Design and Code Inspection Metrics

Alison A. Gately
Software Engineering Laboratory
Defense Systems Segment
Raytheon Systems Company
50 Apple Hill Drive - T3SV14
Tewksbury, MA 01876-0901
Telephone: (978) 858-4192
Email: Alison_A_Gately@res.raytheon.com

Abstract

The considerable amount of time and cost saved by detecting errors early in the software development cycle demonstrate the need to improve design and code inspection efficiency (Humphrey, 1994; Knight & Meyers, 1993). Some past research suggested the advantages of smaller review groups (Porter, Siy, Toman, & Votta, 1997; Blackburn, Scudder, & Van Wassenhove, 1995; Carmel, 1995; Yourdon, 1997), while sporadic studies on inspection rates found an ideal reviewing pace (Russell, 1991; Mackertich, 1995). In this study, historical inspection data from a large real-time embedded system were analyzed with the intention of improving the current review process. This post hoc experiment investigated the errors found per thousand source lines of code (Error/KSLOC) in terms of three independent variables: review rate, preparation rate, and number of reviewers. Until now, we were not aware of any statistical investigation that examined ideal preparation rates for inspections.

Prior to analysis, we hypothesized that all independent variables influenced the Error/KSLOC during reviews. Analysis involved both the non-experimental approach of correlations and the true experimental approach of analysis of variance (ANOVA). The investigation revealed that in design inspections, preparation rate and review rate were significant factors; in code inspections, only review rate was significant; all other variables were insignificant. The outcome of the experiment was discussed in terms of implementing statistical findings, comparing results to previous studies, and applying these concepts to future work.

1. Introduction

Previous industry investigations offer that even the most experienced software engineer injects errors. Past studies show the defect density of software ranges from 49.5 to 94.6 per thousand lines of code (Jones, 1996), while other studies reveal even higher density estimates (Humphrey, 1994). Extensive research offers that detecting errors early in the software development cycle minimized both time and cost. For example, another study by Watts Humphrey found an average error correction time ratio of 1 requirement, 3 to 6 design, 10 coding, 15-40 development test, 30-70 acceptance test, and 40 to 1000 during operation. Furthermore, research showed an average cost per defect ranged from $90-$120 in inspections to $10,000 in test. Companies that used an early inspection system reported a tenfold reduction in the number of defects found in test and up to an 80 percent reduction in test costs, including the cost of the inspections (Humphrey, 1994). Consequently, errors detected at earlier life cycle phases drastically reduce time and cost.

Even though uncovering errors early in the development phase clearly saves time and cost, limited research has been performed to improve the inspection process. The finite amount of research is surprising because inspections are considered a critical and common method of early phase error detection. In the past, most inspection studies were restricted to the number of reviewers at an inspection. A small group size was found to be beneficial, even though the recommended amount of reviewers vary between two (Porter et al., 1997)

and approximately six reviewers (Blackburn et al., 1995; Carmel, 1995; Yourdon, 1997). Limited studies exist on the preparation and inspection rates for meetings. We found a single examination which offered statistical data on rate: Bell-Northern Research uncovered an ideal code inspection pace of 150 lines per hour for their large-scale development project (Russell, 1996). The only other available study was at Raytheon, when empirical research implied an ideal review rate 100-250 SLOC/Hour for aggregate design and code data (Mackertich, 1995).

Wanting to enhance our process at Raytheon Systems Company, we decided to thoroughly examine design and code inspection records that were stored in unanalyzed databases. Our historical data were generated during the development of a large real-time embedded system. We conjectured the data would provide insight into improving the inspection process. Our goal was to determine common inspection variables that could be manipulated to increase the amount of errors detected at design and code reviews.

This study examined three independent variables with respect to the errors found in the peer review process. In order to assure the results could be applied to future work, the variables were normalized. The independent variables were:

- Review Rate (SLOC/Hour) = Reviewed SLOC/Review Time
- Preparation Rate per Reviewer (SLOC/Hour per Reviewer) = Reviewed SLOC/(Total Preparation Time/Number of Reviewers)
- Number of Reviewers (Reviewers) = All Participants at the Review, excluding the Presenter.

The definition of preparation rate was an average speed for the reviewers based on the review team's preparation time. The dependent variable was:

- Error/KSLOC = Combined Logical Errors/(Reviewed SLOC/1000).

Therefore, if the number of errors detected increased at different levels of the independent variables, then these factors could be manipulated to increase review efficiency. Cost, a factor in analysis of error detection, was not a restricting issue, because detecting an error at reasonable rate during an inspection was less costly than having the error escape and detecting it at a later phase, refer to Humphrey's ratio data above (1994). This was the first known study on preparation rate for code reviews or on preparation and review rate for design reviews.

The historical data were 13,655 and 17,286 developed and inspected SLOC for design and code, respectively. After excluding the outliers, there were 43 design and 157 code reviews.

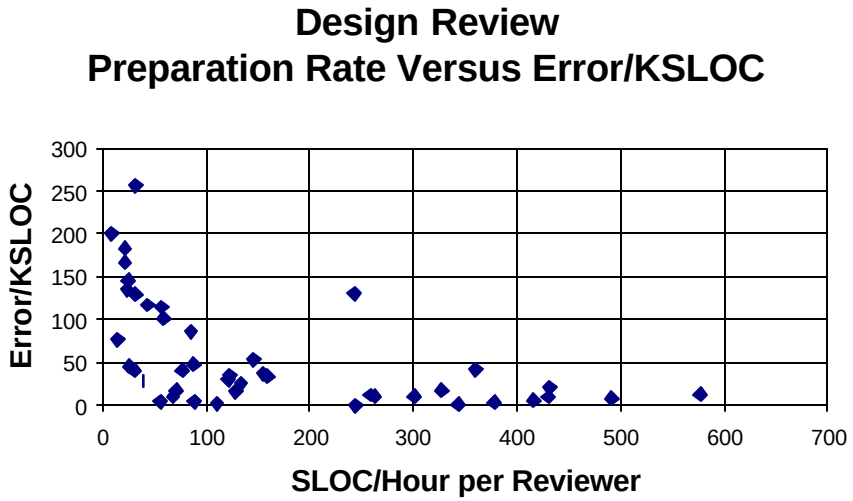
2. Data Observations

The table offers mean and standard deviations for the independent and dependent variables.

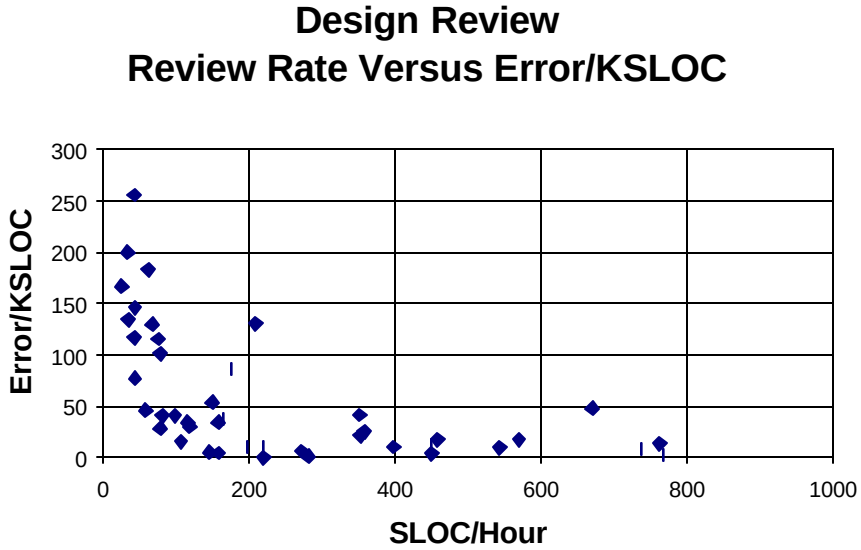
	DESIGN		CODE	
Variables	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
Preparation Rate	165.2	153.7	113.9	79.3
Review Rate	242.2	217.3	203.3	140.9
Number of Reviewers	6.5	2.2	5.5	1.8
Error/KSLOC	57.4	63.5	31.0	31.1

The large standard deviations for review and preparation rates were indicative of skewed data. For instance, some inspection rates were over 600 SLOC/Hour.

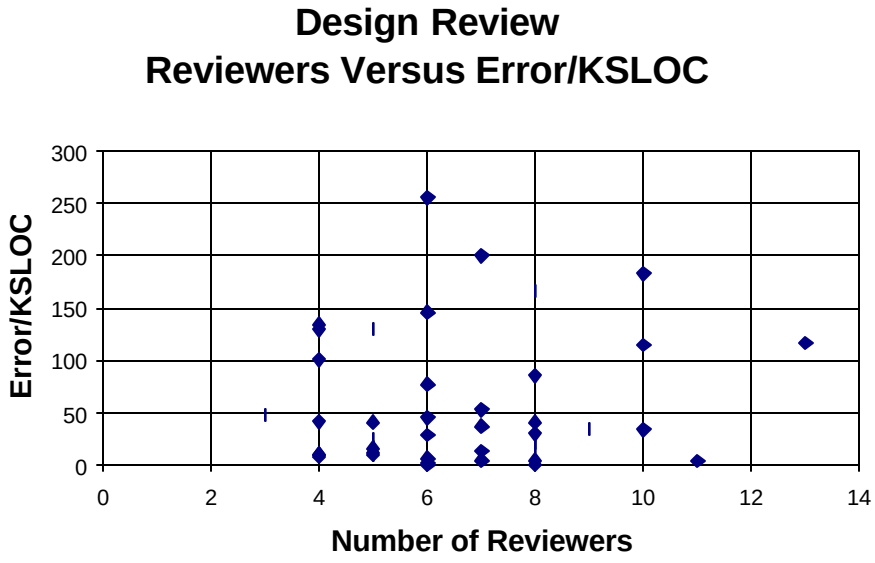
Next, design and code inspections Error/KSLOC were graphed with respect to each independent variable in XY-charts. The charts and discussion follow. These visuals may be beneficial when the formal analyses are discussed.



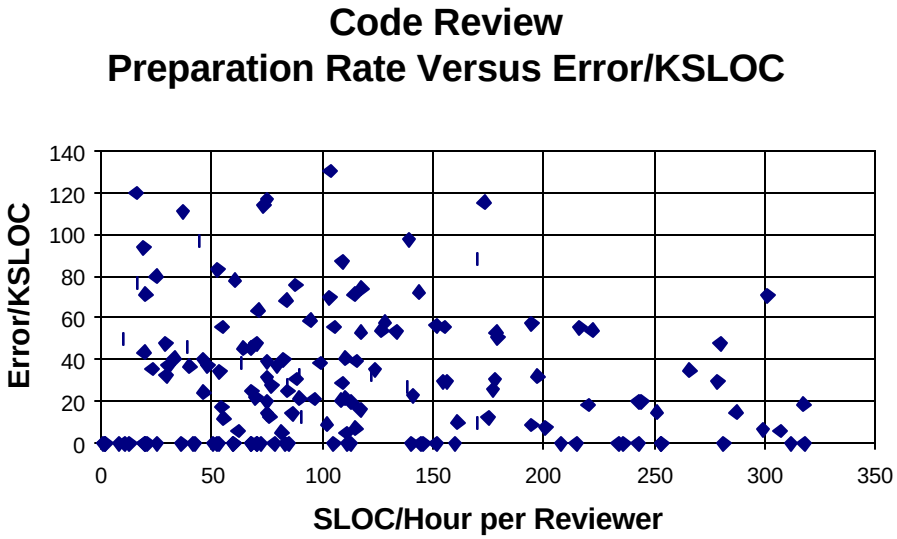
In the example above, we see more Error/KSLOC detected at slower preparation rates. The right skewed data plot reveals that a preparation rate over 250 SLOC/Hour per Reviewer never detected more than 50 Error/KSLOC. This chart empirically offers the faster a review is prepared, the less errors detected; hence, preparing slowly may be advantageous.



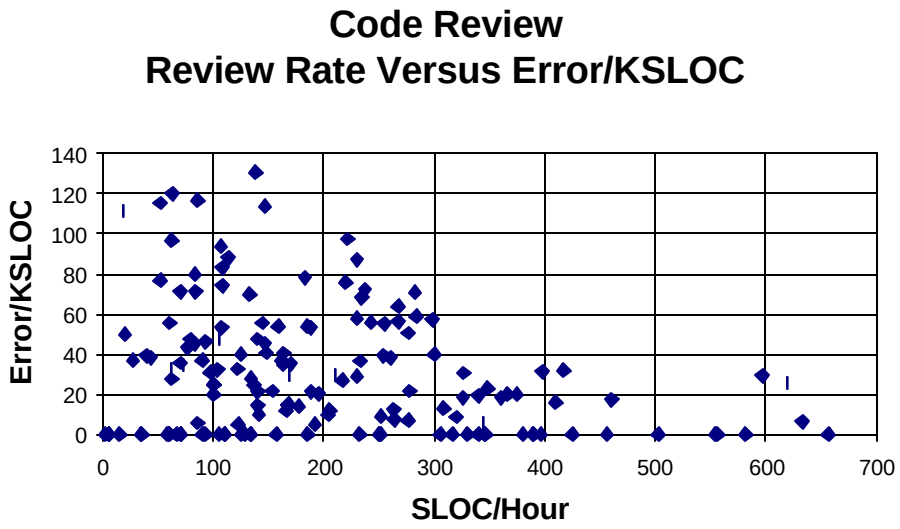
The XY-chart reveals a pattern similar to the first chart; faster inspection rates reveal fewer errors. We see a decline in Error/KSLOC at about 200 SLOC/Hour per Reviewer. This speculatively suggests review speed effects the amount of errors found.



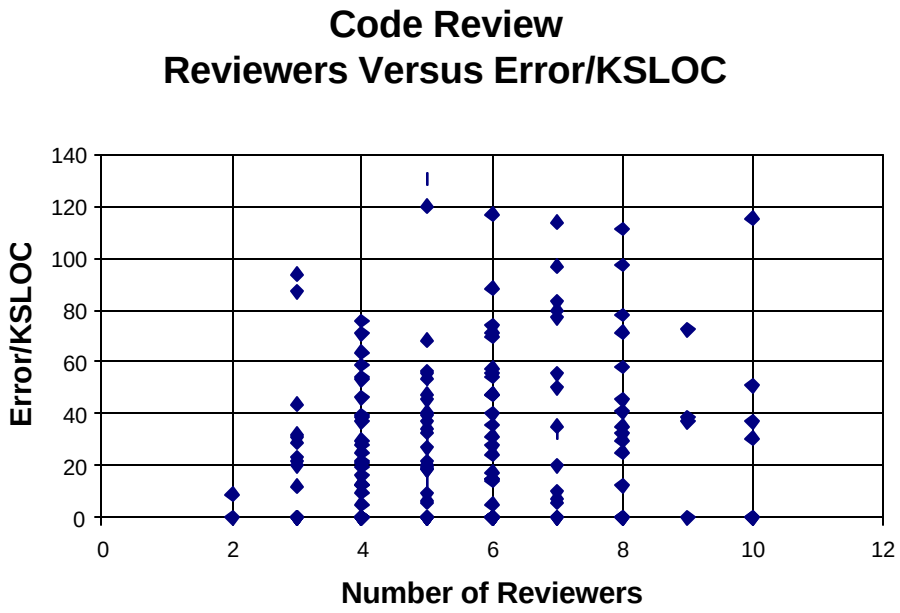
The number of reviewers plot is inconclusive. Between 4 and 13 reviewers reveal some inspections over 100 Error/KSLOC. Also inspections with the same number of reviewers fail to offer a consistent message; for example, 6 reviewers discover between 0 and 256 Error/KSLOC. We must wait for more formal analysis to uncover possible patterns.



Even though the code chart is not as strongly skewed as the preparation rate design plot, we see that higher Error/KSLOC values were associated with the slower preparation rates. Therefore, we suspect some influence of preparation rate on Error/KSLOC for code reviews.



In this code inspection chart, we see a distinct curve with less Error/KSLOC discoveries after 300 SLOC/Hour. The plot suggests that a faster reviewing pace may lead to missed error detection.



Once again, the number of reviewer failed to divulge a pattern. Greater Error/KSLOC values range both among the number of reviewers and within the number of reviewers categories.

3. Analysis and Results

Now that we speculated about Error/SLOC with respect to our three variables, we employ rigorous analysis to obtain conclusive results. The data were examined first as a correlational study and then as a true experiment. By applying experimental techniques, we were able to examine the independent variables at different levels. The descriptions of the two analyses are below.

3.1. Non-Experimental Approach

In the non-experimental approach, correlations are used. Correlation analysis determines a linear relationship between variables, i.e. as the independent variable changes, the amount of defects change. For the design data, the correlation between Error/KSLOC versus review rate and Error/KSLOC versus preparation rate were statistically significant, $r(41) = |-0.54|$ and $|-0.53|$, $p<0.01$, respectively. For the code data, the correlation between Error/KSLOC and review rate was statistically significant, $r(41) = |-0.28|$, $p<0.01$. The interpretations of these results follow:

- As review rate decreased, the number of errors found tended to increase in both design and code reviews.
- As preparation rate decreased, the number of errors found tended to increase in design reviews. Preparation rate showed a non-significant negative correlation in code reviews.
- As preparation time increased, the time spent at the review tended to increase in both design and code reviews.

The last consistently strong result was not examined in the study, but it was interesting nonetheless. It seemed to imply that if a reviewer did not spend time preparing then s/he had nothing to discuss at the review.

Two non-significant correlations were noteworthy. First, there were no correlations between number of reviewers and other data. Compatible with past results, the number of reviewers and other variables were not linearly related (Blackburn et al., 1995; Carmel, 1995; Yourdon, 1997). Second, the correlation for Error/KSLOC and preparation rate in code inspection offered empirical trends, as we saw in the XY-chart. Additionally, when we analyzed the data at less rigorous levels, significance was seen. The lack of significant negative correlation for Error/KSLOC and preparation rate in code reviews at our strict level, may be explained by experience; less experienced reviewers often tackle code review preparation as an educational assignment. See Discussion section.

3.2. True Experiment Approach

Once the correlations were found, the data were analyzed from a true experiment approach using analysis of variance (ANOVA). ANOVAs enable us to implement the Scheffé test, which offers actual ideal rates and number of reviewers. This beneficial technique allows us to extend reviewers descriptive guidelines. Numerous multilevel one-factor between-subject design ANOVAs were performed on the three independent variables with respect to Error/KSLOC at the $\alpha = 0.05$ level. If the Scheffé test revealed some levels of the variables to be statistically equivalent, the mean with fastest preparation rate, fastest review rate, or least number of reviewers was recommended for efficiency purposes. See the tables below for results.

DESIGN INSPECTIONS	ANOVA Results	Scheffé Test Results
Preparation Rate	$F(10,32) = 3.62$, $MS_e=2481.6$, $p<0.05$	Less than 250 SLOC/Hour per reviewer
Review Rate	$F(13,29) = 4.28$, $MS_e=1998.2$, $p<0.05$	Less than 250 SLOC/Hour
Number of Reviewers	Failed to reveal a significant difference	N/A

CODE INSPECTIONS	ANOVA Results	Scheffé Test Results
Preparation Rate	$F(6,150) = 1.35$, $MS_e=963.2$, $p>0.05$ (Emp advantage seen, but failed to reveal a significant difference)	N/A
Review Rate	$F(13,143) = 2.08$, $MS_e=895.6$, $p<0.05$	Less than 300 SLOC/Hour
Number of Reviewers	Failed to reveal a significant difference	N/A

Once again, code inspection preparation rate and Error/KSLOC failed to reveal a significant difference at the $\alpha = 0.05$ level. At less rigorous levels, $\alpha = 0.25$, significance was revealed and less than 200 SLOC/Hour

prevailed as the recommended rate. Although discussion of changed analysis level is unusual, as experimenters, we considered it obligatory not to discard the potential importance of preparation rate for code reviews in its first study on account of Raytheon Systems Company educational use of code inspections. See Discussion section.

4. Discussion

By observing our historical data, we found the ideal preparation and review rates for design inspections were less than 250 SLOC/Hour and the ideal review rate for code inspections was less than 300 SLOC/Hour. Implementation involved presenting the findings to management, who in turn discouraged reviewers to deviate from these ideal rates.

Additionally, Raytheon System Company management seeks to uncover a project specific cost metric to offer with future data. This would expand upon the extrapolation of Humphrey's time and cost ratios (1994).

A possible reason for some statistically non-significant findings was the lack of a project experience metric. For example, we incorrectly hypothesized that team size influenced Error/KSLOC, as seen in previous studies. This contradiction may be easily explained by experience. New reviewers are often assigned to code inspections as an educational assignment. Therefore, the number of reviewers increased, but the majority of the contributions were feasibly limited to key team members. In addition, empirical trends in preparation rates for code inspections indicate slower rates were beneficial, but significance at the chosen statistical level was not seen. In this case, a measure of experience in the recording process may have been very revealing, e.g. new reviewers spend a disproportionate amount of time preparing for their new task. In further study, experience could be measured by the summation of the years of reviewer project experience per review team.

In the future, we would like to see replication of these findings in another real-time embedded system. To our knowledge no such comprehensive investigation has been completed. Furthermore, performing a controlled experiment by assigning review team size, preparation and review rates to design and code inspections would allow cause and effect relationships to be confirmed.

References

Blackburn, J., Scudder, G., & Van Wassenhove, L. (1995). Improving Speed and Productivity of Software Development: A Survey of European Software Developers, Fontainebleau, France: INSEAD Working Papers.

Campo, M. J. (1996-1998). [Interviews with Michael Campo, manager of Air Defense Battle Management Department]. Tewksbury, MA: Raytheon Systems Company.

Carmel, E. (1995). Cycle Time in Packaged Software Firms. Journal of Product Innovation Management, 12(2), 110-123.

Humphrey, W. (1994, December) A Personal Commitment to Software Quality. Web sight: *http://www.sei.cmu.edu*, Pittsburgh, PA: The Software Engineering Institute.

Jones, T. C. (1986). Programming Productivity. New York, NY: McGraw-Hill, 172.

Kiess, H. O., & Bloomquist D. W. (1985). Psychological Research Methods. Newton, MA: Allyn and Bacon, Inc.

Knight, J. C. & Meyers, E.A. (1993, November). An Improved Inspection Technique. Communications of the ACM. 36(11), 51-61.

Mackertich, N. (1995, October). Applying SPC to the Software Development Process. Symposium conducted at the Forty-Ninth Annual Northeast Quality Control Conference, Falmouth, MA.

Porter, A. A., Siy, H. P., Toman, C. A., & Votta, L. G. (1997). An Experiment to Assess the Cost-Benefits of Code Inspection In Large Scale Software Development. IEEE Transactions of Software Engineering, 23(6), 329-346.

Russell, G. W. (1991, January). Experience with Inspection in Ultralarge-Scale Developments, IEEE Software, 25-31.

Yourdon, E. (1997, October). Metrics for Death-March Projects. Symposium conducted at Eighth International Conference On Applications of Software Measurement, Atlanta, GA.

Alison A. Gately

Alison A. Gately is a member of Raytheon Systems Company Software Engineering Laboratory in the Defense Systems Segment. Her primary focus is innovating and maintaining metric processes for both the former Raytheon Electronic Systems and the current Raytheon System Company. Ms. Gately's accomplishments include analysis of design/code inspection efficiency, defect density and defect containment data along with continuous process improvement. Her most recent work involves controlling SLOC growth and forming initiatives consistent with a six-sigma software approach. Currently she is leading the creation of the of the nationwide RSC Web-based Software Metric Database Training Tool.

In addition to Ms. Gately's metric work, she is active in other facets of engineering. She is a trainer for Raytheon Systems Company and supports recruitment activities. Also, she holds memberships in IEEE and PBK. Alison graduated *magna cum laude* from St. Lawrence University with B.S. degrees in mathematics and psychology.