An Empirical Retrospect on Software Defects Detection Techniques

Meer Tauseef Ali[1], Dr .Syed Abdul Sattar[2]

[1.] Research Scholar,(PP.COMP.SCI.O398) Department of Computer Science, Rayalaseema University, Kurnool, A.P.

[2.] Professor, Principal, Nawab Shah Alam Khan College of Engg. & Tech., Hyderabad.

Abstract:- Inspections and testing represent core techniques to ensure reliable software. Inspections also seem to have a positive effect on predictability, total costs and delivery time. This paper will be helping in finding the solution to these questions such as issues like the effectiveness and cost-efficiency of inspections will be consider first then, The cost-effectiveness and defect profile of inspection meetings vs. individual inspections will be studied . After performing the above two issues there is a need to find the relation between complexity / modification-rate and defect density, and finally need to specify whether the defect density for modules can be predicted from initial inspections over later phases and deliveries.

These studies revealed that inspections indeed are the most cost-effective verification technique. Inspections tend to catch 2/3 of the defects before testing Inspection meetings were also cost-effective over most testing techniques, so they should not be omitted. Inspection meetings also found the same type of errors (Major, Super Major) as individual inspections.

Keywords:- Predictability, defect density effectiveness and cost-efficiency.

1. INTRODUCTION:

The software development process is made up of a number of activities: analysis, design, development, testing, and maintenance. Although the main activities in these phases are different from each other, there are internal relationships among them. These activities should not be treated in isolation and should not be limited to a specific phase. This is especially true with software defect detection. Although in different phases the activities of software defect detection are given different names: inspection, testing, or maintenance; intrinsically they are the same thing under a different name. So to improve software defect detection, all of the three activities: inspection, testing, and maintenance, should be addressed as a whole.

1.1 The Need of a New Approach

As we can see that most of the current research treats inspection and testing in isolation, with some solely focuses on inspection and others on testing. Although some research relates to both inspection and testing, they treat these two as rivals, rather than complements of each other.

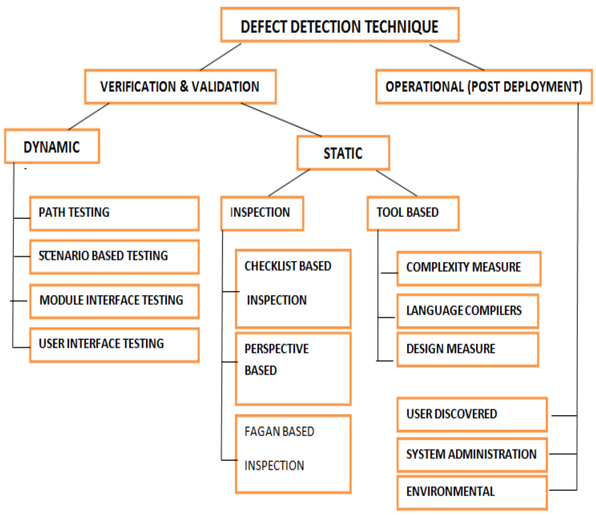

Fig 1: Defect Detection Technique.

They compared different types of inspection and testing, and tried to draw a conclusion on which one is more effective than the other. As well, the third essential defect detection technique,(Fig 1) maintenance, was ignored by the research. As a result, the research fails to realize the relationship between inspection, testing, and maintenance, and fails to give guidance on controlling, evaluating, or improving the defect detection process

2. RELATED LITERATURE:

Quality in terms of reliability is of crucial importance for most software systems. This paper only refers to general test methods; Inspections were systematized by Fagan [1] [2] and represent one of the most important quality assurance techniques. Inspections prescribe a simple and well-defined process, involving group work, and have a well-defined metrics. Inspections can be applied on most documents, even requirements. [3] They also promote team learning, and provide general assessment of reviewed documents.

2.1 Defect Detection Activities

A defect might originate in one development stage and be detected in the same or a later stage. For instance, a missing interface in a design specification could propagate to the coding stage, resulting in missing functionality in the code. You might detect this design defect during a design inspection, code inspection, unit test, function test, or system test. Because defect detection focuses on abstraction levels, we consider that primary defect detection activities are at the same level of abstraction and secondary defect detection activities are at a different level. For example, for design defects, design inspection and functional testing are primary activities, and code inspection and unit testing are secondary defect detection activities.

2.2 Evaluation criteria

What are the criteria for selecting techniques? Should you choose the most effective or the most efficient method? Efficiency in this context means the number of defects found per time unit spent on verification, and effectiveness means the share of the existing defects found.

2.3 Effectiveness and Efficiency

Absolute levels of effectiveness of defect detection techniques are remarkably low. In all but one experimental study, the subjects found only 25 to 50 percent of the defects on average during inspection, and slightly more during testing (30 to 60 percent). This means that on average, more than half the defects remain! The Berling and Thelin case study reported 86.5 percent effectiveness for inspections and 80 percent effectiveness for testing.[4] However, these are based on an estimated number of defects that the technique could possibly find, not on the total number of defects in the documents.

The experimental studies found 1 to 2.5 defects per hour. The size of the artifacts was at most a few hundred lines of code. This is small from an industrial perspective, where professionals deal with more complex artifacts, struggle with more communication overhead, and so on. Consequently, the efficiency in the industrial case studies is lower: 0.01 to 0.82 defects per hour. The variation is also much larger, which might be due to different company measures of efficiency.[3]

The practical implication of the primary defect detection methods' low effectiveness and efficiency values is that secondary detection methods might play a larger role.

3. ORGANIZATION OF THE STUDY

This paper presents results from these two studies of inspection and testing:

3.1 Threats to internal validity:

We have used standard indicators on most properties (defect densities, inspection rates, effort consumption etc.), so all in all we are on agreed ground. However, w.r.t. Module complexity we are unsure, and further studies are needed. Whether the recorded defect data in the Organization database are trustworthy is hard to say. We certainly have discovered inconsistencies and missing data, but our confidence is pretty high.

3.2 Threats to external validity:

Since Organization has standard working processes world-wide, we can assume at least company-wide relevance. However, many of the findings are also in line with previous empirical studies, so we feel confident on general level.

3.3 Validity of the empirical results

We can analyze threats to the validity of empirical studies along four dimensions: internal, conclusion, construct, and external validity.[5] From a practitioner point of view, external validity is the most important.[6] From a researcher point of view, internal validity is traditionally considered the key to successful research. However, it's important to balance all dimensions of validity to achieve trustworthy empirical studies.

On the other hand, the inconclusive results indicate the presence of factors that weren't under experimental control. By using established analysis methods, the studies also seem to limit conclusion validity threats.

This paper presents the results from the two studies described in the previous section and tries to conclude the questions and hypothesis stated in next section. Two definitions will be used throughout this section, effectiveness and efficiency:

Effectiveness: the degree to which a certain technique manages to find defects, i.e. diagnosed defect rate (defects per "volume-unit"), regardless of cost.

Efficiency: cost-efficiency (defects found per time-unit) of the above.

4. The Evaluation and Results

Our research questions and hypotheses deal with the three things, namely the role of inspection vs. testing in finding defects, e.g. their relative effectiveness and cost. The relation between general document properties and defects densities of individual modules through phases and deliveries

The different types of documents are presented in the table 1 below:

Table 1: Types of. Document

Document type	Application Information
ADI	Adaptation Direction
AI	Application Information
BD	Block Description
BDFC	Block Description Flow Chart
COD	Command Description
FD	Function Description
FDFC	Function Description Flow Chart
FF	Function Framework
FS	Function Specification
FTI	Function Test Instruction
FTS	Function Test Specification
IP	Implementation Proposal
OPI	Operational Instruction
POD	Printout Description
PRI	Product Revision Information
SD	Signal Description
SPL	Source Parameter List
SPI	Source Program Information

Questions and hypotheses

4.1 Observation

O1: How cost-effective are inspections?

4.2 Questions

Q1: Are inspections performed at the recommended rates?

Q2: How cost-efficient are the inspection meetings?

Q3: Are the same kind of defects found in initial inspection preparations and following inspection meetings?

4.3 Hypotheses

The hypothesis is paired: one null hypothesis, $\mathbf{H}_0$, which is the one that will actually be tested and an alternative hypothesis, $\mathbf{H}_a$, which may be considered valid if the null hypothesis is rejected. For the statistical tests presented in this paper, a significance of level of 0.10 is assumed.

We will present two hypotheses. In each case, the null hypothesis will represents the positive conclusion, and the alternative hypothesis will conclude with the opposite. The three hypotheses are: **H1:** Is there a significant, positive correlation between defects found during field use and document complexity?

H2: Is there a significant, positive correlation between defects found during inspection/test and module complexity?

Is there a significant correlation between defects rates across phases and deliveries for individual documents/modules? (i.e. try to track "defect-prone" modules).

4.1 O1: How cost-effective are inspections?

In this section we shall describe, and compare the efficiency of inspections and testing at the Organization Table 2 is taken from Study 1 and shows the effectiveness of inspections and testing, by comparing the number of defects found per hour.

Table 2: Total defects found, Study 1.

Activity	Number of Defects	Defect percentage
Inspection preparation	1206	61.78
Inspection meeting	38	1.98
Desk check (code review + unit test)	525	26.89
Function test	116	5.93
System test	22	1.13
Field use	46	2.33
Total	1953	100

Table 2 shows that inspections are the most effective verification activity, finding almost (61.78+1.98=63.76) 64% of total defects found in the project. Second best is the desk check that finds almost 27%.

To analyze which of the verification activities that are most effective, the time spent on the different activities was gathered. Table 3 shows the time spent on the six verification activities.

Table 3: Cost of inspection and testing, defects found per hour, Study 1.

Activity	No. of Defects	Effort	Time spent to find one defect [h:m]	Time spent on defect fixing [h]	Estimated saved time by early defect removal
Inspection preparation	1206	1023	00:51	404.3	10654.8
Inspection meeting	38	488	12:51		
Unit test	525	1654	03:09	--	--
Function test	116	9000	77:35	--	--
System test	22	--	--	--	--
Field use	46	--	--	--	--

When combining effort and number of defects, inspections proved to be the most cost-effective. Not surprisingly, function test is the most expensive activity. It should be noted that only human labour is included for desk check and function test. The costs of computer hours or special test tools are not included. Neither is the human effort in designing the test cases.

Table 4: Total defects found, Study 2.

Activity	Number of Defects	Defect percentage
Inspection preparation	5821	71.08
Inspection meeting	510	6.22
Desk check	1032	13.21
Emulator test	777	9.49.
Total	8190	100

In Study 2 it was not possible to get defect data from function test, system test and field use. Instead the data made it possible to split up the desk check, which actually consist of code review and unit test. Table 4 shows the results.

Again, the data show that inspections are highly effective, contributing to 71 % of all the defects found in the projects. Desk check is second best, finding almost 13 % of the defects in the projects. Compared to Study 1, there is an improvement in the inspection meeting, whose effectiveness has increased from 2% to 6%. Table 5 shows the effort of the different activities from Study 2.

Table 5: Cost of inspection and testing, defects found per hour, Study 2.

Activity	No. of Defects	Effort	Time spent to find one defect [h:m]	Time spent on defect fixing [h]	Estimated saved time by early defect removal
Inspection preparation	5821	7232	01:15	15258	53907.1
Inspection meeting	510	4180	08:12		
Desk check	1032	3172	01:56		-
Emulator test	777	5704	07:20		-

The inspection meeting itself has a much better cost efficiency in Study 2 (8 h : 12 min per defect), compared to Study 1 (12 h : 51 min per defect).

Although desk check on code seems to be the most effective method in Study 2, this method is not as general as inspections, which can be used on almost any phase/document of the process.

In Study 2, covering 130,000 man-hours, a total of 26670 hours were spent on inspections. It has been calculated that inspections did save 53907 hours, which would have been necessary to use on correcting defects otherwise found by testing.

That is, saving of 41% of the total project effort. Study 1 covered 26,000 man-hours where 1915 hours were spent on inspections. In this study it was calculated that they saved 10654 hours.

4.2 Q1: Are inspections performed at the recommended rates?

Table 6: Planned versus actual inspection-time consumption in Study 2.

Document information	Actual	Recommended	Defects

					time			
Doc. type	No.of Doc.	Total no. of pages	Avg length of doc.	Actual time	Planning constant	Recomm.Time	Total no of defects	Defect density per page
ADI	1	9	9	47	26	26	16	1.71
AI	38	313	11	1325	94	2714	256	0.82
BD	53	1349	33	1869	52	2132	608	0.45
BDFC	70	4389	81	4590	135	7301	1043	0.24
COD	5	40	10	137	18	73	49	1.23
FD	43	1494	45	3162	49	1630	1019	0.68
FDFC	25	1166	61	1599	34	642	439	0.38
FF	18	476	34	1128	26	364	472	0.99
FS	18	317	23	1235	31	437	267	0.84
FTI	3	787	393	281	18	36	29	0.04
FTS	3	200	100	1092	18	36	57	0.29
IP	4	85	28	334	20	59	95	1.12
OPI	7	79	16	169	26	130	18	0.23
POD	5	30	7	151	26	104	38	1.26
PRI	74	757	13	2146	125	7114	519	0.69
SD	5	77	19	390	23	94	61	0.8
SPL	35	183	7	542	104	2808	90	0.49
Total	**407**	**11751**		**20197**		**25700**	**5075**	**0.43**

Here we want to see if the recommended inspection rates were applied in the inspections. The results are presented in table 6. Note that not all document types are included.

According to the recommended rates, the inspections are performed to fast (see table 6). 20197 hours are spent on inspections, whereas 25700 hours are recommended expenditure. The defect average per page is 0.43.

Study 1 also concluded with the same result, i.e. that inspections at organization are performed to fast according to recommended inspection rates.

As reported in other literature, plots on preparation rate and defect detection rate (se figure 1) shows that the number of defects found per page decreases as the number of pages (document length) per hour increases. Inspection performed to fast will then results in decreased detection rate.

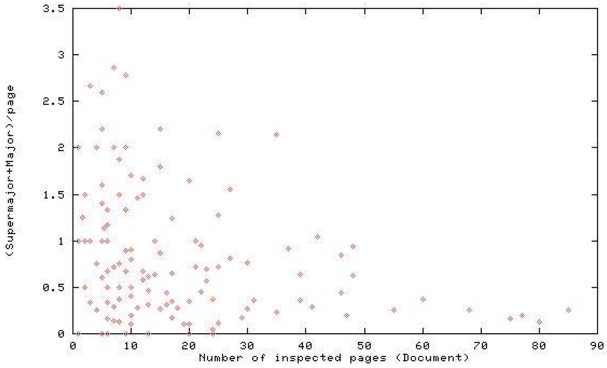

Figure 2: Number of pages inspected and defect detection rate, Study 1.

4.3 Q2: How cost-efficient are the inspection meetings?

Table 7: Time consumption for inspection, Study 2.

	Preparation	Inspection Meeting	Defect fixing	Sum
Hours	7232	4180	15258	26670
[%]	27.12%	15.67%	57.21%	100,00%

Table 7 together with figure 3, shows the time consumption for each step of the inspections from Study 2. Effort *before* individual preparation and inspection meeting has been proportionally distributed on these two activities.

Table 8: Cost efficiency and defect classification from inspections, Study 2**.**

	Major	**Super Major**	**Sum def**	**Effort**	**Efficiency**

					ects		[defects/h]
	[#]	[%]	[#]	[%]	[#]	[h]	[defects/h]
Preparation	5664	97.26	159	2.74	5823	4440	1.311
In meeting	494	96.86	16	3.14	510	2542	0.200
In defect log	6158	97.23	175	2.76	6333	6981	0.907

Table 8 from Study 2, shows the number of defects recorded in preparations, in meetings, and the total. As mentioned, the defects are classified in two categories:

- **Major:** Defects that can have a major impact later that might cause defects in the end products, and that will be expensive to clean up later.
- **Super Major:** Defects that have major impact on total cost of the project.

It turns out that 8% of defects found by inspections are found in meetings, with a cost-efficiency of 0.2 defects per hour. Compared to function test and system test, inspection meetings are indeed cost-effective in defect removal.

4.4 Q3: Are the same kind of defects found in initial inspection preparations and following inspection meetings?

We will also like to investigate what types of defects are found during preparations versus inspection meetings. Note: We do not have data on whether inspection meetings can refute defects reported from individual inspections (“false positive”), cf. [Votta93]. Our data only report new defects from inspection meetings (“true negative”). As Table 8 from Study 2, shows that 2.7% of all defects from inspections are of type Super Major, while the rest are Major.

For preparation, the Super Major share is 2.7%. For meeting the share is 3.1%, i.e. only slightly higher. We therefore conclude that inspection meetings find the same “types” of defects as by individual preparation. No such data were available in Study 1.

5.6

4.5 H1: Correlation between defects found during field use and document complexity

Intuitively, we would say that the faults in field use could be related to complexity of the module, and to the modification rate for the module. The modification rate indicates how much the module is changed from the base product, and the complexity is represented by the number of states in a module. For new modules the modification grade is zero. Correlation between modules and defect rates for each unit, (i.e., not the absolutely number of faults, but faults per volume-unit) has not yet been properly checked.

In Study 1, the regression equation can be written as:

$$\mathbf{N_{fu} = \alpha + \beta N_s + \lambda N_{mg}}$$

Where N_{fu} is number of faults in field use, N_s is number of states, N_{mg} is the modification grade, and α, β, and λ are coefficients. $\mathbf{H}_0$ can only be accepted if β and λ are significantly different from zero and the significance level for each of the coefficients is better than 0.10.

The following values were estimated:

$$\mathbf{N_{fu} = -1.73 + 0.084 * N_s + 0.097 * N_{mg}}$$

Predictor	Coefficient	St Dev	t	P
Constant	-1.732	1.067	-1.62	0.166
States	0.084	0.035	2.38	0.063
Moderate	0.097	0.034	2.89	0.034

The values for estimated coefficients are given above, along with their standard deviations, *t*-value for testing if the coefficient is 0, and the *p*-value for this test. The analysis of variance is summarized below:

Source	DF	SS	MS	F	P
Regression	2	28.68	14.34	9.96	0.018
Error	5	7.20	1.44	-	-
Total	7	35.88	-	-	-

It should be noted that the coefficients are not significant, but that the states and modification rate are significant. The F-Fisher test is also significant, and therefore the hypothesis, $\mathbf{H}_0$ can be accepted based on the results from the regression analysis.

4.6 H2: Correlation between defects rates across phases and deliveries for individual documents/modules

This hypothesis, from Study 2, uses the same data as for hypothesis 2. To check for correlation between defect densities across phases and deliveries, we have analyzed the correlation between defect densities for modules over two projects. Because the lack of data in this analysis, only Project A and Project B where used (see table 9). Table 11 shows the correlation results.

Table 11: Correlation between defect density in Project A and B, Study 2.

Variable	Defect density Project A	Defect density Project B
Defect density Project A	1	0.4672
Defect density Project B	0.4672	1

With a correlation coefficient of 0.4672, we cannot conclude that there exists a correlation between the two data sets. We had only 6 modules with complete data for both projects for this test. The test should be done again, when a larger data set are available.

5. CONCLUSION

After analysis of the data, the following can be concluded for the Organization, Software inspections are indeed cost-effective:

They find 70% of the recorded defects, cost 10% of the development time, and yield an estimated saving of 20%. I.e., finding and correcting defects before testing pays off, also here. 8% of the defects from inspections are found during the final meeting, 92% during the individual preparations. The same distributions of defects (Major, Super Major) are found in both cases. However, Gilb's [8] insistence on finding many serious defects in the final inspection meeting is hardly true.

The recommended inspection rates are not really followed: only 2/3 of the recommended time is being used. Individual inspections (preparations) and individual desk reviews are the most cost-effective techniques to detect defects, while system tests are the least effective.

The final inspection meeting is not cost-effective, compared to individual inspections, in finding defects. The identified defects in a module do not depend on the module's complexity (number of states) or its modification rate, neither during inspections nor during testing.

However, the defect density for one concrete system (Study 1) in field use correlated positively with its complexity and modification rate. We had insufficient data to clarify whether defect-prone modules from inspections continued to have higher defect densities over later test phases and over later deliveries.

The collected, defect data has only been partly analyzed by organization itself, so there is a huge potential for further analysis. The defect classification (Major and Super Major) is too coarse for causal analysis in order to reduce or prevent future defects, i.e. a process change, as recommended by Gilb. We also lack more precise data from Function test, System test and Field use. It is somewhat unclear what these findings will mean for process improvement at organization At least they show that their inspections are cost-effective, although they could be tuned w.r.t. recommended inspection rate (number of inspected pages per hour).

References:

[1] Michael E.Fagan:"Design and Code Inspection to Reduce Errors in Program Development",IBM Systems J.Vol 15,No. 3, 1976.
[2] Michael E.Fagan:"Advances in Software Inspections",IEEE Trans. on Software Engineering,SE-12(7):744--751,July 1986.
[3] Victor R. Basili, Scott Green, Oliver Laitenberger,Filippo Lanubile,Forrest Shull, Sivert Sørumgård, and Marvin V. Zelkovitz:"The Empirical Investigation of Perspective-Based Reading",39 p.,Empirical Software Engineering,1996.
[4] T.Berling and T.Thelin,"An Industrial Case Study of the Verification and Validation Activities," Proc.9th Int'l Software Metrics Symp.,IEEE CS Press,2003,pp. 226–238.
[5] C.Wohlin et al., Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers,2000.
[6] A.Rainer,T.Hall and N. Baddoo,"Persuading Developers to 'Buy into' Software Process Improvement: Local Opinion and Empirical Evidence,"Proc.Int'l Symp.Empirical Software Eng.,IEEE CS Press,2003,pp.
[7] Lawrence G.Votta:"Does Every Inspection Need a Meeting?"In Proc. ACM SIGSOFT 93 Symposium on Foundation of Software Engineering.ACM Press,December 1993.

[8] Tom Gilb and Dorothy Graham:"Software Inspections",Addison-Wesley, London, UK, 1993.