Empirical Study of Inspection and Testing Data at Ericsson, Norway

Reidar Conradi, Norw. Univ. of Technology and Science (NTNU), Trondheim, Norway #
Amarjit Singh Marjara, Cap Gemini AS, Trondheim, Norway ##
Børge Skåtevik, STC, Vatlandsvåg, Norway

p.t. Univ. Maryland, Phone +1 (301)405-1255, Fax –6638, conradi@idi.ntnu.no,
Phone +47 73.829111, amarjit.marjara@capgemini.no.

This paper was presented at PROFES'99, Oulu, Finland, 22-24 June 1999.
This is a revised version for 24th NASA Software Eng. Workshop, Washington, 1-2 Dec. 1999.

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 presents a case study of inspections and testing, done at the Ericsson development department outside Oslo in Norway. This department develops and maintains customer-defined services around AXE phone switches, i.e. the functionality around the "star"" and "square" buttons on house telephones.

AXE development at Ericsson world-wide uses a simple, local experience database to record inspections and testing data. Two MSc students from NTNU have been given access to such historical data in 1997 [Marjara97] and 1998 [Skaatevik99]. The results from these two diploma theses constitute the basis for this paper.

The paper will study questions such as:

– The effectiveness and cost-effectiveness of inspections,

– The cost-effectiveness and defect profile of inspection meetings vs. individual reading,

– The relation between complexity/modification-rate and defect density,

– Whether the defect density for modules can be predicted from inspections for later phases and deliveries.

The paper is organized as follows: Section 1 summarizes some relevant parts of the state of the art, especially of inspections. Section 2 first describes the Ericsson context, and Section 3 describes questions and hypotheses for the study. Section 4 describes the organization of the study, and Section 5 presents and discusses the results. Section 6 sums up the paper and recommends some future work.

Contents

Preface

The paper will present results from two MSc theses at NTNU, that have analyzed historical defect data at Ericsson in Oslo, Norway -- related to their AXE switches. Ericsson has practised Gilb inspections for many years, and collects defect data from inspections and testing in a small database.

These studies revealed that inspections indeed are the most cost-effective verification technique. Inspections tend to catch 2/3 of the defects before testing, by spending 10% of the development effort and thereby saving about 20% of the effort (by earlier defect correction, a ``win-win''). 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.

We also found that there is a correlation between module complexity, modification rate, and the defect density found during field-use, but not during inspections and test. Due to missing data, we

could not find out whether the defect density of modules repeated itself across inspection/test phases and over several deliveries, i.e. we could not predict ``defect-prone" modules. Defect classification was also unsatisfactory, and prevented analysis of many interesting hypotheses.

1. State of the art

Quality in terms of reliability is of crucial importance for most software systems.

Common remedies are sound methods for system architecture and implementation, high-level languages, formal methods and analysis, and inspection and testing techniques. Especially the latter two have been extensively described in the literature, and vast empirical materials have been collected, analyzed and published. This paper only refers to general test methods, so we will not comment on these here.

Inspections were systematized by Fagan [Fagan76] [Fagan86] 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. They normally produce a high success rate, i.e. by spending 10% of the development effort, we diagnose 2/3 of the defects before testing, and save 20% of the total effort -- a win-win: so “quality is free”. Inspections can be applied on most documents, even requirements [Basili96]. They also promote team learning, and provide a general assessment of reviewed documents.

Of current research topics are:

- The role of the final inspection meeting (emphasized by Tom Gilb [Gilb93], see also [Votta93].
- When to stop inspections?
- When to stop testing, cf. [Adams84]?
- The effect of root-cause-analysis on defects.
- The role of inspection vs. testing in finding defects, e.g. their relative effectiveness and cost-effectiveness.
- The relationship between general document properties and defects.
- Defect densities of individual modules through phases and deliveries.

Our research questions and hypotheses deal with the three latter.

2. The company context

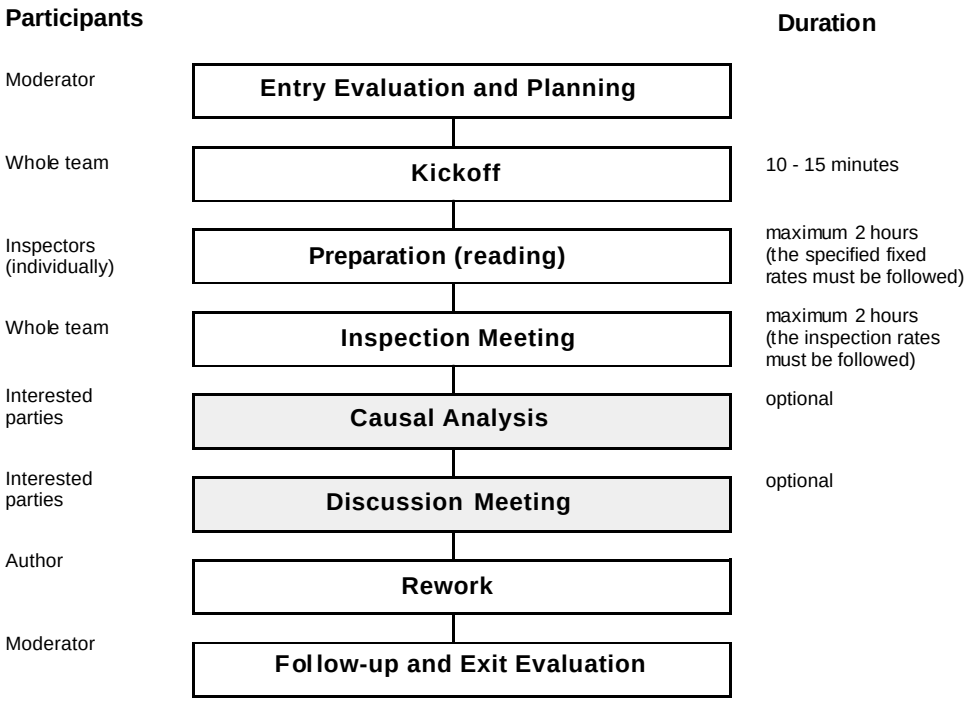
Ericsson employs about 100,000 people world-wide, whereof 20,000 in development. They have company-wide and standardized processes for most kind of software development, with adaptations for the kind of work being done. Ericsson has adopted a classical waterfall model, with so-called "tollgates" at critical decision points. In all this, verification techniques like inspections and testing are crucial. Inspection is done for every life-cycle document, although we will mostly look at design and code artifacts. Testing consists of unit test, function test and system test, where the two latter may be done at some integration site different from the development site (e.g. Stockholm).

We will only study design inspections (in-groups), simplified code reviews (by individuals) and partly testing in this paper.

The inspection process at Ericsson is based on techniques originally developed by Michael Fagan [Fagan76] at IBM and refined by Tom Gilb [Gilb93]. The process is tailor-made by the local development department. In addition there is a simplified code review done by individual developers (data from code review and unit test are sometimes merged into a “desk check"). Thus full inspections are only done upon design documents in our studies. Data from inspections/reviews and testing are collected in a simple, proprietary database and used for local tuning of the process. Defects are classified in Major, SuperMajor and Questions (the latter is omitted here) -- thus no deep categorization.

We have studied software development at the Ericsson site outside Oslo. It just passed CMM level 2 certification in Oct. 1998, and aims for level 3 in year 2000. The Oslo development site has about 400 developers, mostly working on software. The actual department has about 50 developers, and works mostly on the AXE-10 digital software switch, which contains many subsystems. Each subsystem may contain a number of modules. The development technology is SDL design language (SDT tool from Telelogic) and their proprietary PLEX language from the late 1970s (own compilers and debuggers).

Figure 1. Basic inspection process at Ericsson for design artifacts (documents).



The first level inspection process

Special inspection groups are formed, called product committees (PC), to take care of all impacts on one subsystem. In this paper, we will only look at subsystem-internal inspections, not across subsystems. The inspection process is indicated in figure 1 above, and follows Fagan/Gilb inspections wrt. overall set-up, duration etc. The number of inspectors per document is typically 3-4. Special check-lists are used for each document type.

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

Table 1. Document types (18 such).

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

Each of these document types have specific, recommended inspection rates (Skåtevik99).

3. Questions and hypotheses

3.1 One Observation

O1: How (cost-)effective are inspections and testing?

3.2 Three Questions

Q1: Are inspections performed at the recommended inspection rates?

Q2: How cost-efficient are the inspection meetings?

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

3.3 Three Hypotheses

For each question we present 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 level (*p*-level) of 0.10 is assumed.

The three alternative 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 document complexity?

H3: Is there a significant correlation between defect rates across phases and deliveries for individual documents/modules? (i.e. try to track "defect-prone" modules)?

4. Organization of the study

We have performed two studies where we have collected and analyzed historical data from software department at Ericsson in Oslo. Torbjørn Frotveit, our middleman at Ericsson, has all the time furnished us with the requested data.

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

- **Study 1:** This is the work done in a diploma thesis from 1997 [Marjara97]. Marjara investigated inspection and test data from Project A of 20,000 person-hours (14 person-years). Defect data in this work included inspection, desk check, function test, system test and partly field-use.

- **Study 2:** This is the follow-up work done in the diploma thesis from 1998 [Skåtevik99]. This thesis has data from 6 different projects (Project A-F), including the project Marjara used in Study 1. It represents over 100,000 person-hours (70 person-years). The test data in this work include only data from inspection and desk check, since later testings were done by other Ericsson divisions. However, it was possible to split desk check in code review and unit test, and data from these to activities are presented. Data from field-use are not included, due to same reasons as for function- and system test.

Threats to internal validity:

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

Threats to external validity:

Since Ericsson 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.

5. The results and evaluation of these

This chapter presents the results from the two studies described in the previous section (4), and tries to conclude the questions and hypotheses stated in section 3.
Two definitions will be used throughout this section, effectiveness and cost-effectiveness:

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

Cost-effectiveness: effort spent to find one defect.

5.1 O1: How (cost-)effective are inspections and testing?

Here we shall describe and compare the effectiveness and cost-effectiveness of inspections and testing at Ericsson in Oslo. The effort spent *before* invidual reading is proportionally distributed over inspection reading and inspection meetings. The inspection-phase effort spent *after* inspection meetings are similarly merged into "defect fixing" (se Figure 1). Table 2 is taken from Study 1 and shows the effectiveness of inspections and testing. All efforts are in person-hours, sometimes just called hours.

Table 2. Efficiency: total defects found, Study 1.

Activity	Defects [#]	[%]
Inspection reading, design	928	61.8
Inspection meeting, design	29	1.9
Desk check (code review + unit test)	404	26.9
Function test	89	5.9
System test	17	1.1
Field-use	35	2.3
Total	1502	100.0

Table 2 shows that inspections are the most effective verification activity, finding almost 64% of total defects found in the project. Second best is the desk check that finds almost 27%. We also see that 3% of the defects found by inspections are found in the meetings. To analyze which of the verification activities that are most effective, the effort spent on the different activities was gathered. Table 3 shows the effort (person-hours) spent on the six verification activities.

Table 3. Effort and cost-efficiency on inspection and testing, Study 1.

Activity	Defects [#]	Total effort on defect detection [h]	Cost-effectiveness [h:m per defect]	Total effort on defect fixing [h]	Estimated saved effort by early defect removal ("magic formulae") [h]
Inspection reading, design	928	786.8	00:51	311.2	8200
Inspection meeting, design	29	375.7	12:57		
Code review and unit test	404	1257.0	03:07	-	-
Function test	89	7000.0	78:39	-	-
System test	17	-	-	-	
Field-use	35	-	-	-	

When combining effort and number of defects, inspections proved to be the most cost-effective. Not surprisingly, function test is the most expensive activity (note: we have no effort data om system test). It should be noted that only human labor is included for desk check (code review and unit test) and function test. The costs of computer hours or special test tools are not included. Neither is the human effort spent in designing the test cases.

In Study 2 it was not possible to get defect data from function test, system test and field-use (representing 9.3% of the defects in Study 1). Instead the data made it possible to split up the desk check, which actually consist of code review and unit test (emulator test). Table 4 shows the results.

Table 4. Efficiency: total defects found, Study 2.

Activity	Defects [#]	[%]
Inspection reading, design	4478	71.1
Inspection reading, design	392	6.2
Desk check, code	832	13.2
Unit test, code	598	9.5
Total	6300	100.0

Again, the data show that inspections are highly effective, contributing to 77% 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 3% to 8% for defects found during inspections.

Table 5 shows the effort (person-hours) of the different activities from Study 2. In this study, no data from Function test or later tests were available.

Table 5. Effort and cost-efficiency on inspection and testing, Study 2.

Activity	Defects [#]	Total effort on defect detection [h]	Cost-effectiveness [h:m per defect]	Total effort on defect fixing [h]	Estimated saved effort by early defect removal ("magic formulae") [h]
Inspection reading, design	4478	5563	01:15	11737	41000
Inspection meeting, design	392	3215	08:12		
Desk check, code	832	2440	02:56		-
Unit test, code	598	4388	07:20		-

The inspection meeting itself is more cost-effective in Study 2 (8h:12min per defect) than in Study 1 (12h:57min per defect).

In Study 2 covering 100,000 person-hours, a total of 20,515 person-hours were spent on inspections (including 11,737 person-hours on defect fixing). It has been calculated that inspections did save 41,000 person-hours, which would have been necessary to locate and correct defects otherwise found by later testing. That is, a net saving of 21% of the total project effort.

Study 1 covered 20,000 person-hours where 1474 person-hours were spent on inspections (including 311.2 person-hours on defect fixing). In this study it was calculated that Ericsson saved 8200 person-hours, or a net saving of 34%!

5.2 Q1: Are inspections performed at the recommended inspection rates?

Here we want to see if the recommended inspection rates were actually applied. The results are presented in table 6. Note, that all this applies to design documents, not source code.

Table 6. Recommened rate versus actual total effort during inspections in Study 2.

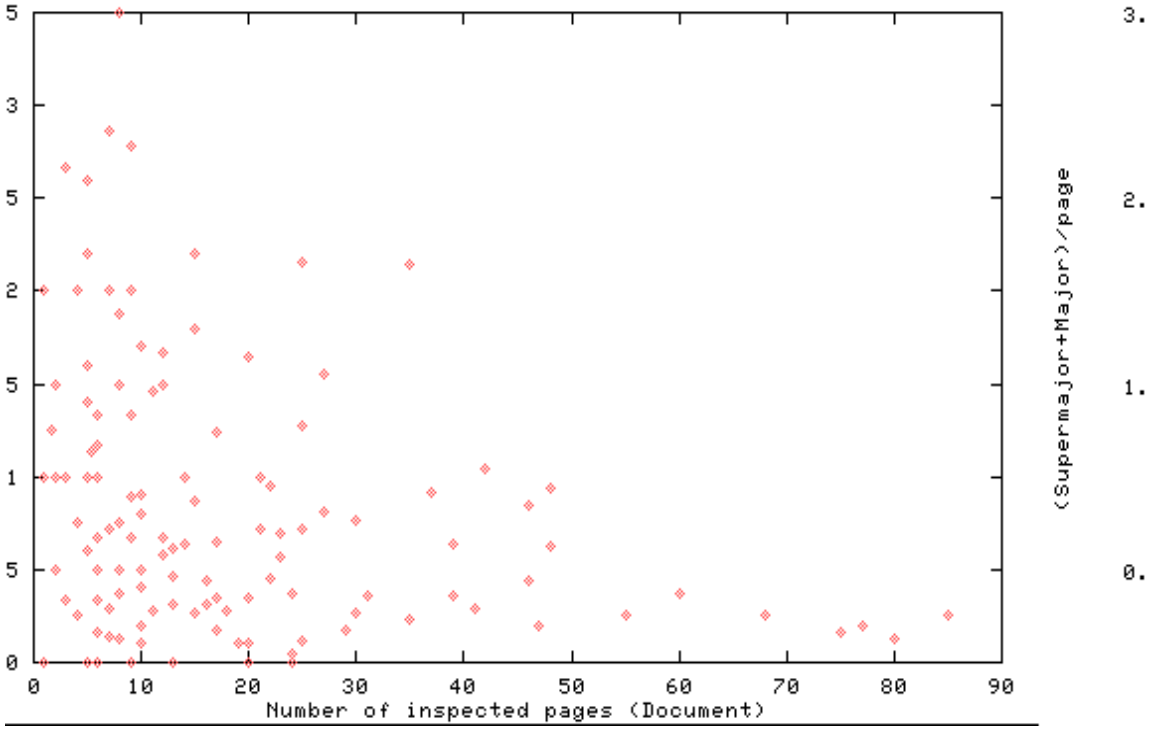
Type of effort	Total inspection effort including defect fixing [h]	Share [%]
Actual effort, Study 1	1474	54%
Recommended inspection rate, Study 1	2723	--
Actual effort, Study 2	20,515	78,6%
Recommended inspection rate, Study 2	26,405	--

Thus in Study 2, inspections are performed too fast. Only 20,515 person-hours are actually spent on inspections including defect fixing – being 78.6% of the recommended expediture of 26,405 person-hours. The average number of defects per page is 0.43.

Study 1 concluded with even more deviating results, as only 54% (1474 actual person-hours out of 2723 recommended person-hours) are totally used during inspections including defect fixing.

As reported elsewhere, plots on reading rate and defect detection rate (see figure 2) show that the number of defects found per page decreases as the number of inspected pages (document length) per hour increases. Inspection performed too fast will then result in decreased detection rate. However, we have not done any (re)analysis of "optimal" reading rates here. Also note, that the individual reading rate is a *part* of the total inspection rate mentioned e.g. in Table 6.

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



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

Table 7 shows the effort consumption for each step of the inspections including defect fixing from Study 2. Effort *before* individual reading and inspection meeting has been proportionally distributed on these two activities.

Table 7. Effort consumption for inspection and defect fixing, Study 2.

	Inspection Reading	Inspection Meeting	Defect fixing	Sum
Person-hours	5563	3215	11737	20515
[%]	27.12%	15.67%	57.21%	100.00%

Note that 57.2% of the "inspection-time effort" is spent on defect fixing in Study 2 (11,737 of 20,515 person-hours), while only 21.1% is spent on such (311.2 out of 1473.7 person-hours) in Study 1.

Table 8 from Study 2, shows the number of defects recorded in reading, in meetings, and the total.

Table 8. Cost-effectiveness and defect classification from inspections, Study 2.

	Major defects		Super Major defects		Sum defects	Defect detection effort	Cost-effectiveness
	[#]	[%]	[#]	[%]	[#]	[h]	[h:m per defect]
Inspection Reading	4356	97.2%	122	2.7%	4478	5563	01:15
Inspection Meeting	380	96.9%	12	3.1%	392	3215	08:12
Entire inspection	4736	97.2%	134	2.7%	4870	8778	01:48

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.

In Study 2, 8% of the defects found by inspections are found in the meetings, with a cost-effectiveness of 8h:12min of person-effort. Compared to function test and system test, inspection meetings are indeed cost-effective in defect removal.

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

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

For inspection reading, the Super Major share is 2.7%. For inspection 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 reading.

No such data were available in Study 1.

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

Intuitively, we would say that defects detected 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 per module (taken from a state machine diagram and reported by TeleLogic's SDL tool called SDT). For new modules the modification grade is zero. Correlation between modules and defect rates for each module (i.e., not the absolutely number of defects, but defects per volume-unit) have not yet been properly checked.

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

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

where N_{fu} is number of defects (faults) in field-use, N_s is number of states, N_{mg} is the modification grade, and α, β, and λ are coefficients. $\mathbf{H}_1$ 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:

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

Predictor	Coefficient	StDev	t	P
Constant	-1.732	1.067	-1.62	0.166
States	0.084	0.035	2.38	0.063
Modrate	0.097	0.034	2.89	0.034

Here are s = 1.200, R^2 = 79.9%, and $R^2_{(adj)}$ = 71.9%, where s is the estimated standard deviation about the regression line, R^2 is the coefficient of determination, and $R^2_{(adj)}$ is similar but adjusted for degrees of freedom. That is, if a variable is added to an equation, R^2 will get larger, even if the added variable is of no real value. To compensate for this, $R^2_{(adj)}$ is chosen as coefficient of determination.

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

In the table above, DF is the degrees of freedom, SS is the total sum of squares corrected for the mean, MS is mean sum of squares, F is the Fisher observator for F-test, and P is the significance level for this test.

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 $\mathbf{H}_1$ *can be accepted,* based on the results from the regression analysis.

5.6 H2: Correlation between defects found during inspection/test and document complexity

The relevant data come from Study 2. Because just some of the modules are found over several lifecycles, only 12 modules out of 443 could be used for this analysis. 12 modules out of 443, shows that we should probably have checked out more thoroughly relations between phases in same lifecycle, not just between different lifecycles.

Since data are collected for each document type, and each module in each phase consists of different number of document types, one document type is selected through all the phases. The document type selected is BDFC (Block Description Flow Chart). Table 9 shows the results. Field marked with "-" means that the data are missing, or no module exists. Because all the modules presented in this table only were included in project A through E, project F were excluded.

Table 9. Defect data for BDFC documents over different modules and projects, Study 2.

Module name	Project A			Project B			Project C			Project D			Project E		
	Def/page	Complexity	Defect found basic test	Def/page	Complexity	Defect found basic test	Def/page	Complexity	Defect found basic test	Def/page	Complexity	Defect found basic test	Def/page	Complexity	Defect found basic test
SUSAACA	0.04	72.0	25	0.28	80.5	3	-	-	-	-	-	-	-	-	-
SUSAACT	0.10	177.5	12	0.10	179.0	4	-	-	-	-	-	-	-	-	-
SUSCCTB	0.42	117.5	58	0.80	120.5	24	-	-	-	-	-	-	-	-	-
SUSCR	-	-	-	0.13	95.5	11	-	-	-	3.80	89.00	-	-	-	-
SUSCWC	0.29	-	23	0.10	-	-	-	-	-	-	-	-	-	-	-
SUSCWHF	-	-	11	0.50	-	-	-	-	-	-	-	-	-	-	-
SUSCWP	0.06	220.5	7	0.27	240.0	13	-	-	-	-	-	-	-	-	-
SUSSCR	0.08	244.5	22	-	295.5	34	-	-	-	-	-	-		-	-
SUSACF	0.14	47.0	32	0.37	62.5	28	-	-	-	-	-	-	0.24	66.0	-
SUSAP	0.26	67.0	42	-	-	10	-	-	-	-	-	-	0.04	78.0	-
SUSCCTA	0.34	269.5	118	-	297.5	132	1.00	299.5	3	-	-	-	-	-	-
SUSCS	0.06	257.0	14	0.90	267.5	34	0.18	254.5	21	-	-	-	-	-	-

Each project has data on defects per page found in inspections, the complexity of each module, and number of defects found in unit test (here called base test) for each block.

Hypothesis 2, uses the data presented above, and checks whether there exist a correlation between defects found during inspection/test and complexity for a module. The regression equation used to state this hypothesis can be written as:

$Y = \alpha X + \beta$, where Y is defect density, X is the complexity, and α and β are constants.

$\mathbf{H}_0$ can only be accepted if α and β are significantly different from zero and the significance level for each of the coefficients is better than 0.10. The following values were estimated:

$Y = 0.1023*X + 13.595$.

Table 10. Estimated values, Study 2

Predictor	*Estimate*	*Standard error*	*t*	*p*
β	13.595002	18.52051	0.73	0.4729
α	0.1022985	0.093689	1.09	0.2901

It indicates that the linear regression line must be rejected if a significance of level 0.10 is assumed, i.e., *neither* $\mathbf{H_2}$ *nor* $\mathbf{H_0}$ *can be refuted.* So more data is needed.

However, Ericsson reports that the best people often are allocated to develop difficult modules and more attention is generally devoted to complex software. This may explain why no significant correlation was found. More studies are anyhow needed here.

5.7 H3: Correlation between defect 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.

Correlation: 0.472	**Defect density in Project A vs. Defect density in Project B**

With a correlation coefficient of 0.4672, we cannot conclude that there exists a significant 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. So *neither* $\mathbf{H_3}$ *nor* $\mathbf{H_0}$ *can be refuted.*

6. Conclusion

After analysis of the data, the following can be concluded for Ericsson in Oslo:

- Software inspections are indeed cost-effective: They find around 70% of the recorded defects, take 6% to 9% of the development effort, and yield an estimated saving of 21% to 34%. I.e., finding and correcting defects before testing pays off – so "quality is free".

- 7% of the defects from inspections (3% in Study 1, 8% in Study 2) are found during the final meeting, while 93% are found during the individual reading. Almost the same distribution of defects (Major, Super Major) are found in both cases. However, Gilb's insistence on finding many (serious) defects in the final inspection meeting is not supported here.

- By comparison, [Votta93] reports that 8% of the defects are found in the final inspection meeting. Votta therefore proposes to eliminate them, since they are costly (7-14 times less cost-efficient than individual reading in our studies) and since their logistics is bothersome (binding up many busy people and thus victims to sudden cancellations). However, inspection meetings are indeed cost-efficient compared to function tests (6 times more cost-effective in Study 1), and presumably to later tests too. Inspection meetings also fulfill important social functions, like dissemination of knowledge and promotion of team spirit. At Ericsson they also serve to give an overall quality check or approval of design documents.

- Individual reading and individual desk reviews are the most cost-effective techniques to detect defects, while system tests are the least cost-effective.

- The recommended inspection rates are not really followed, since only 54% to 79% of the recommended effort is being used.

- 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 number of defects 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 Ericsson 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 Ericsson. At least they show that their inspections are cost-effective, although they could be tuned wrt. recommended reading rate (number of inspected pages per person-hour, as part of overall inspection rates).

On the other hand, a more fine-grained data seem necessary for further analysis, e.g. for root-Cause-Analysis (also recommended by Gilb). More detailed information is needed on “false positives” and on overlap in detected defects among inspectors to allow capture-recapture analysis. Such defect classsification seems very cheap to implement at defect recording time, but is almost impossible to add later. However, Ericsson seems rather uninterested to pursue such changes, e.g. since “approval from headquarters” is necessary to modify the current inspection process. However, due to a change in technology platform from SDL and PLEX to UML and Java, Ericsson will anyhow have to revise their inspection process towards object-oriented technologies and corresponding inspection techniques [Travassos99].

Inspired by these findings, NTNU is anyhow interested to continue its cooperation with Ericsson on defect studies in the context of the SPIQ project. Their defect database seems under-used, so these studies may encourage a more active utilization of collected data. Further, NTNU has under way further longitudinal studies at Ericsson, spanning over several development phases and release cycles.

Acknowledgements: We thank Torbjørn Frotveit and other contacts at Ericsson for their time and interest in these investigations, and the Norwegian SPIQ project on Software Process Improvement for economic support. We also thank Oliver Laitenberger from Fraunhofer IESE, Kaiserslautern for insightful comments.

7. References

[Adams84]) Edward Adams:
"Optimizing Preventive Service of Software Products",
IBM Journal of Research and Development, (1):2—14, 1984.

[Basili96] Victor R. Basili, Scott Green, Oliver Laitenberger,
Filippo Lanubile, Forrest Shull, Sivert Sørumgård, and Marvin V. Zelkovitz:
"The Empirical Investigation of Perspective-Based Reading",
J. of Empirical Software Engineering, Vol. 1, No. 2, 1996, p. 133—164.

[Fagan76] Michael E. Fagan:
"Design and Code Inspection to Reduce Errors in Program Development",
IBM Systems J. Vol. 15, No. 3, 1976, p. 182—211.

[Fagan86] Michael E. Fagan:
"Advances in Software Inspections",
IEEE Trans. on Software Engineering, SE-12(7):744—751, July 1986.

[Gilb93] Tom Gilb and Dorothy Graham:
"Software Inspections",
Addison-Wesley, London, UK, 1993.

[Marjara97] Amarjit Singh Marjara:
"An Empirical Study of Inspection and Testing Data" (Study 1),
IDI. NTNU, Trondheim, Norway, 22 Dec. 1997.
108 p., EPOS TR 308 (diploma thesis).

[Skåtevik99] Børge Skåtevik:
"An Empirical Study of Historical Inspection and Testing Data at Ericsson" (Study 2) ,
IDI, NTNU, Trondheim, Norway, 8 Feb. 1999.
90 p., EPOS TR 350 (diploma thesis).

[Travassos99] Guilherme H. Travassos, Forrest Shull, Michael Fredericks, Victor R. Basili:
"Detecting Defects in Object Oriented Designs: Using Reading Techniques to Increase Software Quality",
Proc. Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'99),
Denver, Colorado, 3—5 Nov.1999.
In ACM SIGPLAN Notices, Vol. 34, No. 10, Oct. 1999, p. 47—56.

[Votta93] Lawrence G. Votta:
"Does Every Inspection Need a Meeting?",
In Proc. ACM SIGSOFT 93 Symposium on Foundation of Software Engineering.
ACM SIGSOFT Engineering Notes, Vol. 18, No. 5, December 1993, p. 107—114.

Updated: 08 Dec. 1999 (rc), File: ~spiq/presentasjoner/artikler/NASA99-ericsson-v3.doc