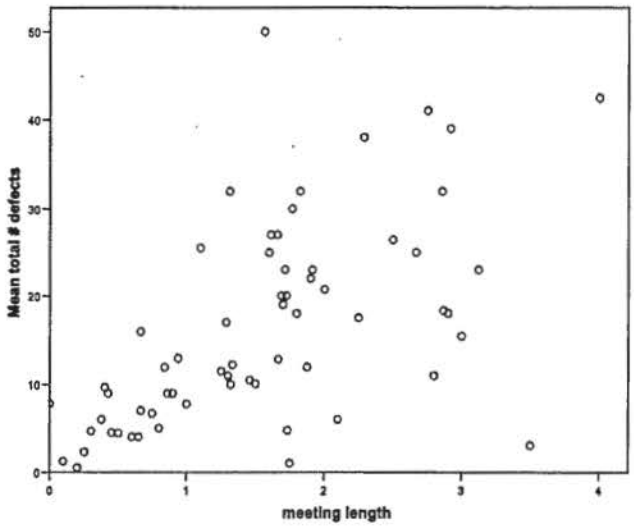

Figure 1. Scatterplot relating meeting length and number of defects found (each dot represents a mean over all inspections with the same meeting length).

We generated a number of regression trees using various combinations of independent variables and number of defects found as the dependent variable, but few of them were very strong (most correlation coefficients were < 0.5, and all had a relative error >= 50%). However, they did provide some insight into the independent variables most likely to affect total defects found. Since regression tree analysis is very sensitive to outliers, we removed 6 records with total numbers of defects found that were suspiciously high (all were above 150). Removing these outliers resulted in stronger and more informative trees. One (correlation coefficient .61, relative error 65%) used only the three inspection parameters (meeting length, team size, and page rate) as independent variables, while the second (correlation coefficient .74, relative error 56%) used all independent and intervening variables except the NASA Center at which the inspection took place (the Center dominated all other independent variables when it was included in the analysis). In both cases, the dominant independent variable distinguishing between inspections finding low vs. high numbers of defects was meeting length, which supports Finding 1 because meeting length is a major component of effort.

Figure 2 shows the distribution of total defects found, including the 6 "outliers" that were excluded from the regression tree analysis. The plot shows a large number of inspections with relatively very high numbers of defects found. We compared the set of 55 inspections with numbers of defects found more than one standard deviation above the mean to the remaining inspections in terms of average team size, meeting length, and other dependent variables. We found that the high-defect inspections were higher effort (in terms of meeting effort, preparation effort, and total effort), involved more participants, and found more defects per page than the rest of the inspections. All of these comparisons were significant ($p<.05$) using the Mann-Whitney test. Again, this supports the conclusion that the secret to finding more defects is not counter-intuitive, but consists of using more people and more time.

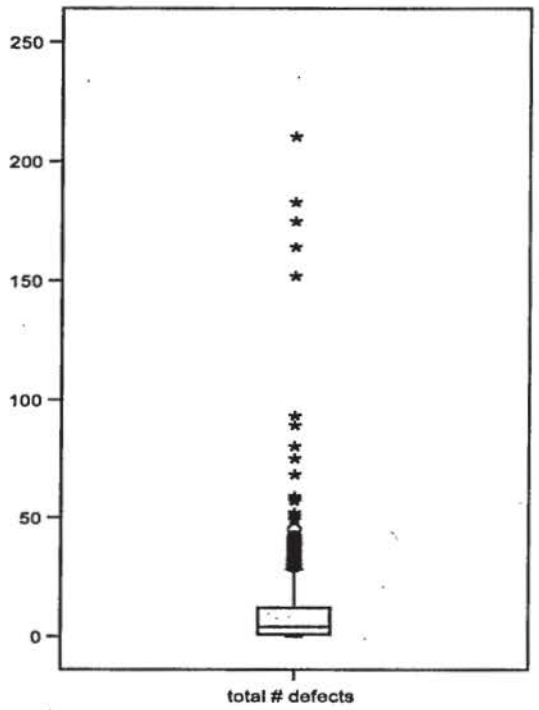

Figure 2. Distribution of number of defects found. The 25th percentile, median, and 75th percentile of the data are represented as horizontal lines.

The relationship between page rate and inspection effectiveness deserves a bit more examination. The current heuristics give different thresholds for the optimal page rate for different types of inspection artifacts (code, design, etc.). However, as can be seen from Table 1, the percentage of inspections conforming to the page rate heuristic has always been small, especially for the contemporary dataset. Moreover, conforming to the page rate heuristic results in more defects found only in the historical inspections, not the contemporary. This implies that the original page rate heuristic is more difficult to adhere to, and less useful, than it was previously. This leads us to investigate how the page rate heuristic might be relaxed and what the effects of doing so would be.

Table 5. Variations in numbers of defects found in code inspections as page rate increases.

CODE INSPECTIONS			
# of Inspections	Page rate less than	Average # of defects found	Average # of defects/page (effectiveness decrease)
28	10	4.3	4.45
77	20	4.1	1.81(59%)
145	40	4.9	1.1(75%)
258	80	4.6	0.7(84%)
368 (100%)	2667	4.42	0.57(87%)

As a starting point, we calculated the average number of defects found, and defects found per page, for code inspections with page rates under various thresholds. The results are shown in Table 5. They show that, in fact, increasing the page rate does not have a severe impact on total defects found. However, there is a severe penalty associated with the number of defects found per page. There is a significant correlation ($p < .05$) between page rate and defects per page for code inspections. From Table 5, we see that relaxing the page rate heuristic to 20 pages/hour (as opposed to the recommended 10 pages/hour) results in an almost 60% reduction in the number of defects found per page, and there is a 75% penalty when relaxing the page rate to 40 pages/hour. Further, the vast majority of code inspections are well above this threshold.

Similar results were found for design and test inspections. For example, relaxing the page rate heuristic to 40 pages/hour (as compared to the recommended 20 pages/hour) for design inspections results in a 28% reduction in defects found per page, and a 40% reduction for a page rate up to 80 pages/hour. For test inspections, relaxing the heuristic to 40 pages/hour (from the recommended 20) results in a 40% hit in defects found per page. There were very few requirements inspections in the dataset with data on both defects and page rate, and these did not represent much variation in page rate, so this analysis was not very meaningful for requirements inspections.

If we assume that the inspections in which the page rate was higher did not, in general, have lower true defect density (an unsubstantiated but not unreasonable assumption), then these results imply that higher page rates are associated with missed defects. This is consistent with the justification for the original page rate heuristics, i.e. that higher page rates will result in a more superficial inspection. This finding was hinted at earlier, in fact, in the analysis presented in Table 4, which showed that, overall, inspections that conform to the page rate heuristic do in fact perform better than those that do not in terms of defects found per page. However, this leads to a dilemma. Table 1 shows that violating the page rate heuristic is common, and is becoming more common. Clearly, it is tempting, given schedule and budget pressures, to speed up the inspection process by increasing page rate, when it appears that developers can handle the increased amounts of material. This analysis shows, however, that there is a hefty price being paid for this.

Finding 2: Inspections of all types (except possibly requirements) would be significantly more effective (in terms of defects found per page) if they adhered to the original page rate heuristics.

4.2 Maximizing defects per hour

Another relevant measure when evaluating inspections is the efficiency of the inspection process, one indicator of which is the number of defects found in the inspection per person-hour of total effort. Under a variety of conditions (e.g. limited resources, software of low criticality, heavy reliance on testing, etc.), maximizing the defects found per hour may be a more important inspection goal than finding the maximum number of defects. With this in mind, we examined our dataset for insights into potential heuristics for maximizing the dependent variable "defects per hour."

Our first finding in this section paints a broad picture of defects per hour among the inspections in our dataset. It arises from the boxplot in Figure 3, which shows the distribution of defects per hour over the entire dataset, as well as descriptive statistics.

Finding 3: With a few exceptions (treated in later findings), the defects per hour for most inspections remains within a narrow range of 1-2 defects per hour of total effort.

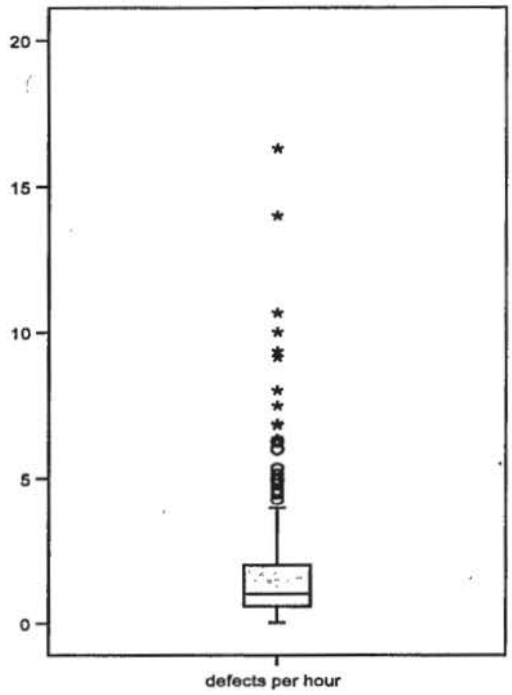

Figure 3. Distribution of defects per hour over entire dataset.

While the range of values for defects per hour is over 16, both the mean and the median are less than 2. In fact, 75% of the values are less than 2. However, the clear presence of significant outliers in Figure 3 led us to further investigate this small collection of "super inspections" that resulted in abnormally high numbers of defects per hour. We defined a subset of our inspection data, which we called "hyper DpH inspections", to include all inspections with defects per hour greater than one standard deviation above the mean for defects per hour over all inspections, giving a threshold of about 3.4 defects per hour. The hyper DpH inspection set consisted of 52 inspections, with 496 inspections constituting the set of inspections with "normal" defects per hour values.

We conducted a number of exploratory analyses on the hyper DpH inspection set to both characterize it and compare it to the rest of the dataset. From this analysis, we concluded that the hyper DpH inspections had both lower total effort and higher numbers of defects found than other inspections. Further, the lower levels of effort were due to lower numbers of participants, shorter meeting times, and shorter preparation times. Seventy-five percent of the hyper DpH inspections had between 1 and 3 participants, and a meeting length between .25 and .5 hours. Hyper DpH inspections also tended to be more recent than other inspections, part of small projects, and were mostly (but not exclusively) code inspections.

This analysis leads to the following finding:

Finding 4: For code inspections on small projects, defects per hour is maximized when the number of participants is between 1 and 3, and the meeting length is between .25 and .5 hours.

This finding is meaningful in this context because inspection experts at NASA have been struggling with how to streamline the inspection process for small projects. Full inspections that conform to the inspection heuristics seem to be overkill for such projects, which can't afford the high levels of effort. Finding 4 suggests that there is in fact a streamlined, but still efficient, approach.

Finding 4 was tested, using a Mann-Whitney test limited to contemporary code inspections from small projects, comparing values of defects per hour between those inspections that fell within the thresholds specified in Finding 4 and those that did not (ignoring, for the moment, page rate, which will be discussed later). The results show that code inspections on small projects that are described by Finding 4 (1-3 participants and .25 - .5 hours meeting length) have higher numbers of defects per hour than those that don't (2.8 vs. 1.8). However, this finding was not statistically significant. The relevant dataset in this case is very small; only 23 code inspections from small projects had defect and effort data, 16 of which conformed to Finding 4.

We then tested the relationship between compliance with the heuristics and defects per hour only on the "normal" inspections (i.e. excluding the hyper DpH set). The results agrees with the results from the analysis against the entire dataset (as presented in Table 4), where out-of-compliance inspections (with the exception of the meeting length heuristic) outperform in-compliance inspections. Thus, the results in Table 4 are not an artifact of the hyper DpH inspections.

In an effort to gain further insight about what maximizes defects per hour, we generated several regression trees, using defects per hour as the dependent variable, and various combinations of independent variables. Although none of the trees generated were very good (all had correlation coefficients <= 0.5, and relative error >= 78%), they did provide some insight into the independent variables most likely to affect defects per hour. When using all independent and intervening variables in the tree, the most significant variable was the NASA Center at which the inspection took place. This implies that the factors affecting defects per hour are different for each Center, thus making it hard to generalize. It may also point to undocumented differences in inspection processes between Centers. When the Center variable was removed from the analysis, the resulting tree showed that meeting length and software type were the most influential variables. The role of meeting length is obvious, as it is involved in the calculation of effort. Looking more closely at the role of software type (e.g. attitude, orbit software, etc.), we compared the mean defects per hour in the "normal" inspections of each software type. We found that only inspections of software classified as "other" had significantly higher values for defects per hour than inspections of any other software types. This

result is not very informative, except that it indicates that application domain may play some role in maximizing defects per hour, even within the range specified in Finding 3.

We next turned our attention to the page rate heuristic and its relationship to efficiency (i.e. defects per hour). As mentioned previously, the current heuristics specify different page rate thresholds for different types of inspected artifacts (code, design, etc.). The hyper DpH inspections (which were mostly code inspections) had lower page rates on average than other inspections, but the median is similar to the "normal" inspections. The difference in means is not significant; both groups have very large standard deviations for page rate, and the box plot for this data (see Figure 4) shows that the difference in means is due primarily to a number of inspections in the normal DpH set with extremely high page rates. Most of the inspections with very high page rates did not result in a high number of defects per hour (i.e. they are in the normal DpH set). This implies that the page rate has little effect on whether an inspection has a high number of defects per hour or not, except in the case where the page rate is exceptionally high. A Spearman test for correlation between page rate and defects per hour for code inspections was not significant. The same is true for requirements and test inspections, but there was a significant but weak (coefficient of .255) correlation for design inspections.

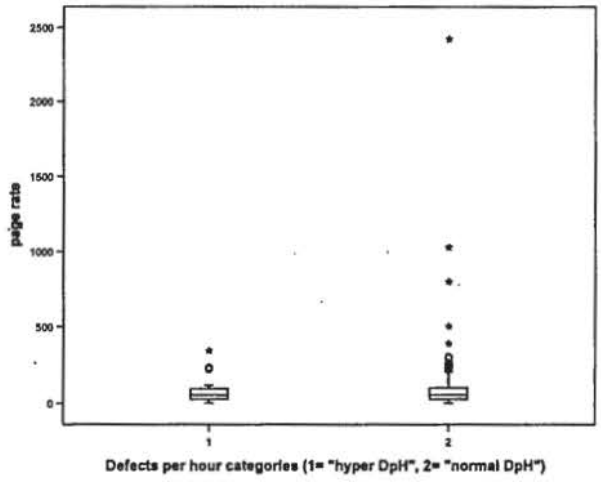

Figure 4: Comparison of page rates in "hyper DpH" inspections and other code inspections.

We also compared inspections of each type conforming to different page rate thresholds to see if they behaved differently in terms of defects found per hour (similar to the analysis shown in Table 5 using total defects found and defects found per page). This analysis shows no effect of page rate for test or requirements inspections (of which there were too few for the analysis to be meaningful). However, a pattern did emerge for code inspections. The analysis is shown in Table 6, where we see that, with the exception of a few inspections with page rates less than 10 (which happens to be the original page rate heuristic threshold), efficiency increases as page rate increases. That is, when more pages/hour are inspected, more defects are found per hour. However, the increases are not large, and recall that the correlation between page rate and defects per hour, for code inspections, is not significant.

Table 6. Variations in numbers of defects found per hour in code inspections as page rate increases.

CODE INSPECTIONS		
This many inspections:	**had a page rate less than:**	**and found, on average, this many defects per hour:**
13	10	1.89
23	15	1.32
35	20	1.36
82	40	1.67
170	80	1.87
192	100	1.92
255 (100%)	2667	1.96

Thus, we are motivated to relax, but not eliminate, the page rate heuristic for code inspections. This will make it easier for inspection planners to stay within the recommended range for page rate, but still obtain a high number of defects per hour. As a candidate threshold for the new page rate heuristic, we choose the 75th percentile of the page rate range for hyper DpH inspections (see Figure 4), which is 95.

Finding 5: The efficiency of a code inspection (based on defects per hour) can be preserved even when the page rate is increased, but the page rate should not be set arbitrarily high. It is still not recommended to exceed 95 pages/hour.

For design inspections, there is a significant correlation between page rate and defects per hour, where defects per hour also increases with page rate, up to about 80 pages/hour, after which it decreases (see Table 7). Thus we conclude in Finding 6 that, for design inspections, the heuristic for page rate could be relaxed to 80 pages/hour. Note that there is a tradeoff between leveraging Findings 5 or 6 with Finding 2, where relaxing the page rate could lead to missing more defects.

Table 7. Variations in numbers of defects found per hour in design inspections as page rate increases.

DESIGN INSPECTIONS		
# of inspections	Page rate less than	Average # of defects/hour found
63	20	0.8
108	40	0.84
129	60	0.94
144	80	1.01
155 (100%)	230	0.99

Finding 6: For design inspections, efficiency can be improved by relaxing the page rate heuristic to 80 pages/hour.

5 Discussion

Below, we reiterate the specific research questions that guided our analysis and discuss the major related findings.

Q1: What effect do inspection parameters have on other inspection outcomes, besides total number of defects found?

As shown in Table 4, compliance with the inspection heuristics brings inconsistent results. The team size and page rate heuristics generally help in improving the defect detection effectiveness of the inspection, but decrease efficiency in terms of effort expended and defects per hour. Moreover, the effectiveness benefits of the number of participants heuristic evaporates when we examine the number of defects detected per page. The meeting length heuristic has the opposite effect, i.e. being in compliance results in more efficient but less effective inspections. However, the effect of the meeting length heuristic is a bit difficult to assess because so few of the inspections in the dataset were out of compliance. Consequently, we can conclude that the original inspection heuristics had room for improvement.

Findings 3-6 summarize our findings about the effect of the inspection parameters on defects found per hour, which is an outcome variable that characterizes the efficiency of an inspection. Finding 3 show that the inspection parameters have very little effect on the efficiency of most inspections. However, Finding 4 implies that, for code inspections on small projects, certain values of inspection team size and meeting length can maximize defects found per hour. Also, higher values for page rate appear to produce equal to higher defects found per hour for design and code inspections (Findings 5 and 6). However, according to Finding 2, this increased efficiency comes at a price in terms of much lower numbers of defects found per page. This implies that page rate greatly affects the tradeoff between inspection efficiency and effectiveness, and so the choice of a value for page rate must be made carefully with regard to the goals of any particular inspection.

Q2: Are there variations in the heuristics that are appropriate for different context factors?

In addressing Q2, we searched for evidence of significant differences in inspection behavior in all the subsets we could form based on context factors (i.e. the intervening variables in Table 2). The most interesting variation we found, encapsulated in Finding 4, had to do with code inspections on small projects. For this set of inspections, much lower thresholds for meeting length and number of participants appeared to result in high numbers of defects per hour.

Q3: Are there variations in the heuristics that would strengthen the relationship between compliance and effectiveness or efficiency?

Our findings shed light on a complicated story about how inspection outcomes are related to their inputs, the subject of Q3. Part of our original intent was to update the inspection planning heuristics concerning appropriate values for the number of inspection participants, the length of the meeting, and the inspection page rate. In some cases, we were able to characterize the benefits and risks of relaxing the current heuristics, e.g. for page rate (Findings 2, 5 and 6). We found it difficult, however, to formulate hard thresholds for the heuristics based on our dataset. The original heuristics were aimed at maximizing the number of defects found, and at least in part they still achieve that goal (see Table 3). We also found that more resources brought to bear in an inspection will result in more defects found (Finding 1). What we did not find, however, was any upper limit to the benefit of more resources. That is, there appears to be no point of diminishing returns, at least within the ranges represented by our dataset. More inspectors spending more time will always, it appears, result in more defects being found. This makes it difficult to formulate any heuristic in the format currently being used (i.e. giving recommended ranges for the inspection parameters).

However, defect detection is not the entire story, so we looked at other inspection outcomes that take into account inspection costs, focusing on the number of defects found per person-hour of effort. Finding a set of heuristics that maximized defects per hour would provide a balanced tradeoff between inspection effectiveness and efficiency. However, such heuristics proved to be elusive. We found that, except for a small subset of inspections, the defects per hour remained in a fairly small range and appeared to be unaffected by the inspection parameters or other intervening variables (Finding 3). For one fairly narrow subset of inspections, however, we found that a very different set of heuristics would help us achieve significantly higher numbers of defects per hour (Finding 4).

There are, clearly, limitations to our analysis and to the dataset upon which it is based. One major limitation is the lack of field defect data, which means that our measures of inspection effectiveness are limited to the raw numbers of defects found. The true measure of inspection effectiveness would be the percentage of total defects in the product that are found by the inspection. We did not have the ability to take that into consideration. Our dataset, although large by the standards of most empirical studies in software engineering, still would have been more useful had it been larger and more diverse. There were many combinations of variables that could not be tested because of a lack of data.

6 Related Work

There exist a number of articles in the literature recommending values of planning parameters for software technical reviews and inspections. We categorize these into (1) work that provides recommendations derived from personal or community insights; (2) work that reports the values used in industrial inspection activities; and (3) work that makes recommendations formed from results of controlled experiments.

Table 8 summarizes recommendations in the literature falling into the first category. The second category includes work by Blakely and Boles [4] that reported the use of a page rate of 200 LoC/hr for code inspection activities at Hewlett-Packard. Doolan [5] reported on requirements inspections at the Seismic Software Support Group (SSSG) at Shell Research where inspection teams usually consisted of 5-7 persons, inspecting 7-8 pages per hour, with meeting length of no more than 2 hours. Eickt et al [6] provided data from 13 reviews at AT&T. The review meetings were scheduled in 2-hour time slots with 6-10 reviewers for requirements reviews and 5-8 reviewers for design reviews.

We can observe little agreement as to the recommended team size, although there is a tendency to keep it relatively small, especially for code inspections. In industry, inspection team size tends to be larger than the recommended team size in the literature from Table 8. A meeting length of 2 hours seems to be the accepted norm. Comparing the suggested page rates introduces an additional challenge due to the different units (e.g. LOC vs. pages) used to express the measure for different artifacts. We can still observe several commonalities. For example, the page rates for code inspection tend to be within 150-200 LOC per hour. On the other hand, the page rates for text-based documents range widely from 3-8 pages per hour.

Table 8 Related work suggesting values for inspection planning parameters based on personal or collective experience, ordered chronologically

Authors	Recommended		
	Team Size	Meeting Length	Page Rate
IEEE Standard for Software Review [7]	3-6		
Fagan [8]	4 (increase team size only if inspected code belongs in a number of interfaces)	2 hours (no more than 2 sessions in a day)	
Ackerman et al [9]	3 (including 1 moderator)		
Gilb and Graham [10]	2-3 for efficiency or 4-5 for effectiveness		0.5-1.5 pages (1 page ~ 600 non-commentary words) and decrease the rate for high risk documents
Barnard and Price [11]			100-150 LOC (with maximum of 500 LOC)
Owens [12]	5-6 for requirement or 1-2 for source code		
Johnson [13]	6-9	Less frequent of meetings	
Laitenburger and DeBaud [14]	3-4 (including 1 author and 1 moderator)		
Wiegers [15]		2 hours	3-4 pages for requirement, design, project plans, and process description; and 150-200 LOC for source

Table 9 summarizes related work exploring the relationships between various planning parameters and the effectiveness and efficiency of inspections through controlled studies. It is difficult to make comparisons among these studies mainly due to the different way each one measures effectiveness and efficiency, in addition to what planning parameters they investigated. Note that some of the studies explored planning parameters that were not addressed by the NASA heuristics; however they serve to point out the diverse factors that may impact an inspection's performance.

Table 9 Related work based on controlled experiments, ordered alphabetically

Authors	Outcome Measures	Planning Parameters - Findings
Boodoo et al [16]	Efficiency – cost savings from doing inspection (and finding defects early)	For design inspections: Team size: Optimal team size increases when cost of post-design detection and fixing activities are higher and meeting duration is lower.
Buck (extracted from [17])	Effectiveness – number of defects	Inspection rate: 90-125 non-commentary statements per hour for code inspections, but more variable for requirements, test plans, and user documentation
Halling and Biffl [18]	Effectiveness – defects lost and gained from meeting	Team size: No significant difference between teams of size 5 and of size 6.
Laitenberger et al [19]	Effectiveness – number of defects found	Preparation rate: Preparation rate significantly affects effectiveness. Material size: Material size has only limited impact on effectiveness.
Porter et al [20]	Effectiveness – defects found/KNCSL	For code inspections: Team size: No significant difference between teams of size 2 and of size 4, but both were better than teams of 1.
Raz and Yaung [21]	Effectiveness – probability of defect escaped the inspections	For design inspections Effort: Effectiveness is reduced with the more effort spent. Material size: It is also reduced when inspected material size is larger. On average, material of size less than 600 KLoC are less than 50% likely to contain escaped defects.
Weller [22]	Effectiveness – defects found/KLoC Efficiency - effort	Team size: 4-person teams were twice as effective and more than twice as efficient as 3-person teams. Preparation rate: Teams with lower preparation rates (<200 lines per hour) had higher effectiveness than teams with higher preparation rates. This increase of effectiveness can offset the loss of effectiveness from using teams with smaller size.

Agreement is mixed between our findings and those in the literature summarized in this section. For example, we found that increasing effort through increasing team size can increase defect detection. Further, we did not find one team size in terms of maximizing effectiveness. On the other hand, we agree with several authors who recommend smaller team sizes for code inspections, although we could only make such a recommendation in the context of small projects and when efficiency is the primary goal. Similarly, we found that increasing inspection effort through increasing the meeting length also provides an increase in inspection effectiveness. This finding seems to conflict with the strict 2-hour meeting length established by the community, although we only explored this relationship using meeting length values up to 4 hours. Our finding, which states that by increasing inspection effort we can also increase inspection effectiveness, also disagrees with Raz and Yaung's finding [21]. Finally, our findings with respect to page rate are in line with much of the literature, that repeatedly finds that defects are missed in inspections when the page rate is too high [19, 21, 22].

7 Implications and Conclusions

In this paper, we have presented results from an exploratory analysis of data from a diverse set of software inspections at NASA. Although the data had some limitations, they provided an unusually rich view of the variety in inspection parameters, context factors, and outcomes. The major findings of this analysis include:

Finding 1: Inspections in which large numbers of defects were reported involved higher levels of effort and more participants.

Finding 2: Inspections of all types (except possibly requirements) would be significantly more effective if they adhered to the original NASA page rate heuristics.

Finding 3: With a few exceptions (treated in Finding 4), the defects per hour for most inspections remains within a narrow range of 1-2 defects per hour of total effort.

Finding 4: For code inspections on small projects, defects per hour is maximized when the number of participants is between 1 and 3, and the meeting length is between .25 and .5 hours.

Finding 5: The efficiency of a code inspection (based

on defects per hour) can be preserved even when the page rate is increased, but the page rate should not be set arbitrarily high. It is still not recommended to exceed 95 pages/hour.

Finding 6: For design inspections, efficiency can be improved by relaxing the page rate heuristic to 80 pages/hour.

While these findings have very specific implications for inspection planning heuristics at NASA, we do not consider the thresholds indicated to be generalizable to any other organization. However, there are more general implications of these findings, and the analysis that led to them, that are relevant for the rest of the software community.

First, it is clear from our analysis that it is very difficult to find an optimal configuration of parameters (i.e. a "sweet spot") that represents a good balance between inspection effectiveness and efficiency. Adding more resources to an inspection will always result in more defects found (Finding 1), and the defects per hour (number of defects found per person-hour of total inspection effort) will always remain within a fairly narrow range, with few exceptions (Finding 3). Intuitively, one must imagine that there is, in fact, an upper bound to the amount of resources (people and time) that can be effectively brought to bear on an inspection, but such a bound does not seem to be evident in our dataset.

Second, it appears that any inspection heuristics need to be revisited from time to time, both from the point of view of effectiveness and compliance. In our dataset, it was clear that NASA teams were increasingly out of compliance with the page rate heuristic, i.e. they were inspecting much more material per hour of meeting time than the heuristics recommended. Findings 5 and 6 show, moreover, that efficiency is equal or enhanced when the page rate is allowed to increase. However, Finding 2 shows that a significant cost, in terms of inspection effectiveness (in particular defects found per page), is paid for this increase in efficiency and, in fact, a very large number of inspections in the dataset would most likely have been more effective if they had conformed to the original page rate heuristic.

Finally, the analysis we were able to do, and the insights gained, point to the value of collecting and analyzing detailed inspection data. NASA has been a pioneer and a leader in the practice of software inspections [23] for several decades, and so is in the unusual (but hopefully not unique) position of having a rich historical record from which to learn about and improve upon inspection practice. Our analysis shows that having such a rich experience base can yield valuable feedback that is directly actionable for any organization.

ACKNOWLEDGMENTS

This work was sponsored by NASA grant NNG05GE77G, "Full-Lifecycle Defect Management Assessment." The authors wish to thank our collaborators at NASA who made this work possible by helping us compile this dataset from across many projects and Centers.

REFERENCES

1. F. Shull, et al., "What We Have Learned About Fighting Defects," *Book What We Have Learned About Fighting Defects*, Series What We Have Learned About Fighting Defects, ed., Editor ed.^eds., 2002, pp. 249-258.
2. J.C. Kelly, et al., "An Analysis of Defect Densities Found During Software Inspections," *Journal of Systems Software*, vol. 17, no. 2, 1992, pp. 7.
3. NASA, "NASA Procedural Requirements 7150.2 Subject: NASA Software Engineering Requirements," *Book NASA Procedural Requirements 7150.2 Subject: NASA Software Engineering Requirements*, Series NASA Procedural Requirements 7150.2 Subject: NASA Software Engineering Requirements, ed., Editor ed.^eds., 2009, pp.
4. F.W. Blakely and M.E. Boles, " A Case Study of Code Inspections," *Hewlett-Packard Journal*, vol. 42, no. 4, 1991, pp. 6.
5. E.P. Doolan, " Experience with Fagan's Inspection Method," *Software – Practice and Experience*, vol. 22, no. 2, 1992, pp. 10.
6. S.G. Eickt, et al., "Estimating Software Fault Content Before Coding," *Proc. Software Engineering, 1992. International Conference on*, 1992, pp. 59-65.
7. , "IEEE standard for software reviews," *IEEE Std 1028-1997*, 1998.
8. M.E. Fagan, "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems Journal*, vol. 15, no. 3, 1976, pp. 30.
9. A.F. Ackerman, et al., "Software inspections: an effective verification process," *Software, IEEE*, vol. 6, no. 3, 1989, pp. 31-36.
10. T. Gilb and D. Graham, *Software Inspection*, Addison-Wesley Longman Publishing Co., 1993.
11. J. Barnard and A. Price, "Managing code inspection information," *Software, IEEE*, vol. 11, no. 2, 1994, pp. 59-69.
12. K. Owens, "Software Detailed Technical Reviews: Finding and Using Defects," *Book Software Detailed Technical Reviews: Finding and Using Defects*, Series Software Detailed Technical Reviews: Finding and Using Defects, ed., Editor ed.^eds., 1997, pp. 128-133.
13. P.M. Johnson, "Reengineering Inspection," *Communications of the ACM*, vol. 41, no. 2, 1998, pp. 4.
14. L. O. and J.M. DeBaud, "An Encompassing Life-cycle Survey of Software Inspection," *Journal of Systems and Software*, vol. 50, no. 1, 2000, pp. 27.

15. K.E. Wiegers, *Peer Reviews in Software*, Addison-Wesley, 2002.
16. B. S., et al., "An Empirical Evaluation of the optimal team size for UML design inspection," *Book An Empirical Evaluation of the optimal team size for UML design inspection*, Series An Empirical Evaluation of the optimal team size for UML design inspection, ed., Editor ed.^eds., 2000, pp.
17. M.E. Fagan, "Advances in Software Inspections," *IEEE Transactions on Software Engineering*, vol. 12, no. 7, 1986, pp. 8.
18. M. Halling and S. Biffl, "Investigating the influence of software inspection process parameters on inspection meeting performance," *Software, IEE Proceedings -*, vol. 149, no. 5, 2002, pp. 115-121.
19. O. Laitenberger, et al., "Quantitative modeling of software reviews in an industrial setting," *Proc. Software Metrics Symposium, 1999. Proceedings. Sixth International*, 1999, pp. 312-322.
20. A.A. Porter, et al., "An experiment to assess the cost-benefits of code inspections in large scale software development," *Software Engineering, IEEE Transactions on*, vol. 23, no. 6, 1997, pp. 329-346.
21. R. T. and A.T. Yaung, "Factors Affecting Design Inspection Effectiveness in Software Development," *Information and Software Technology Journal* vol. 39, no. 4, 1997, pp. 9.
22. E.F. Weller, "Lessons from three years of inspection data [software development]," *Software, IEEE*, vol. 10, no. 5, 1993, pp. 38-45.
23. F. Shull and C. Seaman, "Inspecting the History of Inspections: An Example of Evidence-Based Technology Diffusion," *Software, IEEE*, vol. 25, no. 1, 2008, pp. 88-90.