N92-19435

AN ANALYSIS OF DEFECT DENSITIES FOUND DURING SOFTWARE INSPECTIONS

John C. Kelly, Ph.D, CQA
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91101, USA

Joseph S. Sherif, Ph.D.
California State University, Fullerton
Fullerton, California 92634, USA

Jonathan Hops
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109, USA

ABSTRACT

Software inspection is a technical evaluation process for finding and removing defects in requirements, design, code and tests. Software inspections have been used by a wide variety of organizations for improving software quality and productivity since their original introduction at IBM. The Jet Propulsion Laboratory (JPL), California Institute of Technology, tailored Fagan's original process of software inspections to conform to JPL software development environment in 1987. However, the fundamental rules of the Fagan inspection process were adhered-to.

Detailed data was collected during the first three years of experience at JPL on 203 inspections. Statistics are discussed for this set of inspections. Included, on a per inspection basis, are averages of: staff time expended, pages covered, major defects found, minor defects

found and inspection team size. The inspection team size varied from three to eight participants with the JPL Product Assurance Organization providing most of the moderators.

Analysis of variance (alpha = 0.05) showed a significantly higher density of defects during requirements inspections. It was also observed, that the defect densities found decreased exponentially as the work products approached the coding phase.

Increasing the pace of the inspection meeting decreased the density of defects found. This relationship was observed to hold for both major and minor defect densities, although it was more pronounced for minor defects.

This paper provides guidelines for conducting successful software inspections based upon three years of JPL experience. Readers interested in the practical and research implications of software inspections should find this paper helpful.

INTRODUCTION

This paper describes an analysis of factors influencing the defect density of products undergoing software inspections. Software intensive projects at the Jet Propulsion Laboratory (JPL) require a high level of quality. JPL, a part of the California Institute of Technology, is funded by NASA to conduct its unmanned interplanetary space program. Software inspections were introduced at JPL in 1987 to improve software quality by detecting errors as early in the developmental lifecycle as possible.

Software Inspections are detailed technical reviews performed on intermediate engineering products. They are carried out by a small group of peers from organizations having a vested interest in the work product. The basic process is highly structural and consists of six consecutive steps: planning, overview, preparation, inspection meeting, rework and follow-up. The inspection process is controlled and monitored through metrics and checklists. One of the best fundamental descriptions of this process is Fagan's original article [Fagan, 1976].

JPL tailored Fagan's original process to improve the quality of the following technical products of a software intensive system: Software Requirements, Architectural Design, Detail Design, Source Code, Test Plans, and Test Procedures. For each of these types of products a checklist was tailored for JPL's application domain, standards and software development environment. Supplemental tailoring included the addition of a "third hour" step to Fagan's process. The "third hour" step was first discussed by Gilb [Gilb, 1987]. JPL's "third hour" step includes time for team members to discuss problem solutions and clear-up open issues raised in the inspection meeting. Other tailoring included substantial use of Software Product Assurance personnel as inspection moderators, a JPL specific training program, and new data collection forms.

The analysis of defect densities from inspections was performed for the purpose of 1) ensuring that the conditions of a quality inspection are being met by JPL inspections, 2) verifying previous research findings on inspections and 3) understanding the factors which influence inspection results. It was expected that the results would agree with previous findings on inspections, but due to slight variations in the variables collected, some differences were observed.

Methods

Data was collected on 203 inspections performed on five software intensive projects at JPL. Practically all inspection team members were trained in a one and a half day course on formal inspections [Kelly, 1987]. Software Product Assurance supplied 70% of the moderators. The inspections took place between February 1987 and April 1990. Although the projects used Ada, C and Simula, only 16%

were performed on code. Table 1 shows the types of inspections performed in this study and the sample size for each type.

The data included in this study was recorded on the "Formal Inspection Detailed Report" and "Inspection Summary Report" forms [Appendix 1 and 2]. Each inspection produced a complete set of forms indicated in the process diagram [Figure 1]. This information was placed into a database and monitored. Occasionally, the chief moderator would contact the inspection moderator when reports were abnormal. This was done to provide feedback for inspections which were experiencing difficulties. Eleven inspection reports were rejected for analysis in this sample for the reason that they violated some of the fundamental rules of inspections as shown in [Appendix 3].

Checklists were used to 1) help inspection team members focus on what to look for in a technical work product, and 2) provide categories for classifying defects. A generic checklist was provided in the training materials for each type of inspection: R1, I0, I1, I2, IT1 and IT2. Projects may use the generic checklist or tailor this list to match their own environment and development standards. However, we encouraged projects to maintain the 15 main categories for types of defects shown in the "Formal Inspection Detailed Report". [Appendix 1].

The metrics used to monitor and analyze inspections can be classified into three prime areas: staff time, types of defects and workproduct characteristics. The staff time expended was recorded by total hours during each stage of the inspection process. Part way through our study we began collecting staff time by the role played in the inspection meeting (author, moderator, or inspector). The organizational areas, represented by these participants, were also recorded.

Each defect was classified by severity, checklist category, and "type". The severity of defect was classified either major, minor, or trivial.

Trivial defects in grammar and spelling were noted and corrected, but not included in this data analysis nor on the "Formal Inspection Detailed Report" [Appendix 1].

The "type" of defect (mission, wrong, or extra) was recorded on the forms, but not in the database. This information is not as institutionally critical, however, the authors find it to be a useful guide during the rework stage of the process.

The workproduct characteristics included size (by pages of lines of code), phase and type of product (requirements, test plan, etc.), and project Since inspections were usually introduced relatively early in the developmental lifecycle, when most products were technical documents, the preferred size reporting metric was in pages. A typical page of JPL documentation is single spaced, 38 lines per page in 10 point font size. A page containing a diagram was counted equal to a page of test. The authors felt that number of pages was a more accurate measure of material undergoing inspection than "estimated lines of code" for technical documents, since projects did not have a history of a detailed accounting of the second metrics during the early lifecycle phases. Due to most of the data being reported in pages, different relationships are found than in previous studies. It should be noted that "pages" is more of a producer oriented statistic than it is a product oriented measure. One of the key metrics that was used in this analysis is "density of defects per page". This metric was used to compare inspections of different types and their related factors.

Results

Results showed a higher density of defects in earlier lifecycle products than in later ones. An analysis of variance was performed on data collected from the different types of inspections in the sample (R1, I0, I1, I2, IT1, and IT2). Figure 2 shows the average number of defects found per page for each of the inspection types. The analysis of variance test showed that at Alpha = 0.05, the defect density at the software requirements inspection (R1) is significantly higher than that of source code inspection (I2), and also the defect density at test plan inspection (IT1) is significantly higher than that at test procedures and function inspection (IT2). It was also observed that the defect densities found during inspections decreased exponentially as the development work products approach the coding phase [Figure 3].

The staff hours needed to fix defects were not found to be significantly different across the different phases of the lifecycle [Figure 4]. It should be noted that the defects found and fixed during these inspections originated during the lifecycle phase in which they were detected. Latent defects which were found in high level documents during inspections were recorded as "open issues" and submitted to the change control board. Since the researchers did not know the timely outcome from the control board, these potential defects are not tracked in this study. However, the average cost to fix defects during the inspection process (close to their origin) was 0.5 hours, which is considerably less than the range of 5 to 17 hours to fix defects during formal testing reported by a recent JPL project.

Previously, inspection defect counts were found to decrease as the amount of code to be inspected increased [Buck, 1981]. Figure 5 shows this trend to be sure for the total sample of inspections in this study with respect to defect density per page.

The average inspection team composition and size for this sample are shown in Figure 6 by type of inspection. For development inspection types (R1, I0, I1, and I2) the trend is for larger teams for requirements and high level documents while smaller teams are needed for code. The inspection program at JPL tried to insure that teams were comprised of members from organizations having a vested interest in the work product. The rationale for this was to keep inspections from being biased toward an organization's internal view of the product.

Figure 7 shows the distribution of defects percentage by defect types and defect categories.

CONCLUSIONS

Experience has shown that formal inspection of software is a potent defect detection method; and thus, it enhances overall software quality by reducing rework during testing, as well as maintenance efforts.

The following items highlight the results of JPL experience with formal inspections.

1. A variety of different kinds of defects are found through inspections with Clarity, Logic, Completeness, Consistency, and Functionality being the most prevalent.

2. Increasing the number of pages to be inspected at a single inspection decreases the number of defects found.

3. Significantly more defects are found per page at the earlier phases of the software lifecycle. The highest defect density was observed during Requirements inspections.

4. The cost in staff hours to find and fix defects was consistently low across all types of inspections. On average it took 1.1 hours to find a defect and 0.5 hours to fix and check it (major and minor defects combined).

5. Larger team sizes (6 to 9) for higher level inspections (R1 and I0) are justified by data which showed an increased defect finding capability.

JPL has adopted formal inspections for many of its software intensive projects. The results are very encouraging and show very significant improvements in software quality.

REFERENCES

[1] Buck, F. O., "Indications of Quality Inspections", IBM Technical Report, TR 21-802, September 1981.

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

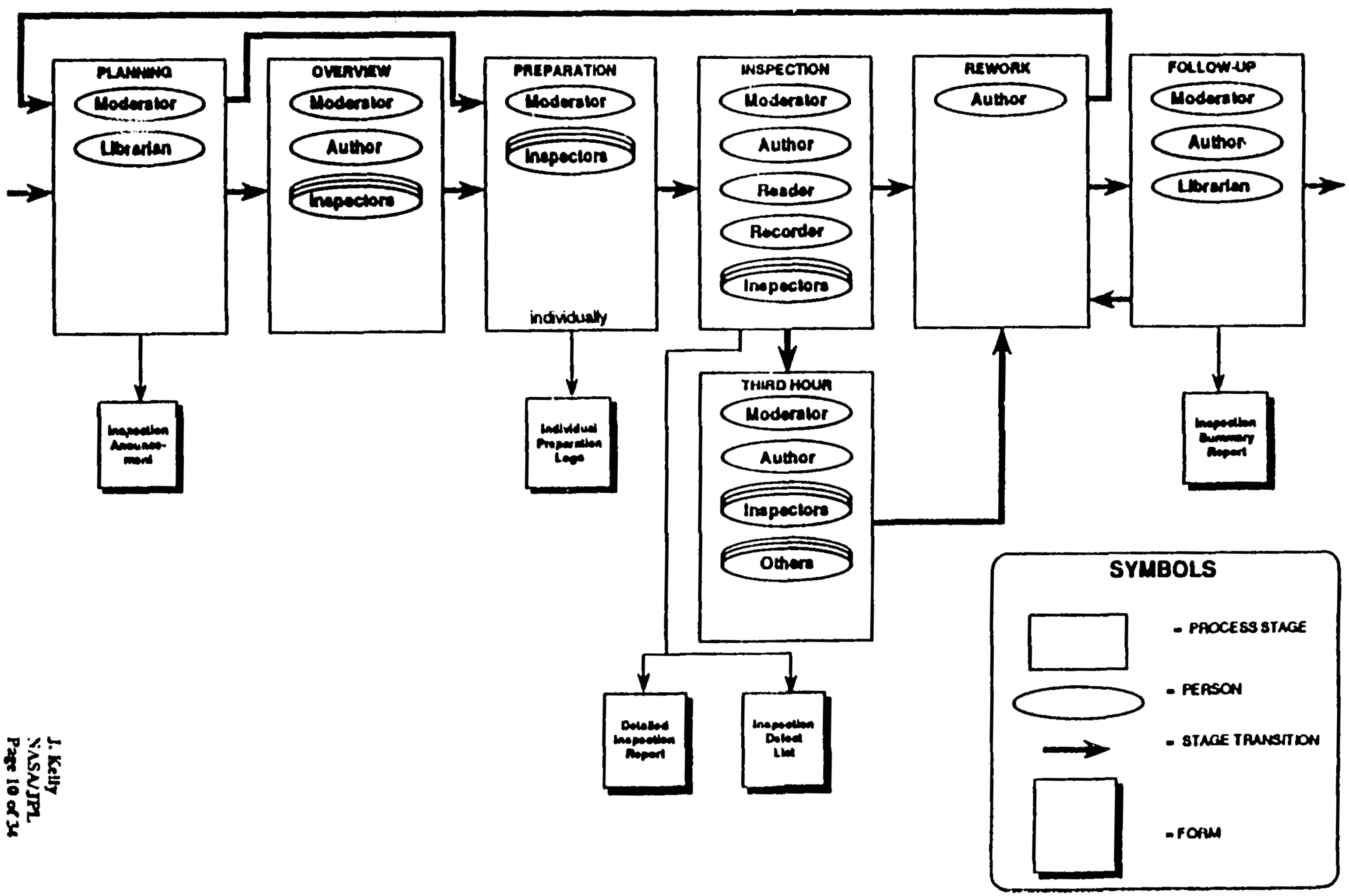
[3] Gilb, T., Principles of Software Engineering Management, Addison Wesley, Reading, MA, 1987.

[4] Kelly, J. C., "Formal Inspections For Software Development", Software Product Assurance Section, JPL, California Institute of Technology, Pasadena, CA, 1987.

Table 1: Types of inspections included in this analysis

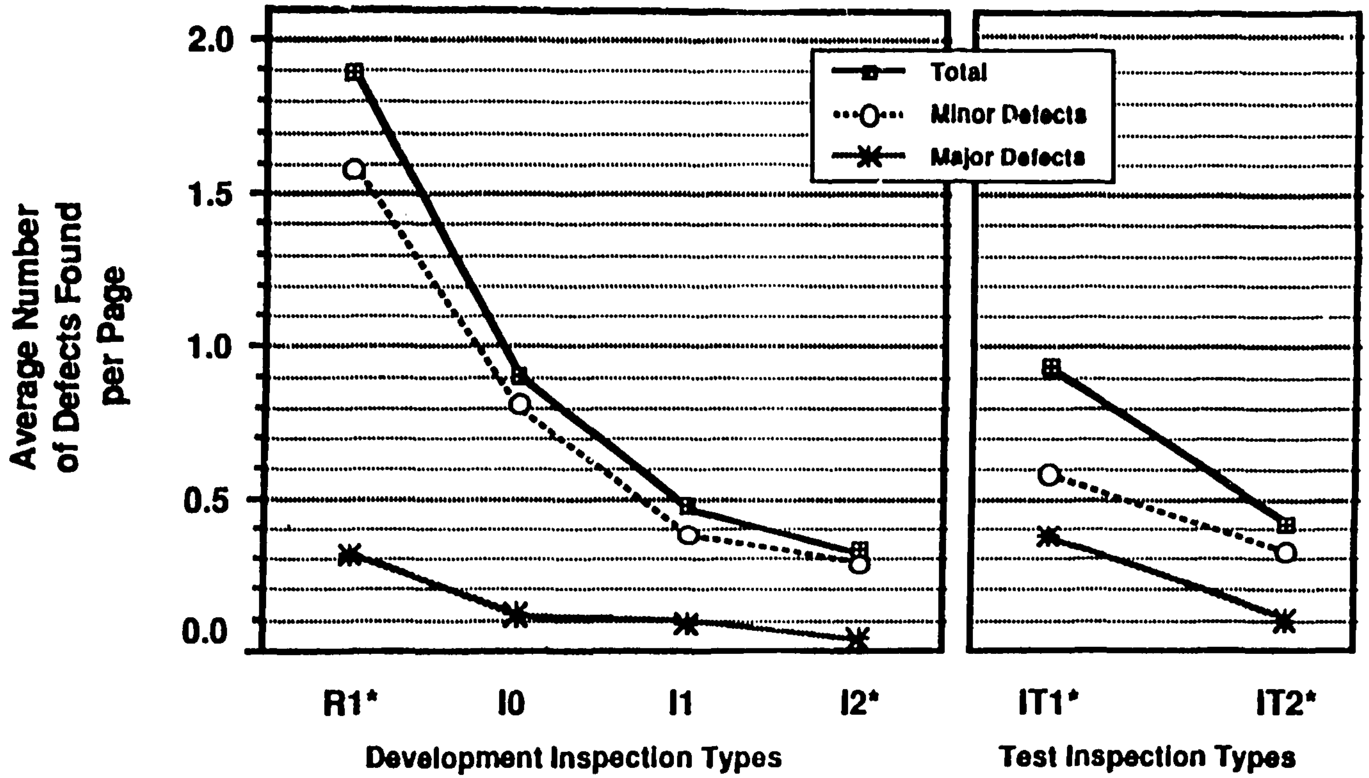
Inspection Abreviation	Inspection Type	Sample Size
R1	Software Requirements Inspection	23
I0	Architectural Design Inspection	15
I1	Detailed Design Inspection	92
I2	Source Code Inspection	34
IT1	Test Plan Inspection	16
IT2	Test Procedures & Functions Inspection	23
	Total:	203

Figure 1: Overview of the Software Inspection Process



J. Kelly
NASA/JPL

Figure 2: Defect Density Versus Inspection Types



* *At the alpha= 0.05 level of significance ANOVA F test showed a significant difference between the defect densities of R1 and I2, and between IT1 and IT2.*

Figure 3. A developed predictive model for defect density as a function of inspection type

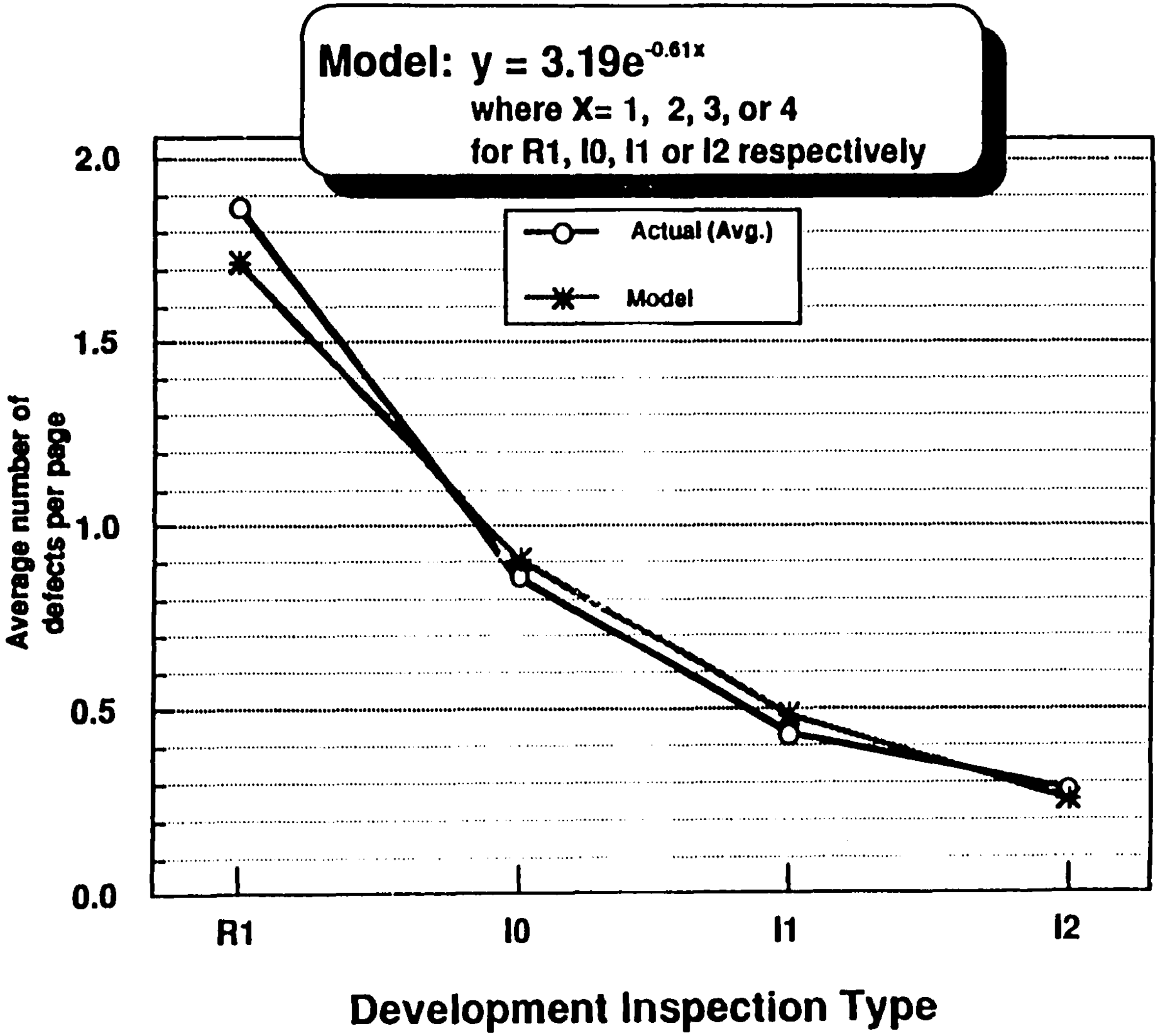
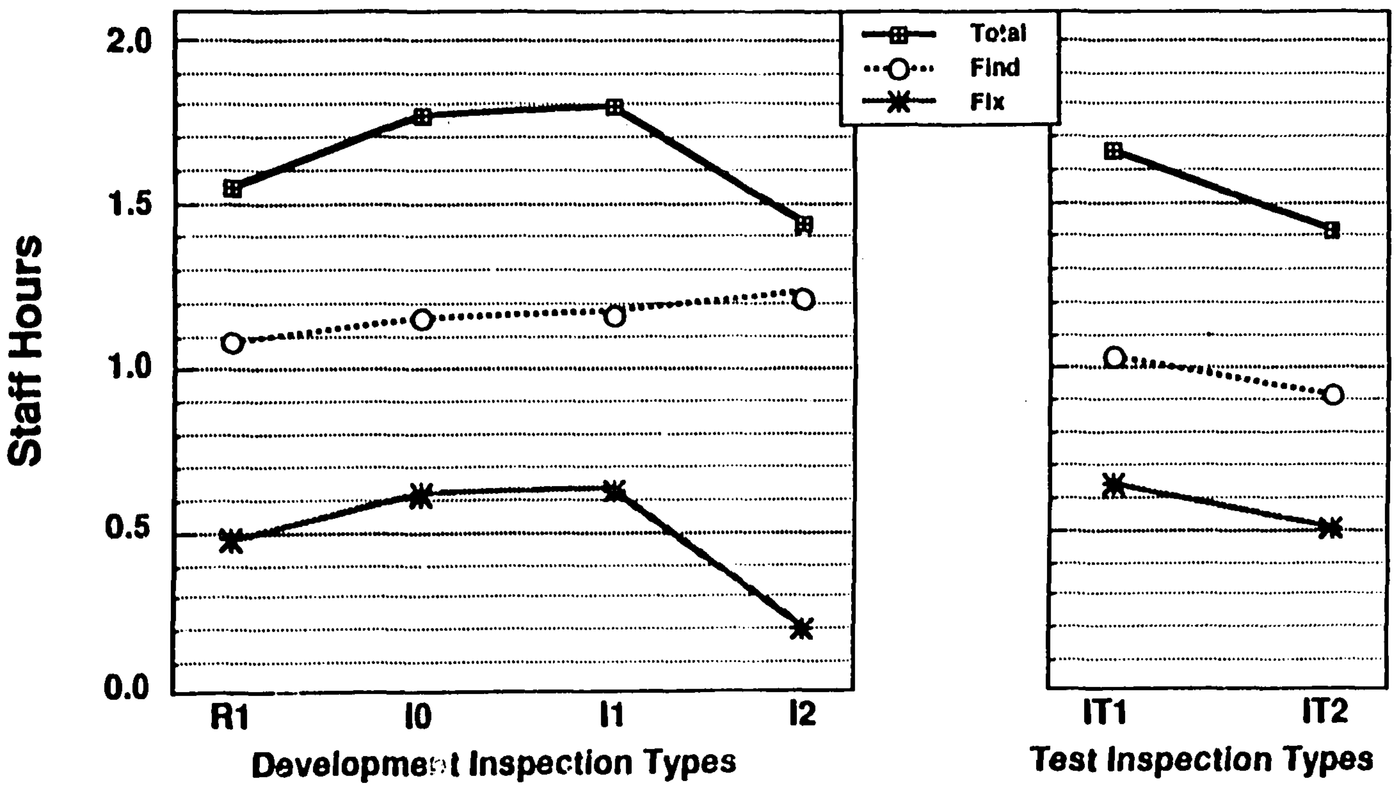
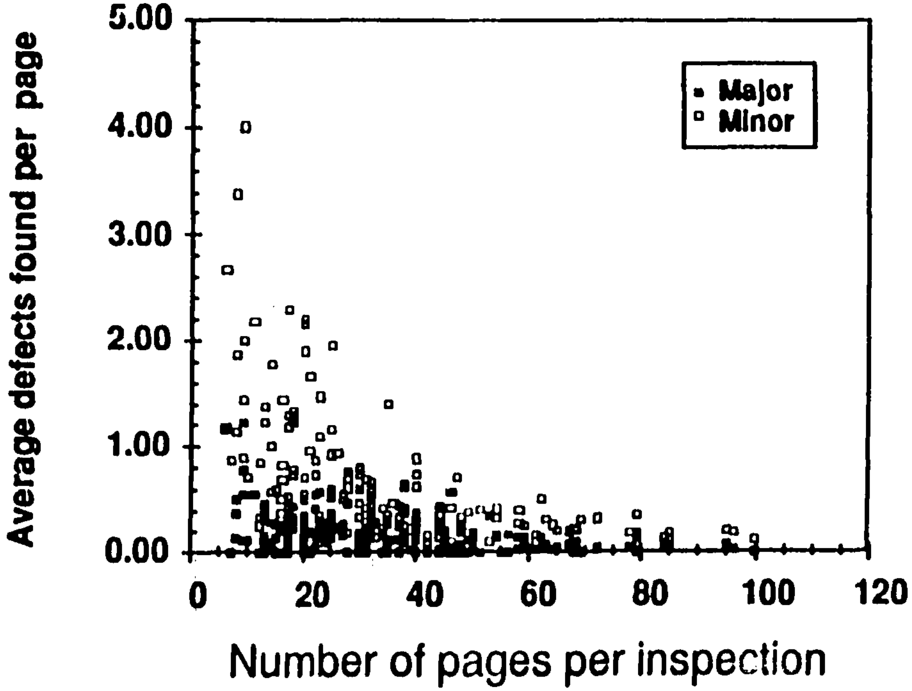


Figure 4: Staff hours per defect.



Resource hours for *finding* include all time expended during Planning, Overview, Preparation, and Meeting phases. Resource hours for *fixing* include all time expended during Rework, Third Hour, and Follow-up phases. Defects include all major and minor defects.

Figure 5. Inspection page rate versus average defects found per page



Note: Inspection "meetings" are limited to 2 hours and moderators are recommended to limit material covered to 40 pages or less.

Figure 6: Team Composition and Size by Inspection Type

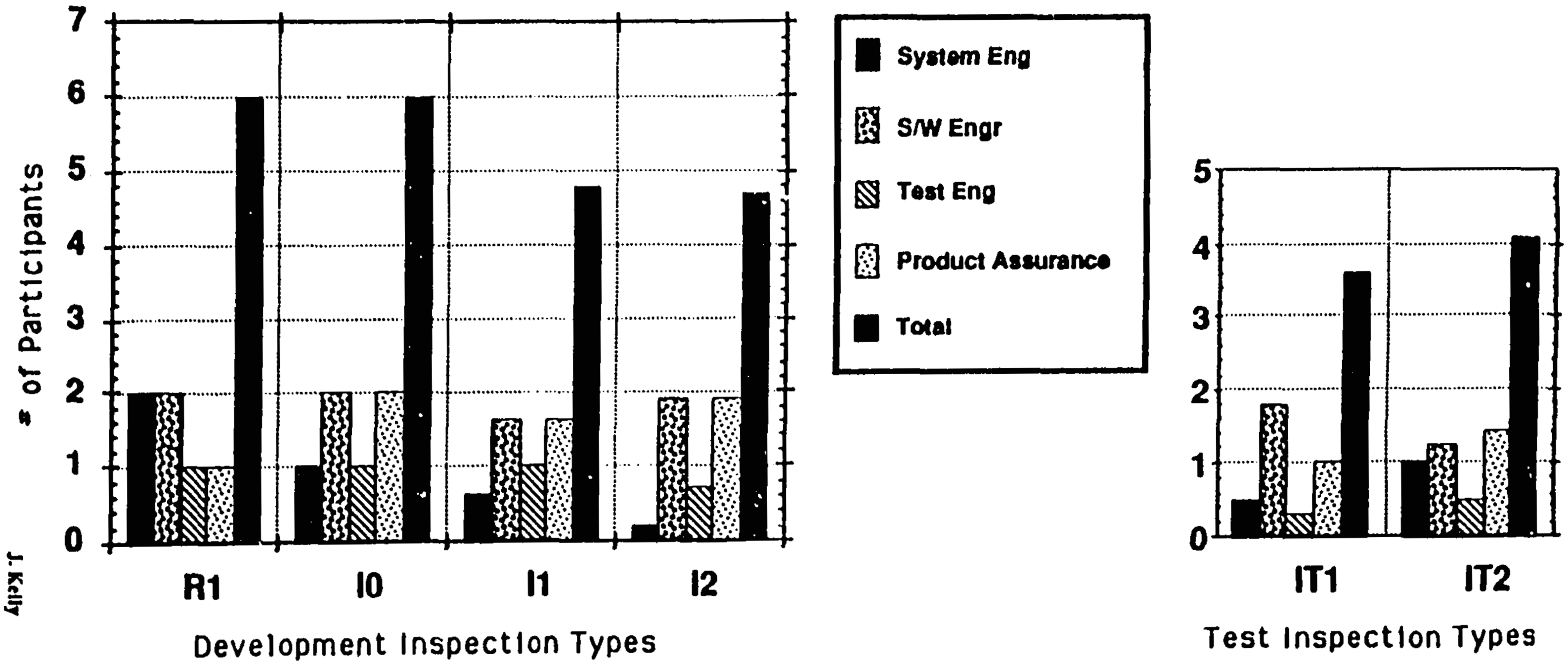
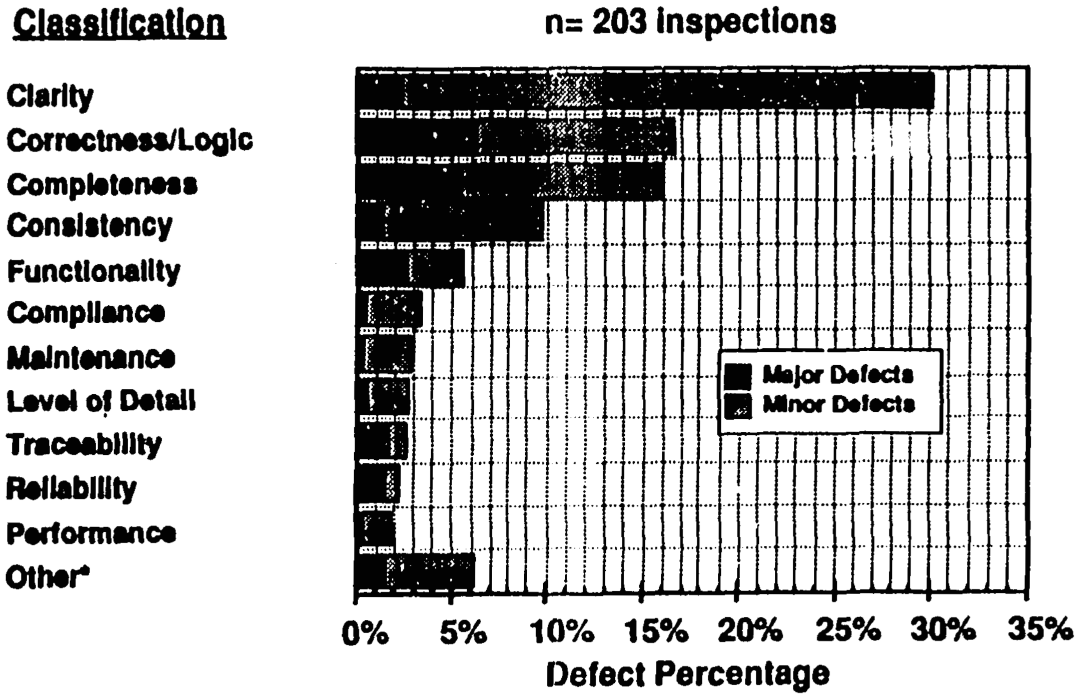


Figure 7: Distribution of defects by classification



* "Other" includes those classifications with fewer than 20 total defects.

JPL

ID# ____________

DETAILED INSPECTION REPORT

Project ____________ Date ____________

Subsystem ____________ Moderator ____________

Unit ____________ Type of Inspection ____________

PLEASE RETURN COMPLETED FORM TO J. KELLY AT MS 125-233

Defects found in Inspected Document:

Checklist Category	MAJOR			MINOR			TOTAL
	M*	W	E	M	W	E	
Clarity							
Completeness							
Compliance							
Consistency							
Correctness							
Data							
Functionality							
Interface							
Level of Detail							
Maintainability							
Performance							
Reliability							
Testability							
Traceability							
Other							
Total							

Open Issue Status

Category	Open Issue Description	Assignee

[illegible] *M = Missing W = Wrong E = Extra [illegible]

Appendix 1: Formal Inspection Detailed Report

JPL

INSPECTION SUMMARY REPORT

ID# ____________

Project ____________ Inspection Meeting Date ____________

Subsystem ____________ Follow Up Completion Date ____________

Unit(s) ____________

Type of Inspection: ☐ R0 ☐ R1 ☐ I0 ☐ I1 ☐ I2 ☐ IT1 ☐ IT2 ☐ Other ____________

Inspection Meeting: ☐ First Meeting ☐ Re-inspection Meeting # Participants ______ Meeting Clock Hours (X.X) ______

Size of Workproduct ______ Distribution: New ______% Modified ______% Reused ______% Deleted ______%

Total Time expended in Person Hours (X.X)

	Planning	Overview	Preparation	Meeting	Rework	Follow-Up	Third-Hour	Total
Inspectors								
Authors								
Moderators								

Check All Attending Inspection Meeting:

☐ Project Engineering ☐ Testing
☐ Systems Engineering ☐ Product Assurance
☐ H/W Developement ☐ Operations
☐ S/W Developement ☐ Other ______

Defects found: Major ______ Minor ______ Defects reworked: Major ______ Minor ______

Open Items (number): Closed ______ Open ______

Comments: ____________

Distribution:

Manager ____________

Moderator ____________

Inspectors ____________

Data Mgr. *J. Kelly 125-233*

Discrepancy Reports/Change Request(s)/Waivers: ____________

STATUS: ☐ PASS ☐ RE-INSPECTION REQUIRED

____
Moderator's Signature

Appendix 2: Inspection Summary Report

Appendix 3
The 10 basic Rules of Inspections:

1. Inspections are carried out at a number of points inside designated phases of the software lifecycle. Inspections are not substitutes for major milestone reviews

2. Inspections are carried out by peers representing the areas of the life cycle affected by the material being inspected (usually limited to 6 or less people). Everyone participating should have a vested interest in the work product.

3. Management is not present during inspections. Inspections are not to be used as a tool to evaluate workers.

4. Inspections are led by a trained moderator.

5. Trained inspectors are assigned specific roles.

Appendix 3
The 10 basic Rules of Inspections:

(Continued)

6. Inspections are carried out in a prescribed series of steps (as shown in figure 1).

7. Inspection meetings are limited to two hours.

8. Checklists of questions are used to define the task and to stimulate defect finding.

9. Material is covered during the Inspection meeting within an optimal page rate range which has been found to give maximum error finding ability.

10. Statistics on the number of defects, the types of defects and the time expended by engineers on the inspections are kept.

VIEWGRAPH MATERIALS

FOR THE

J. KELLY PRESENTATION

6269-0

JPL

An Analysis of Defect Densities Found During Software Inspections

Fifteenth Annual Software Engineering Workshop
November 28-29, 1990
Goddard Space Flight Center,
Greenbelt, Maryland

John C. Kelly, Ph.D.
&
Joseph Sherif, Ph.D.

Section 522
Software Product Assurance
Jet Propulsion Laboratory
Pasadena, California

JPL

What are Software Inspections?

Software Inspections are:

- Detailed Technical Reviews
- Performed on intermediate engineering products
- A highly structured and well defined process
- Carried out by a small group of peers from organizations having a vested interest in the work product
- Controlled and monitored through metrics and checklist

JPL

Types of Software Inspections Included in this Analysis

	SAMPLE SIZE
● R1 Software Requirements Inspection	23
● I0 Architectural Design Inspection	15
● I1 Detailed Design Inspection	92
● I2 Source Code Inspection	34
● IT1 Test.Plan Inspection	16
● IT2 Test Procedures & Functions Inspection	23
TOTAL	203

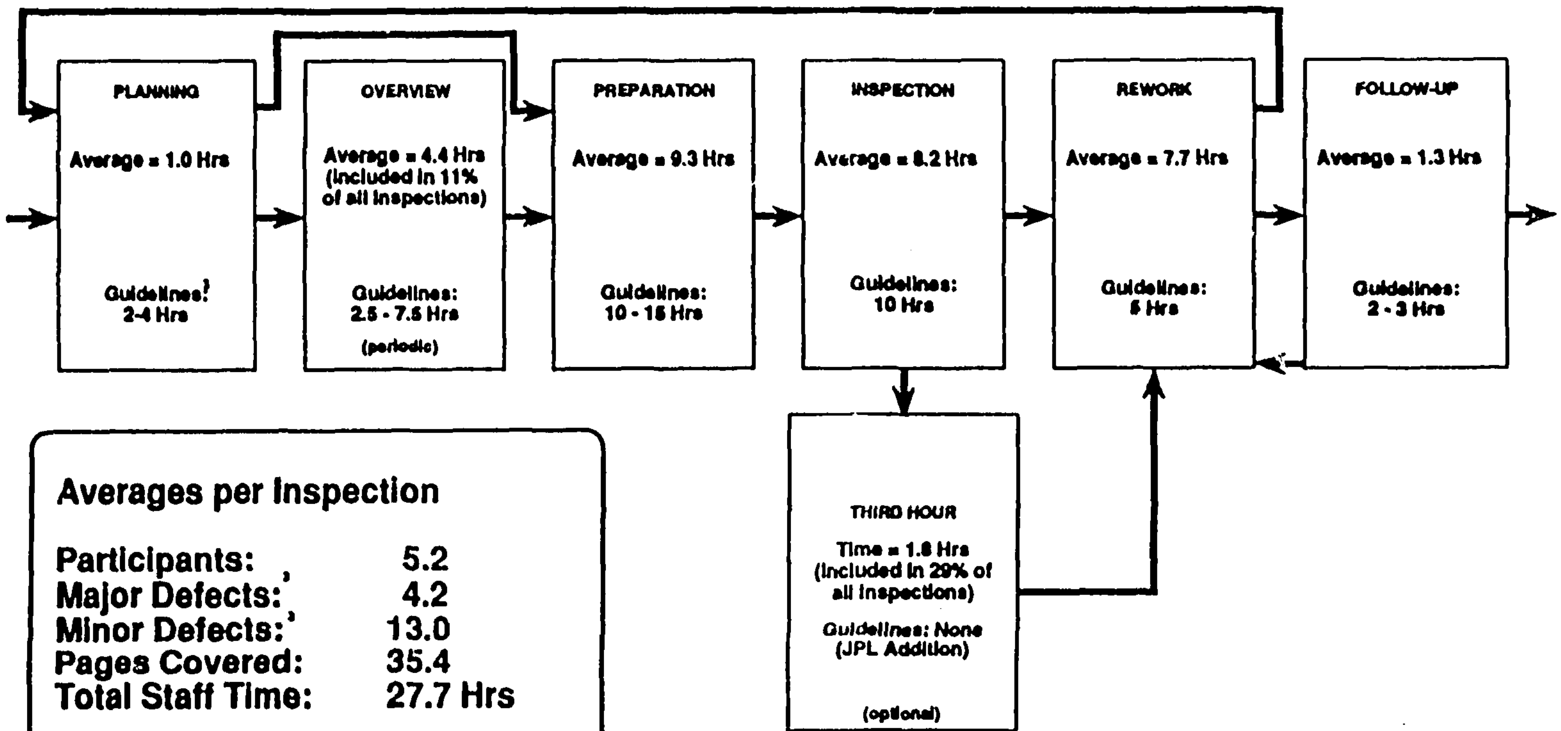
JPL

JPL Tailoring of Existing Inspection Techniques

- Participants and Team Composition
- Third Hour
- Training
- Support Documentation

JPL

Software Inspection Data Summary[1]



Averages per Inspection

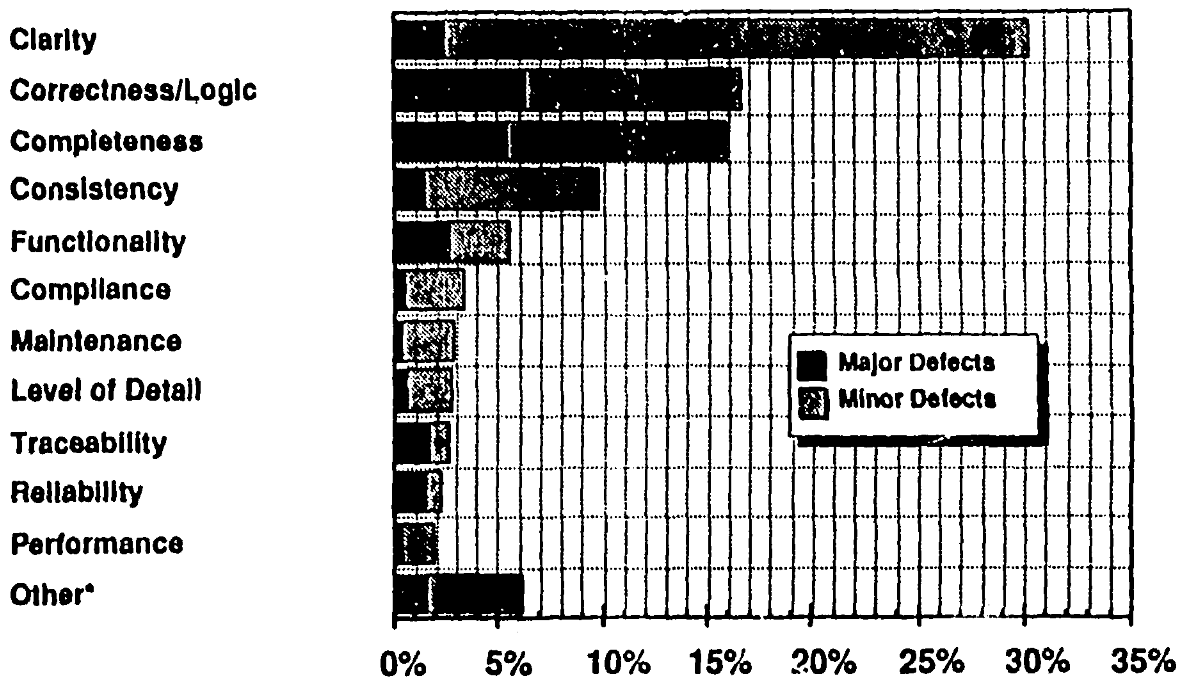
Participants:	5.2
Major Defects:[3]	4.2
Minor Defects:[3]	13.0
Pages Covered:	35.4
Total Staff Time:	27.7 Hrs

1: All times are averages from a sample of 203 JPL Inspections.
2: Guidelines were set in 2/88 based on outside organizations' experience and a team of five inspectors.
3: A <u>major defect</u> is an error that would cause the system to fail during operations, or prevent the system from fulfilling a requirement. <u>Minor defects</u> are all other defects which are non-trivial. <u>Trivial defects</u> in grammar and spelling were noted and corrected, but not included in this data analysis.

JPL

Distribution of Defects By Classification

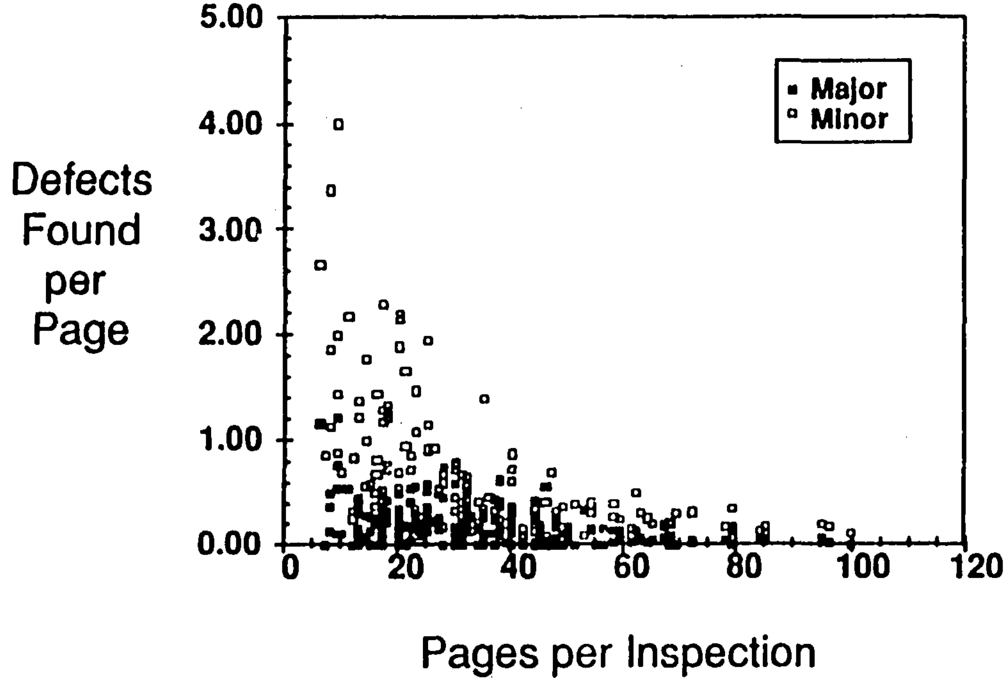
n= 203 inspections



* "Other" includes those classifications with fewer than 20 total defects.

JPL

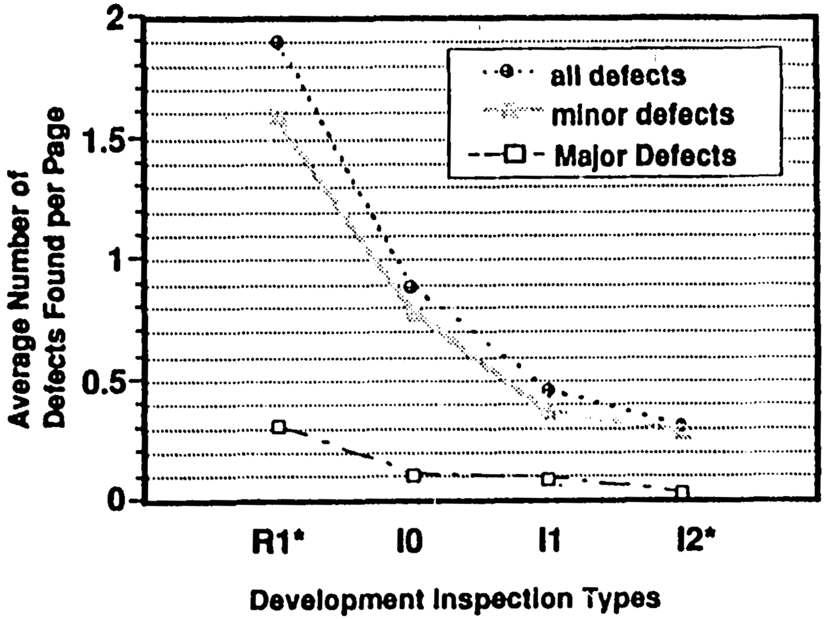
Inspection Page Rate vs. Defect Density

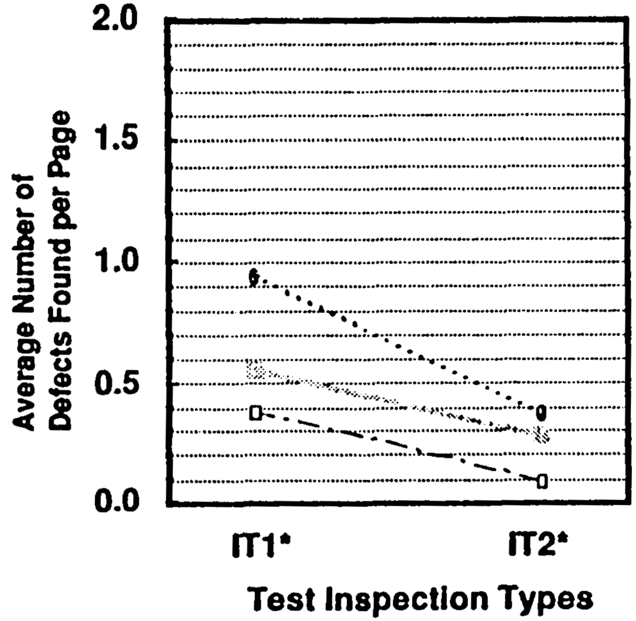


Note: Inspection "meetings" are limited to 2 hours and moderators
are recommended to limit material covered to 40 pages or less.

JPL

Defect Density vs. Inspection Type





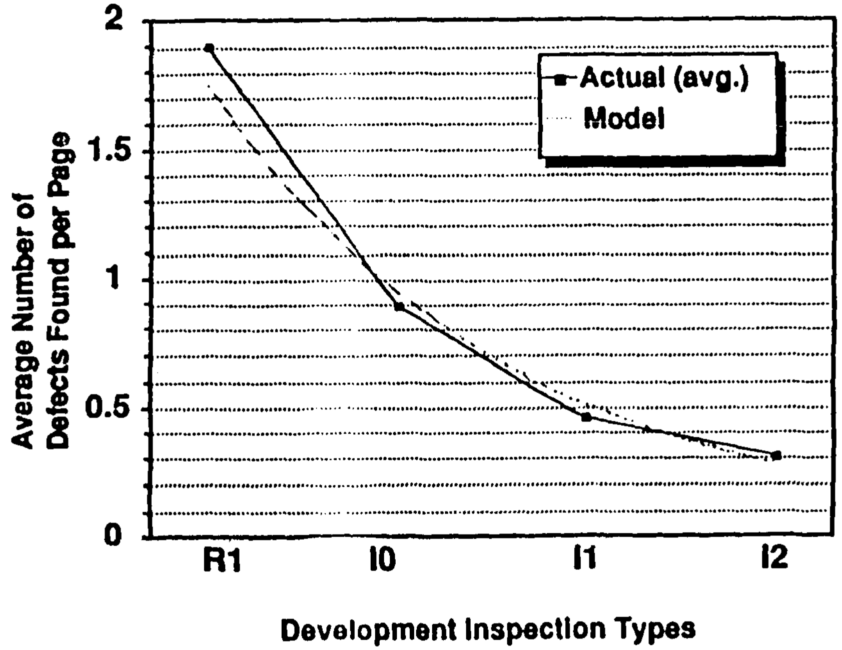
** At the alpha=0.05 level of significance ANOVA F test showed a significant difference between the defect densities of R1 and I2, and between IT1 and IT2.*

J. Kelly
NASA/JPL
Page 28 of 34

JPL

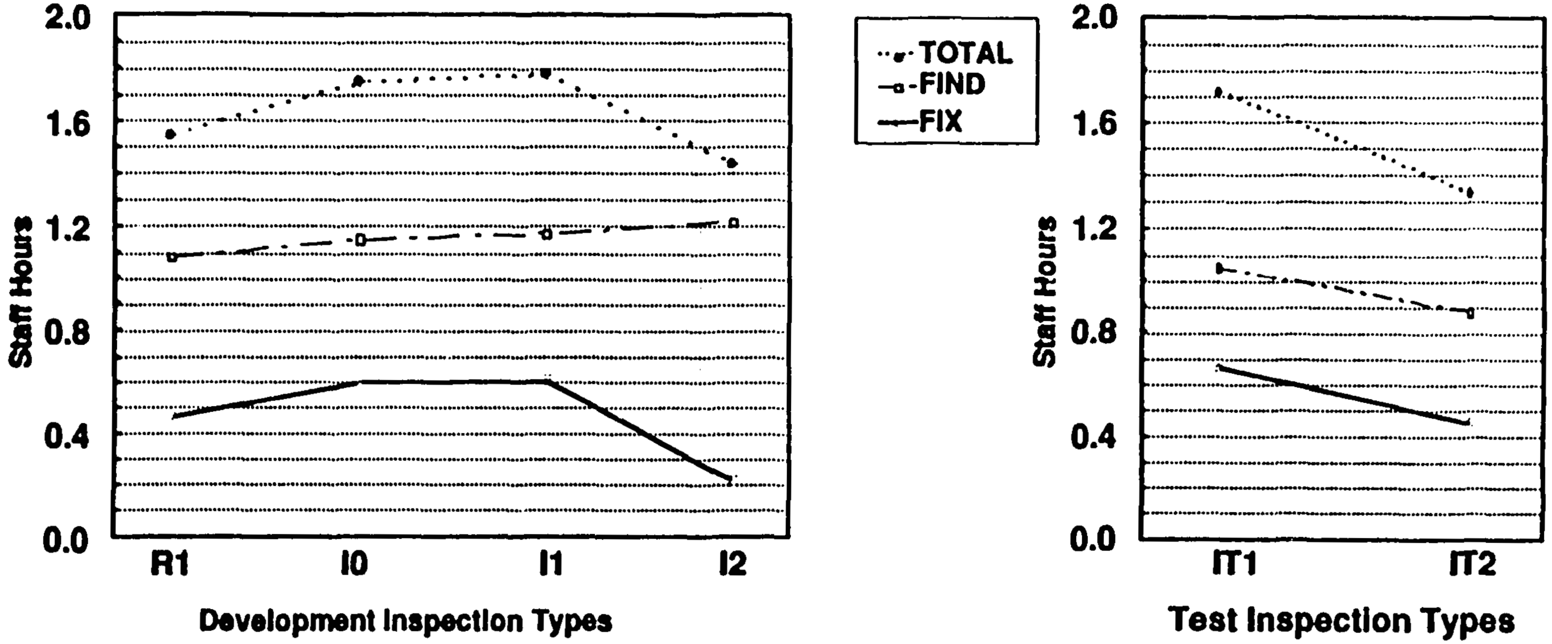
Predictive Model for Defect Density vs. Inspection Type

Model : y = 3.19 exp(-0.61x)
where x= 1,2,3, or 4
for R1, I0, I1, or I2 respectively



JPL

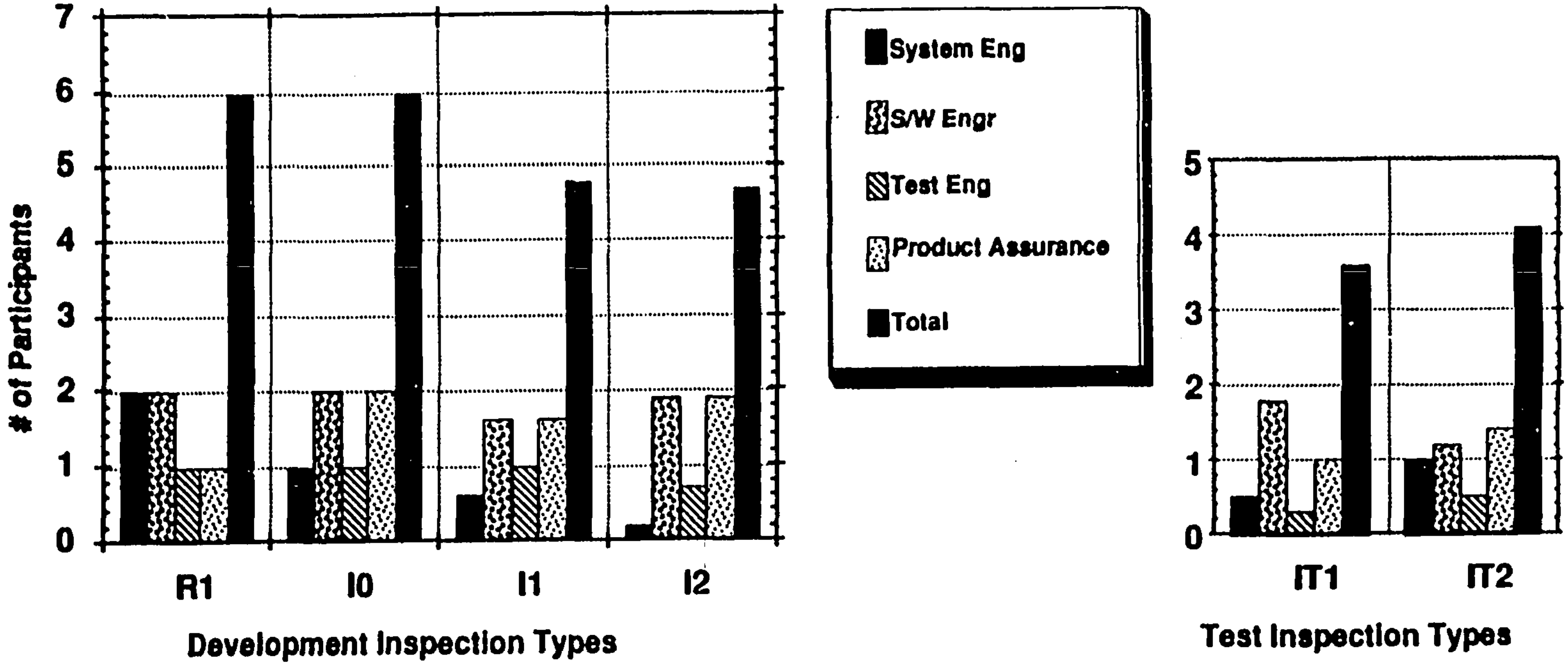
Resource Hours per Defect



In contrast, recent JPL projects reported spending an average of 5 to 17 staff hours to <u>fix</u> each defect during the <u>test phases</u>.

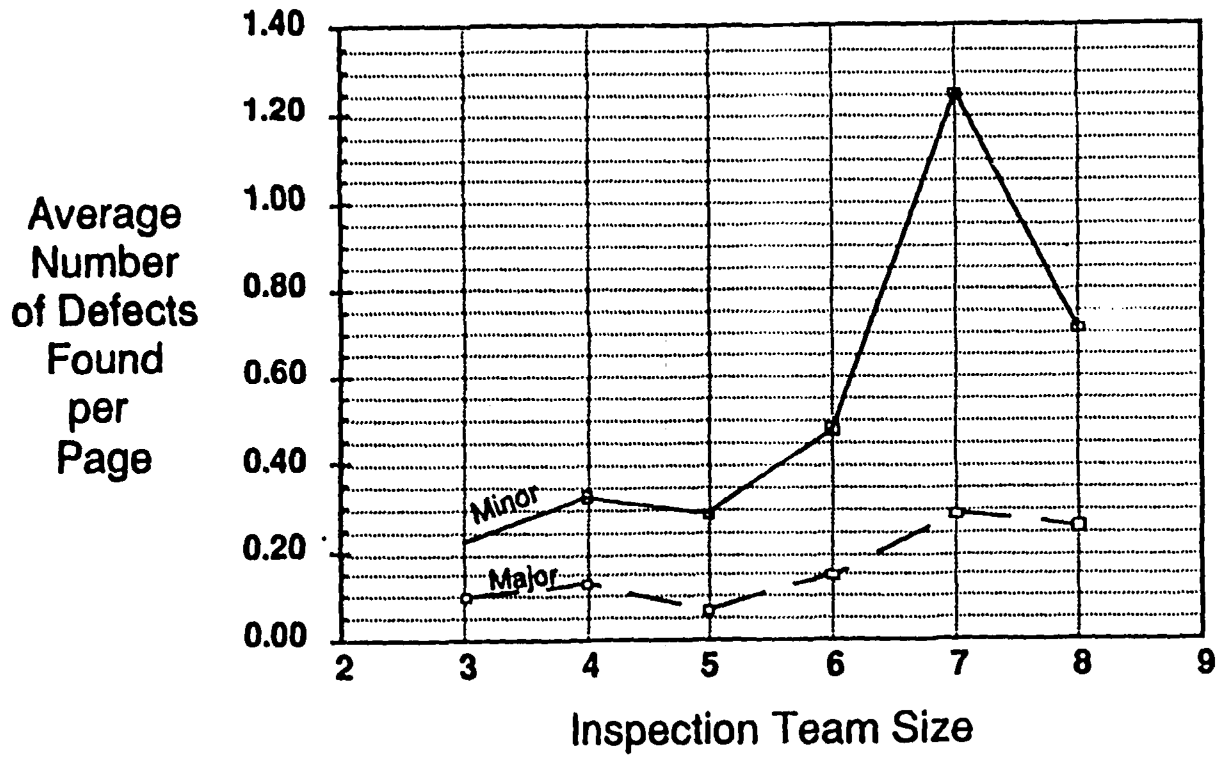
Resource hours for *finding* include all time expended during Planning, Overview, Preparation, and Meeting phases. Resource hours for *fixing* include all time expended during Rework, Third Hour, and Follow-up phases. Defects include all major and minor defects.

Team Composition and Size by Inspection Type



JPL

Team Size vs. Defect Density



Note: I1 inspections are the most frequent for team sizes 3, 4, 5, & 6
R1 inspections are the most frequent for team sizes 7 & 8

JPL

Code Inspections vs. Code Audits

	Avg. Number of Defects Found per Page		
	Major	Minor	Sample Size
Code Inspections	0.022	0.250	34
Code Audits	0.007	0.111	15

Note: 1. The work product history for code inspection sample was: 41% new, 55% reused, and 4% modified. The work product history for code audits sample was: 100% new.

2. For all types of inspections combined the average number of defects found per page was much higher than what was found in code inspections (refer to slide # 8). The overall averages were; Major = 0.119 and Minor = 0.377, for a sample size of 203 inspections.

JPL

CONCLUSIONS

- A variety of different kinds of defects are found through inspections with Clarity, Logic, Completeness, Consistency, and Functionality being the most prevalent.
- Increasing the number of pages to be inspected at a single inspection decreases the number of defects found.
- Significantly more defects were found per page at the earlier phases of the software lifecycle. *The highest defect density was observed during Requirements inspections.*
- The cost in staff hours to find and fix defects was consistently low across all types of inspections. On average it took 1.1 hours to find a defect and 0.5 hours to fix and check it (major & minor defects combined).
- Larger team sizes (6 to 8 engineers) for higher level inspections (R1 & I0) are justified by data which showed an increased defect finding capability.
- Code Inspections were superior in finding defects over Code Audits (single reviewer) by a factor of 3.