Fraunhofer Einrichtung
Experimentelles
Software Engineering

# The Application of Subjective Estimates of Effectiveness to Controlling Software Inspections

**Authors:**
Khaled El Emam
Oliver Laitenberger
Thomas Harbich

Submitted for publication in
Journal of Systems and Software

IESE-Report No. 031.99/E
Version 1.0
April 12, 1999

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competetive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern

# Abstract

One of the recently proposed tools for controlling software inspections is capture-recapture models. These are models that can be used to estimate the number of remaining defects in a software document after an inspection. Based on this information one can decide whether to reinspect a document to ensure that it is below a prespecified defect density threshold, and that the inspection process itself has attained a minimal level of effectiveness. This line of work has also recently been extended with other techniques, such as the Detection Profile Method. In this paper we investigate an alternative approach: the use of subjective estimates of effectiveness by the inspectors for making the reinspection decision. We performed a study with 30 professional software engineers and found that the median relative error of the engineers' subjective estimates of defect content to be zero, and that the reinspection decision based on that estimate is consistently more correct than the default decision of never reinspecting. This means that subjective estimates provide a good basis for ensuring product quality and inspection process effectiveness during software inspections. Since a subjective estimation procedure can be easily integrated into existing inspection processes, it represents a good starting point for practitioners before introducing more objective decision making criteria by means of capture-recapture models or the Defect Detection Profile Method.

Table of Contents

# 1 Introduction

In the recent past, there has been a mushrooming of research activity in developing and improving defect content estimation techniques (DCETs) for software inspections. All of these techniques use quantitative models for estimating the number of defects in a software document from data collected after a software inspection has been carried out. The logic behind applying DCETs is that by estimating the number of defects in a document, the remaining defects can be calculated, and subsequently an objective decision can be made on whether to reinspect the software document or to let it pass to the next phase. In this manner, the document quality (defined in terms of defect density) and the inspection process quality (defined in terms of its effectiveness) can be controlled. Objectivity refers to the fact that the decision making process does not depend on human judgment.

Two classes of DCETs have been studied: capture-recapture (CR) models and the Detection Profile Method (DPM). CR models originate from wildlife research. These models have been applied in biology to the estimation of the size of animal populations, and epidemiology to estimate the size of diseased populations and birth and death rates. The same models can be applied in the context of a software inspection to estimate the number of defects in a software document.

The DPM (Wohlin and Runeson, 1998) involves fitting a curve to the data obtained from an inspection, and using that for predicting the total number of defects in a document. Further investigations have resulted in improvements to DPM and the proposal of a procedure for selection between CR and DPM (Briand et al., 1998a).

Because the major focus of this line of work was the development of objective decision criteria, another, much simpler approach has not been investigated in this context thus far: the use of subjective estimates of inspection effectiveness for making the reinspection decision. The basic concept behind this approach is to ask inspectors after an inspection to estimate the percentage of defects in a document they believe they have actually found. Using this information, one can estimate the total number of defects in a document and the remaining number of defects. In addition to its simplicity, this approach would be appealing for practitioners because, first, it requires only one inspector, perhaps the most experienced one, to make the estimate, and therefore would be applicable irrespective of the total number of inspectors. And second, it neither requires significant changes to an existing inspection implementation, such as collecting more detailed data on defects, nor any extra effort for inspection participants.

The motivation for investigating subjective estimates of effectiveness comes from the earlier work of Selby (Selby, 1985). In a study comparing code reading, functional testing, and structural testing, he noted that readers could estimate quite accurately their own effectiveness: "*This estimation of the number of faults uncovered correlated reasonably well with the actual percentage of faults detected (R = .57, alpha < .0001). Investigating further, individuals using the different techniques were able to give better estimates: code readers gave the best estimates (R =.79, alpha < .0001).... This last observation suggests that the code readers were more certain of the effectiveness they had in revealing faults in the programs.*" Although subjective, if further corroborative evidence suggests that those effectiveness estimates are accurate, then this can provide another approach for controlling software inspections and a good starting point for practitioners before introducing more objective DCETs, such as CR and DPM.

In this paper we show how subjective estimates of effectiveness can be applied to making the reinspection decision for code documents, and report on a study that empirically evaluated the usefulness of the estimates. The study was conducted with 30 professional software engineers at Bosch Telecom GmbH, Germany. The reading technique that the subjects used for defect detection was checklist-based reading (CBR) (Laitenberger et al., 1999). The reason for focusing on CBR is that in a recent literature survey checklist-based reading was found to be the de-facto standard approach for defect detection in many industrial inspection implementations (Laitenberger and DeBaud, 1998). After the defect detection step, the subjects estimated their own effectiveness, i.e., the percentage of defects they thought they have found. The subjective estimates were used in our analysis. Briefly, our results indicate that estimates of defect content based on subjective estimates of effectiveness have a median relative error of zero, and that they provide consistently better document and inspection process quality control than the current default practice of not reinspecting a document.

This paper is organized as follows. In Section 2 we present an overview of empirical work on evaluating DCETs in software engineering, and current uses of subjective estimates in software engineering. Section 3 describes how subjective estimates of effectiveness can be evaluated. Our research method is described in detail in Section 4, and our results in Section 5. We conclude the paper in Section 6 with a summary and suggestions for future work.

# 2 Background

## 2.1 Definition of Reinspections

A recent literature review found that, on average, inspections find 57% of defects in code and design documents (Briand et al., 1998c). However, given the substantial defect detection cost savings that can be accrued by increasing the effectiveness of inspections (Briand et al., 1998c), contemporary research has primarily focused on improved reading techniques, for example, (Laitenberger et al., 1999), and on reinspections, for example, (Eick et al., 1992), for maximizing inspection effectiveness. Our focus here is on reinspections.

A reinspection, as referred to in this paper, is intended to scrutinize an already inspected document anew. The purpose is to identify defects that have been missed during the initial inspection. It is not to focus on the changes made due to the initial inspection.

Some inspection implementations involve a follow-up phase at the end of the inspection process. Fagan (Fagan, 1976) reports that this inspection phase aims at verifying whether the author has taken some remedial action for each issue, problem, and concern detected. He also states that the follow-up phase is an optional one in the inspection process and that it cannot be considered a reinspection.

In their book on software inspections, Strauss and Ebenau (Strauss and Ebenau, 1994) describe the reinspection stage. However, the focus of this is to concentrate on the changes made after the initial inspection, their interfaces and dependencies. This is different from performing a reinspection to identify defects that have been missed.

Adams (Adams, 1999) notes that the occurrence of a reinspection should be a trigger for preventive action since excessive use of reinspections is likely to be inefficient. He suggests improved entrance criteria for inspections, use of preliminary informal inspections, and some form of root cause analysis.

The criteria for making the reinspection decision should be formulated to control:

- **The quality of the document.** By ensuring that the *documents* that pass the inspection attain a minimal quality level. A document quality level would be the remaining defect density.

- **The quality of the inspection.** #Z FOTVSJOH UIBU UIF *inspection process* IBT BDIJFWFE B NJOJNBM RVBMJUZ MFWFM 5IF QSPDFTT RVBMJUZ MFWFM XPVME CF EFGJOFE BT UIF QSPQPSUJPO PG EFGFDUT JO UIF EPDVNFOU UIBU IBWF CFFO JEFOUJGJFE CZ UIF JOTQFDUJPO

.PTU PSHBOJ[BUJPOT IBWF OPU JOTUJUVUJPOBMJ[FE SFJOTQFDUJPOT BT B UFDIOJRVF GPS FOTVSJOH UIF RVBMJUZ PG UIF EPDVNFOU BOE UIF JOTQFDUJPO QSPDFTT 5IPTF UIBU EP IBWF VUJMJ[FE GPS FYBNQMF IJTUPSJDBM OPSNT GPS NBLJOH UIF SFJOTQFDUJPO EF DJTJPO *G UPP NBOZ EFGFDUT BSF GPVOE DPNQBSFE UP UIF OPSN UIFO UIJT JT UBLFO BT FWJEFODF PG B QPPS EPDVNFOU XIJMF UPP GFX BSF UBLFO BT FWJEFODF PG B QPPS JOTQFDUJPO &JDL FU BM )PXFWFS UIJT BQQSPBDI BTTVNFT UIBU WBSJB UJPOT BNPOH SFWJFXT BSF MBSHFS UIBO WBSJBUJPOT BNPOH EPDVNFOUT *G UIJT JT OPU UIF DBTF UIFO UIJT DBO MFBE UP SFJOTQFDUJPOT PG IJHI RVBMJUZ EPDVNFOUT BOE MPX RVBMJUZ EPDVNFOUT NBZ FBTJMZ QBTT OUIFS DPNQBOJFT NBZ IBWF GPMMPXFE B SVMF PG UIVNC UP EFUFSNJOF XIFUIFS UP QFSGPSN B SFJOTQFDUJPO OOF JT UIBU UIF OVNCFS PG EFGFDUT SFNBJOJOH JT SPVHIMZ FRVBM UP UIF OVNCFS PG EFGFDUT GPVOE QFS QBHF (JMC BOE (SBIBN 5IJT SVMF PG UIVNC IPXFWFS EPFT OFJUIFS DPOTJEFS UIF RVBMJUZ MFWFM PG UIF EPDVNFOU OPS UIF RVBMJUZ PG UIF JOTQFDUJPO .PSFPWFS UIFSF JT OP TZTUFNBUJD FWJEFODF PO UIF WBMJEJUZ PS BDDVSBDZ PG UIJT SVMF

8PSL JO UIF %$&5 BSFB IBT BJNFE BU QSPWJEJOH UIF NFBOT GPS NBLJOH UIF SFJO TQFDUJPO EFDJTJPO 8F SFWJFX UIJT XPSL CFMPX

## 2.2 An Overview of DCETs in Software Engineering

OOF DBO VTF DBQUVSF SFDBQUVSF $3 NPEFMT UP EFDJEF PO BO PCKFDUJWF CBTJT XIFUIFS PS OPU UP QFSGPSN B SFJOTQFDUJPO $3 NPEFMT XFSF JOJUJBMMZ EFWFMPQFE UP FTUJNBUF UIF TJ[F PG BOJNBM QPQVMBUJPOT 4FCFS OUJT FU BM 8IJUF FU BM *O EPJOH TP BOJNBMT BSF DBQUVSFE NBSLFE BOE UIFO SF MFBTFE PO TFWFSBM USBQQJOH PDDBTJPOT 5IF OVNCFS PG NBSLFE BOJNBMT UIBU BSF SFDBQUVSFE BMMPXT POF UP FTUJNBUF UIF UPUBM QPQVMBUJPO TJ[F CBTFE PO UIF TBN QMFT PWFSMBQ $3 NPEFMT BSF BMTP BQQMJFE JO FQJEFNJPMPHZ UP FTUJNBUF UIF TJ[F PG B EJTFBTFE QPQVMBUJPO 4UFQIFO BOE GPS UIF FTUJNBUJPO PG CJSUI BOE EFBUI SBUFT $IBOESB 4FLBS BOE %FNJOH XIFSF JOTUFBE PG NVMUJQMF USBQQJOH PDDBTJPOT NVMUJQMF SFHJTUSBUJPO TZTUFNT BSF VTFE

5IF CBTJD JEFB CFIJOE B $3 NPEFM DBO CF JMMVTUSBUFE GPS UIF DBTF PG UXP PDDB TJPOT XJUI SFGFSFODF UP 5BCMF TFF 8JDLFOT 4VQQPTF POF XBOUT UP FTUJNBUF UIF TJ[F $M$ PG BO BOJNBM QPQVMBUJPO UIBU EPFT OPU DIBOHF PWFS UJNF J F OP BOJNBMT FOUFS PS MFBWF UIF QPQVMBUJPO UISPVHI CJSUI EFBUI JNNJHSB UJPO PS FNNJHSBUJPO " OVNCFS $m_{1+}$ BOJNBMT BSF DBQUVSFE PO UIF GJSTU EBZ 5IFTF BOJNBMT BSF NBSLFE TPNFIPX BOE SFMFBTFE JOUP UIF QPQVMBUJPO "GUFS BMMPXJOH TPNF UJNF GPS UIF NBSLFE BOE VONBSLFE BOJNBMT UP NJY B TFDPOE trapping occasion is performed on a second day. On this day, $m_{+1}$ animals are captured. This sample of $m_{+1}$ animals consists of $m_{11}$ animals bearing a mark (animals captured on both days) and $m_{21}$ animals without a mark (newly captured animals). There are also $m_{12}$ animals that were marked on the first day but were not caught on the second day. The values in parantheses are unknown. Therefore we do not know $m_{22}$ which is the number of animals not found in either occasion.

| | | Found on Occasion 2 Yes | No | |
|---|---|---|---|---|
| Found on Occasion 1 | Yes | $m$ | $m_2$ | $m_+$ |
| | No | $m_2$ | $(m_{22})$ | $(m_{2+})$ |
| | | $m_+$ | $(m_{+2})$ | $(M)$ |

Table 1: Incomplete contingency table with observed values of defects found on two occasions.

The odds ratio for such a table can be estimated by:

$$\hat{\alpha} = \frac{m_{11}m_{22}}{m_{12}m_{21}} \quad \text{Eqn. 1}$$

Under independence of the row and column variables, the odds ratio has a value of 1. Therefore, by rearranging Eqn. 1, we can obtain an estimate of $m_{22}$:

$$\hat{m}_{22} = \frac{m_{12}m_{21}}{m_{11}} \quad \text{Eqn. 2}$$

The total number of animals can be estimated by:

$$\hat{M} = \frac{m_{12}m_{21}}{m_{11}} + m_{11} + m_{12} + m_{21} = \frac{m_{1+}m_{+1}}{m_{11}} \quad \text{Eqn. 3}$$

The estimator in Eqn. 3 is known as the Lincoln-Petersen estimator. It was also derived by (Chandra Sekar and Deming, 1949) and applied in the estimation of birth and death rates.

In software engineering different types of CR models have been suggested and evaluated in the context of controlling the testing process (Ardissone et al., 1998; Basin, 1972; Duran and Wiorkowski, 1981; Isoda, 1998; Mills, 1972; Ohba, 1982). They have also been suggested for controlling the inspection process (Eick et al., 1991; Eick et al., 1992). More recently, CR models have been applied in estimating the extent to which software engineering standards are used (El Emam and Garro, 1999).

The idea behind using capture-recapture models for software engineering inspections is to let several inspectors draw samples from the population of defects (by identifying defects during the preparation stage of the inspection). The different inspectors reflect the various occasions depicted in Table 1. Based on the overlap of defects amongst inspectors, one can estimate the number of defects remaining in a software document using, for example, the Lincoln-Peterson estimator. Using this estimate and the known number of defects found in the inspection, it is possible to estimate the number of remaining defects in the inspected document (i.e., estimating $m_{22}$ in the above table). Subsequently, armed with this information, the inspection team can make an objective decision as to whether the document should be reinspected to reduce its defect content before passing it on to the next phase of the development life cycle.

Researchers at Bell Labs first applied CR models for requirements and design inspections (Eick et al., 1991; Eick et al., 1992; Eick et al., 1993). However, in these studies the true number of defects was unknown and therefore an evaluation of their true efficacy was not possible. Later work consisted of a Monte Carlo simulation to evaluate the robustness of different CR estimators to violations of their assumptions (Vander Wiel and Votta, 1993).

Objective empirical evaluation of CR models started with the study of Wohlin et al. (Wohlin et al., 1995). However, this study was conducted with non-software engineering documents. Subsequent work used software engineering documents (Briand et al., 1997; Briand et al., 1998e; Miller, 1998; Runeson and Wohlin, 1998). All of the above work utilized models that were originally developed in wildlife research. Other researchers considered the incorporation of Bayesian methods to estimate defect content and for model selection (Basu and Ebrahimi, 1998), performed further evaluations of assumption violations when using the estimation models (Thelin and Runeson, 1999a), evaluated the applicability of CR models to perspective-based reading (Thelin and Runeson, 1999b), and proposed new estimators for the case when the independence assumption is violated (Embrahimi, 1997).

An alternative approach was proposed in (Wohlin and Runeson, 1998), the Detection Profile Method (DPM). The DPM is an intuitively appealing approach that can be easily explained graphically to nonspecialists. A later study suggested a method for selecting between a CR model and the DPM (Briand et al., 1998a), and this was subsequently further evaluated in (Petersson and Wohlin, 1999).

In addition to the experiences reported by the researchers at Bell Labs, the use of the DPM at an insurance company in Germany was reported in (Briand et al., 1998d), and the application of CR models in telecommunications projects (Ardissone et al., 1998).

Thus far there has been no evaluation of *subjective* approaches for directly controlling software inspections. The only exception is one mention that the document quality (expressed as the number of defects not yet found) as estimated from a CR model matches the intuition of inspectors (Eick et al., 1991; Eick et al., 1993).

## 2.3 Definition of Subjective Estimates of Effectiveness

A subjective estimate of effectiveness, as used in this paper, is defined as an individual inspector's perception of the percentage of defects in a document that s/he has found. This estimate is produced after reading the document and logging the defects that s/he has found. For example, let's say that an inspector found 15 defects in a code document, and s/he estimates that 75% of the defects have been found (i.e., the 15 defects represent only 75% of the total defects in the document). It follows that the inspector estimates that there were 20 defects in total in the document, and that 5 defects remain in the document. As we shall see later in the paper, an individual inspector's estimate can be generalized to an inspection team of arbitrary size.

Of course, the precision of this subjective estimate is expected to be affected by many different variables. One of them may be, for example, the method that the inspector uses to read and understand the inspected document (Laitenberger et al., 1999). It is expected that a more systematic method results in a more precise estimate.

## 2.4 The Use of Subjective Estimates in Software Engineering

Since, to our knowledge, there is no extensive literature on the behavioral and application aspects of subjective estimates of inspection effectiveness, we can at least inform our endeavors by considering the work done in the area of subjective *cost estimation*.[1]

Despite the predominant use of subjective cost estimation practices in industry (Heemstra, 1992; Hihn and Habib-Agahi, 1991; Lederer and Prasad, 1992), there exists a strong bias *against* their use in software engineering. This is exemplified by statements such as "researchers can make a contribution by finding practices that can discourage the use of the informal basis [for cost estimation …], the employment of an informal basis should probably be discouraged […], managers should make diligent efforts to eschew them"[2] (Lederer and Prasad, 1998). While such perceptions may be supported by some empirical

[1] This is also sometimes refered to as informal or intuitive cost estimation.

[2] In this particular article, an informal basis is defined as intuition, comparison to similar, past projects based on personal memory, and guessing.

studies (e.g., see (Lederer and Prasad, 1992; Lederer and Prasad, 1998)), there also does exist a body of empirical evidence showing that subjective cost estimates can outperform more objective techniques (such as parametric models) (Vicinanza et al., 1991), or that they perform at least as well as the objective techniques (Kusters et al., 1990). Furthermore, cost estimation methods based on substantial subjective information have been shown to perform impressively well (Briand et al., 1998b), and contemporary cost estimation models are currently explicitly taking into account subjective expert opinion through the use of Bayesian statistics to improve their predictive performance (Devnani-Chulani, 1997).

Perhaps the strongest statement that we have found in support of subjective estimates of cost, or at least of their utility, was made by Hughes (Hughes, 1996). There, he chides the negative perception that subjective cost estimates have in the research community, and attempts to partially balance this by providing a clarification of how subjective cost estimation is performed in practice.

Thus, despite the negative perceptions of subjective estimates in some quarters of software engineering, this perception is certainly neither universal nor strongly justifiable.

Subjective cost estimation is arguably a more difficult problem than subjective estimation of inspection effectiveness. The reason being that for the former the software system to be developed is barely conceptualized and the estimate is for a task that will be performed in the future, while for the latter the document is completed and the estimate is for a task that has already been performed. However, if subjective estimates have been shown to work well in a more difficult context, such as cost estimation, then they have the potential of working well in an inspection effectiveness context. This assertion is further reenforced by the encouraging results of Selby (Selby, 1985) on the performance of subjective estimates of inspection effectiveness with code documents. The objective of the current study is therefore to test this assertion, and evaluate the utility of subjective estimates of effectiveness for making the reinspection decision.

# 3 Evaluating Subjective Estimates of Inspection Effectiveness

## 3.1 Evaluation of DCETs

The goal of any DCET is to provide an estimate of the number of defects in a document and, thus, a characterization of the document quality *before* the inspection was performed. We will denote this estimate as $\hat{D}_A$. Therefore, if an inspection finds $D_A$ defects, then the *estimated* remaining number of defects in the document after the inspection is $\hat{D}_R = \hat{D}_A - D_F$.

A commonly used criterion for evaluating a DCET in general is to compute the relative error, defined as:

$$RE = \frac{\hat{D}_A - D_A}{D_A} \qquad \text{Eqn. 4}$$

where $D_A$ is the true number of defects in the document. In Appendix A (Section 9) we show that the relative error, as defined above, only makes sense as an evaluative measure if the following two criteria are used to make the reinspection decision.

To control the quality of a document, the document should be reinspected if the following inequality is *not* satisfied:

$$\hat{D}_A < (Q_D \times LOC) + D_F \qquad \text{Eqn. 5}$$

where $LOC$ is the size of the document that was inspected (we use $LOC$ as a measure of size since our study is on code documents; other size measures could be used as well), and $Q_D$ is the minimal defect density that a document should have before it passes the inspection. In addition, to control the quality of the process, the document should be reinspected if the following inequality is *not* satisfied:

$$\hat{D}_A < \frac{D_F}{Q_P} \qquad \text{Eqn. 6}$$

where $Q_P$ is the minimal permissible effectiveness of the inspection.

## 3.2 Evaluating Subjective Estimates of Effectiveness

Below we describe how subjective estimates of effectiveness can be evaluated, and also demonstrate how the individual estimates of effectiveness can be applied to teams of inspectors.

An inspector provides a subjective estimate of his/her effectiveness, which we shall denote as $\hat{E}$. Therefore, if the inspector estimates that 75% of the defects in a document were found, the value of $\hat{E}$ would be 0.75.

Let us define $D_{F1}$ as the number of defects found by the first inspector, and it is this inspector who provides the subjective estimate of his/her individual effectiveness. Let $D_{Fk}$ be the sum of all other unique defects found by the remaining $k$ members of the inspection team that were not also found by the first inspector (i.e., remaining unique defects). The total unique defects found by the inspection team are given by:

$$D_F = D_{F1} + D_{Fk} \qquad \text{Eqn. 7}$$

Now we let:

$$\hat{D}_A = \frac{D_{F1}}{\hat{E}} \qquad \text{Eqn. 8}$$

which is the estimate of defect content using the subjective estimate of effectiveness from the first inspector.

We can define $Q$ as:

$$Q_D = \frac{d_D - D_{F1} - D_{Fk}}{LOC} \qquad \text{Eqn. 9}$$

where $d_D$ is a threshold defect content of the document. The logic of the above formulation is that a threshold defect density, $Q_D$, depends on the document's size, the number of defects found, and a threshold quality of the document. This means that, given the document and the inspection that was performed, $d_D$ is the maximum defect content of the document allowable to achieve the desired quality level. By substituting Eqn. 9 and Eqn. 7 into Eqn. 5, we get:

$$\hat{D}_A < d_D \qquad \text{Eqn. 10}$$

where:

$$d_D = (Q_D \times LOC) + D_{F1} + D_{Fk} \qquad \text{Eqn. 11}$$

Therefore, the decision criterion in Eqn. 10 will be applicable whether an inspection is conducted by one person or a team of inspectors. The only modification that is necessary, depending on the team size, is in the threshold.

We can also define $Q_P$ as:

$$Q_P = \frac{D_{F1} + D_{Fk}}{d_P} \quad \text{Eqn. 12}$$

where $d_P$ is a threshold defect content of the document. The logic of the above formulation is that a threshold effectiveness, $Q_P$, depends on the number of defects found, and a threshold quality of the document. This means that, given the document and the inspection that was performed, $d_P$ is the maximum defect content of the document allowable to achieve the desired effectiveness level. By substituting Eqn. 12 and Eqn. 7 into Eqn. 6, we get:

$$\hat{D}_A < d_P \quad \text{Eqn. 13}$$

where:

$$d_P = \frac{D_{F1} + D_{Fk}}{Q_P} \quad \text{Eqn. 14}$$

Therefore, the decision criterion in Eqn. 13 will be applicable whether an inspection is conducted by one person or a team of inspectors. The only modification that is necessary, depending on the team size, is in the threshold.

The above formulations have demonstrated that a defect content estimate based on an individual inspector's subjective estimate of his/her effectiveness can be used for making the reinspection decision irrespective of the inspection team size. We have also defined two decision criteria for controlling document quality and inspection effectiveness that directly use the estimated defect content.

## 3.3 Evaluating Decision Accuracy

Evaluating a DCET using the relative error criterion can provide a good indication of its bias. However, the sole reliance on relative error in studies that evaluate DCETs is not congruent with the manner in which DCETs are used. The reason is that the decision that needs to be made is binary: pass or reinspect. Therefore, a necessary complementary evaluation criterion would be the decision accuracy. The decision that needs to be made for controlling document quality is:

$$\hat{\lambda}_D = \begin{cases} 1 & , \quad \hat{D}_A < d_D \\ 0 & , \quad \hat{D}_A \geq d_D \end{cases}$$ Eqn. 15

where $\hat{\lambda}_D$ is the decision based on the DCET estimate, and is one (pass) if estimated defect density after the inspection is below a certain threshold, and zero (reinspect) if the defect density is equal to or greater than the threshold. In the above case, if $\hat{D}_A$ is only slightly lower than the threshold or much lower than the threshold, it does not matter because the same decision will be made (pass).

Similarly, one can define the decision for controlling inspection process quality:

$$\hat{\lambda}_P = \begin{cases} 1 & , \quad \hat{D}_A < d_P \\ 0 & , \quad \hat{D}_A \geq d_P \end{cases}$$ Eqn. 16

where $\hat{\lambda}_P$ is the decision based on the DCET estimate, and is one (pass) if the estimated effectiveness is higher than a certain threshold, and zero (reinspect) if it is lower than or equal to the threshold.

In evaluating decision accuracy, one can compare the decision based on the estimates, $\hat{\lambda}$ and $\hat{\lambda}_P$, with the decision that would be made if the DCET was perfectly accurate, which we will denote as $\lambda_D$ and $\lambda_P$ respectively:

$$\lambda_D = \begin{cases} 1 & , \quad D_A < d_D \\ 0 & , \quad D_A \geq d_D \end{cases}$$ Eqn. 17

and:

$$\lambda_P = \begin{cases} 1 & , \quad D_A < d_P \\ 0 & , \quad D_A \geq d_P \end{cases}$$ Eqn. 18

The results of an evaluation study on $N$ inspections can be placed in a confusion matrix as shown in Table 2.

| | | $\hat{\lambda}$ or $\hat{\lambda}_P$: **0** | **1** | |
|---|---|---|---|---|
| $\lambda$ or $\lambda_P$ | **0** | $n_{11}$ | $n_{12}$ | $N_{1+}$ |
| | **1** | $n_{21}$ | $n_{22}$ | $N_{2+}$ |
| | | $N_{+1}$ | $N_{+2}$ | $N$ |

Table 2: Notation for a confusion matrix for evaluating decision accuracy.

We define the decision accuracy in terms of the proportion of correct decisions that would be made using the estimates:

$$Accuracy = A_1 = \frac{n_{11} + n_{22}}{N} \qquad \text{Eqn. 19}$$

However, this definition of accuracy does not take into account the improvement due to the use of subjective estimates. For example, at Bosch Telecom, as well as in many other environments, reinspections are rarely performed. Hence, the "no reinspection" decision can be considered the default one. If this default decision attains the desired product and process quality levels say 90% of the time and the use of subjective estimates also results in achieving the desired quality levels 90% of the time, then using the subjective estimates does not add any value. Thus, even though 90% accuracy for the subjective estimates may seem impressive, under the above condition they are simply an overhead. We therefore propose the following definition of relative accuracy that accounts for improvements over the default decision:

$$Relative\ Accuracy = A_2 = \frac{A_1 - A_d}{A_d} \qquad \text{Eqn. 20}$$

where $A_d$ is the accuracy obtained when using the default decision, which in our case is always pass. More precisely, $A_d$ can be defined with reference to the following confusion matrix:

| | | **Default Decision** | | |
|---|---|---|---|---|
| | | **0** | **1** | |
| $\lambda$ or $\lambda$ | **0** | 0 | $n_{12}$ | $N_{1+}$ |
| | **1** | 0 | $n_{22}$ | $N_{2+}$ |
| | | 0 | N | N |

Table 3: Notation for a confusion matrix for evaluating the default decision.

and:

$$A_d = \frac{n_{22}}{N} \qquad \text{Eqn. 21}$$

The definition in Eqn. 20 indicates how much better a subjective estimate is beyond the default decision. It is positive if the subjective estimate is better, zero if they are the same, and negative if the subjective estimate is worse than the default decision.

# 4 Research Method

## 4.1 Description of the Environment

The data we analyzed in this paper were collected during a study that we performed with a total of 30 professional software developers at Bosch Telecom GmbH between March and July 1998.

Bosch Telecom GmbH is a major player in the telecommunications market and develops high quality telecommunication systems (e.g., modern transmission systems based on SDH technology, access networks, switching systems) containing embedded software. One major task of the embedded software is the management of these systems. There are four typical characteristics for this kind of software. First, it is event triggered. Second, there are real time requirements. Third, the software must be highly reliable which basically means that the software system must be available 24 h. Finally, the developed software system must be tailorable to different hardware configurations. Because of these characteristics and the increasing competition in the telecommunications market, high product quality represents one of the most crucial demands for software development projects at Bosch Telecom GmbH. The requirement of high software quality and reliability highlights the need to make correct decisions about reinspecting documents.

## 4.2 Description of the Study

The study consisted of three sessions in each of which 10 developers participated. Each session consisted of four parts. The first one comprised an intensive exercise introducing the principles of software inspection. This explanation covered the theory behind inspection roles, inspection processes, and different techniques that help individuals detect defects in a code module (i.e., reading techniques). The second part consisted of a practical exercise in which the subjects individually scrutinized a C-code module for defects using a checklist-based reading technique (Fagan, 1976). The code modules were part of running software systems of Bosch Telecom GmbH and, defects were seeded in them beforehand. We changed the code modules after each training session. This was to avoid that an exchange of information between participants of different sessions would bias their performance. While inspecting the code module, the subjects were asked to log all detected defects on a defect report form. After this exercise we asked the subjects to fill out a debriefing questionnaire which required them to make the subjective estimate of the percentage of defects they found in the code module (i.e., $\hat{E} \times 100$).

The third part of a session consisted of a two-person inspection meeting in

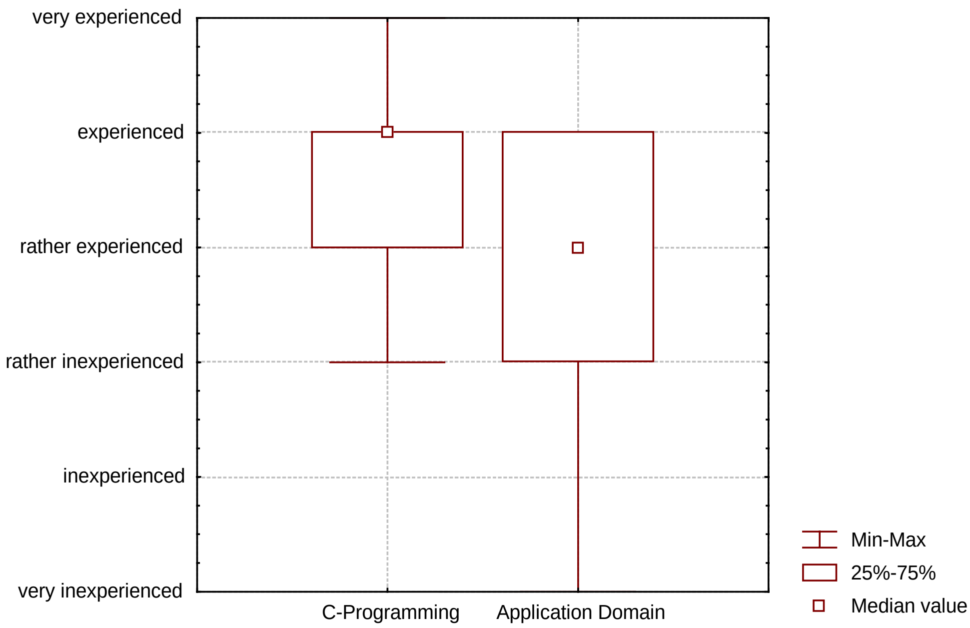


Figure 1 : Subjects' experience with the C-programming language and the application domain

We found that subjects perceived themselves experienced with respect to the programming language (median of 5 on the 6 item scale) and rather experienced regarding software development in the application domain (median of 4 on the six item scale). This corroborates our assumption that our pool of subjects consists of expert developers rather than novices.

## 4.4 Description of Materials – Code Modules

Table 4 shows the size of the code module in Lines of Code (without blank lines), their average cyclomatic complexity using McCabes complexity measure, and the number of injected defects.

| | **Size (LOC)** | **Average Cycl. Complexity** | **Number of injected defects** |
|---|---|---|---|
| Code Module 1 | 666 | 3.20 | 10 |
| Code Module 2 | 375 | 6.67 | 8 |
| Code Module 3 | 627 | 5.44 | 11 |

Table 4: Characteristics of the Code Modules.

As Table 4 reveals the code modules have different characteristics and are considered to be typical of code modules at Bosch Telecom GmbH. The code modules were randomly assigned to each of the training sessions to alleviate the possibility that there is a group-module relationship, We asked either the author or a very experienced software developer (if the author participated in the study) to inject defects into the code modules. These defects should be typical for the ones that are usually detected in testing and should not be detectable automatically by compilers or other tools, such as lint.

The subjects sometimes reported more defects on their defect report forms than were seeded in a code module. When a true defect was reported that was not on the list of seeded defects, we added this defect to the list of known defects and reanalyzed the defect report forms of all the remaining subjects. Whether a defect was a true defect was our decision (to some extent based on discussions with the author or the person who seeded defects). Table 4 shows the final number of defects in each module.

Since during our study we use the results of individual inspectors' actual effectiveness to calculate accuracy, and an inspector's subjective estimate is invariably influenced by the number of defects that he or she actually finds, the results may be distorted by false positives. False positives are potential defects a subject reported on his or her defect report form, which turn out not to be "real" defects. If a subject reports many false positives he or she may be too confident in the quality of the inspected product and the effectiveness of the inspection process. This may result in an overestimation of detected defects. In general we found that our subjects reported very few false positives. Therefore, this effect has only a minor influence on the results.

## 4.5 Description of Materials – Checklists

The CBR approach attempts to increase inspection effectiveness and decrease the cost per defect by focusing the attention of inspectors on a defined set of questions. In our study, we provided the inspectors with a generic checklist, that is, a checklist that was not particularly tailored to the application domain of Bosch Telecom GmbH. We limited the number of checklist items to 27 questions to fit on one page since this is recommended in the literature (Chernak, 1996). We structured the checklist according to the schema that is presented in (Chernak, 1996). The schema consists of two components: "Where to look" and "How to detect". The first component is a list of potential "problem spots" that may appear in the work product and the second component is a list of hints on how to identify a defect in the case of each problem spot. As problem spots we considered data usage (i.e., data declaration and data referencing), computation, comparison, control flow, interface, and memory. For each problem spot, we derived the checklist items from existing checklists (Nasa, 1993) and books about the C Programming language (Deitel and Deitel, 1994; Kernighan and Ritchie, 1990). Hence, the problem spots as well as the questions help reveal defects typical in the context of software development with the C programming language. The complete checklist can be found in Appendix B.

## 4.6 Data Analysis

At the outset, we evaluate the relative error (as defined in Eqn. 4) of the subjective defect content estimate. This gives us an overview of the bias of these types of estimates.

We then evaluate the relative decision accuracy for the two types of scenarios mentioned in Section 3: controlling document quality, and controlling inspection effectiveness. For each scenario we varied the thresholds, $Q_D$ and/or $Q_P$, and the number of defects that may be found by other members of an inspection team. This would allow us to investigate the sensitivity of using subjective estimates to threshold variations, and also to variations in the proportion of defects that the estimator finds.

For the number of defects that are found by other members of an inspection team, we defined:

$$D_{Fk} = (D_A - D_{F1}) \times \omega \qquad \text{Eqn. 22}$$

where $\omega$ is a proportion defined to be 0, 0.25, 0.5, and 0.75.[3] We present our results in the form of plots of the relative accuracy (see Eqn. 20) to evaluate how well subjective estimates perform.

[3] The zero values denote that the inspection team consists of only one person.

# 5 Results

## 5.1 The Relative Error of Subjective Estimates

The box and whisker plot in Figure 2 shows the relative error (as defined in Eqn. 4) of the subjective estimates for each of the three modules (the number of estimates for each plot is 10). As can be seen the median RE is quite close to zero, but may lead to overestimation (module 2) or underestimation (module 3). Also, it will be noted that for module 2 there is one extreme outlier. This outlier can be explained by the fact that this person usually develops software for a different platform within the application domain. This may be the reason that the person was unsure about the percentage of defects found in the inspected code module and, therefore, followed a conservative strategy to provide a very low effectiveness estimate. But since the person's real effectiveness was much higher than the estimated one, a large, positive RE could be observed.[4]

To determine whether there are differences in accuracy across the different modules we performed a median test based on the chi-square statistic. We divided the observations for each module into two classes based on whether they are above or below the overall median (which is zero). A 3x2 contingency table was then used to test for association between the modules and difference from the overall median. The chi-square value was not statistically significant at an alpha value of 0.1. Given that there are no discernable differences in accuracy across the modules, we pool them.

The box and whisker plot in Figure 3 shows the overall relative error. Here, it is seen that the median relative error is zero. This is quite encouraging as it indicates negligible bias in using subjective estimates of effectiveness to estimate defect content.

---

4 Exclusion of this outlier does not affect our conclusions. Therefore, all subsequent analyses include this data point.

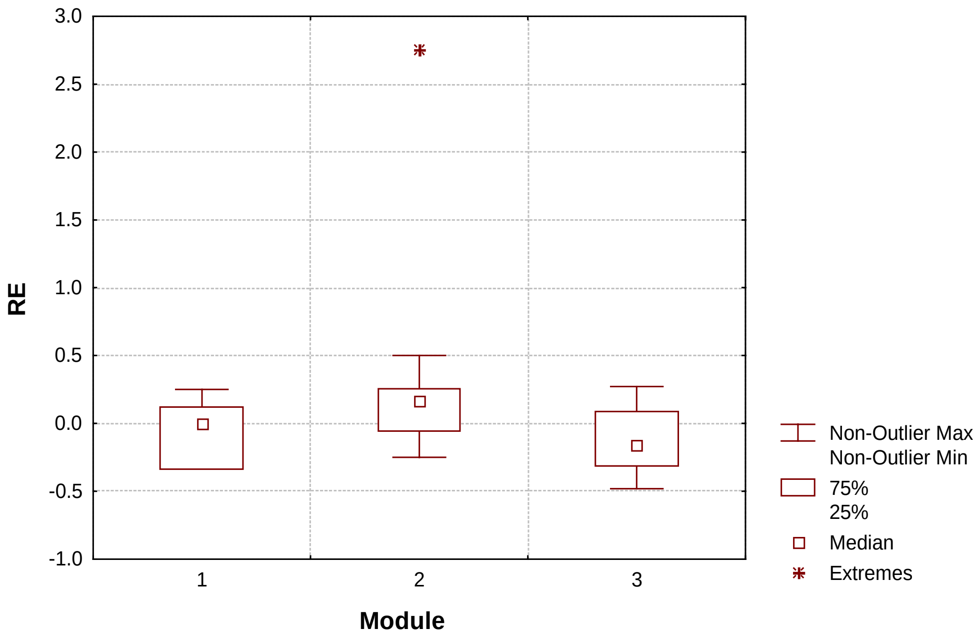


Figure 2: The relative error of subjective estimates for each of the three modules.

## 5.2 Accuracy of Subjective Estimates

To help interpret the results, we first present the default decision accuracy ($A$ ) as the defect density threshold changes in Figure 4. The different lines indicate variations in the number of defects found by the remainder of the inspection team. The x-axis is varied from 1 to 100 defects/KLOC.

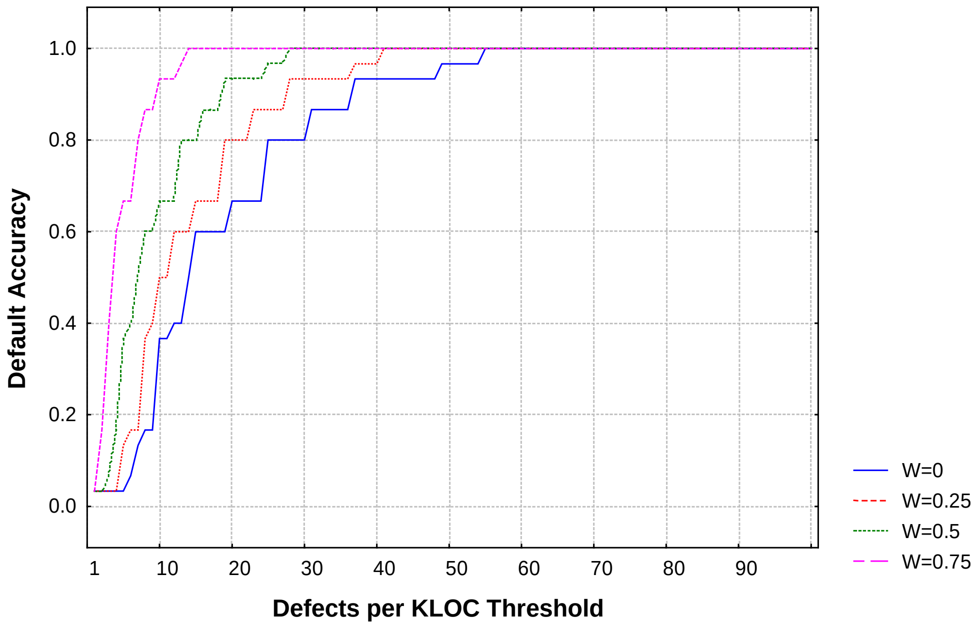


Figure 4: Changes in the accuracy of the default decision as the defect density threshold changes.

It can be seen that at the least stringent thresholds (higher values of defects per KLOC) the default decision is correct almost all the time (accuracy is one). This is because at these low stringency levels the default decision of passing the document is correct almost all of the time. At the more stringent thresholds the default decision of no reinspection will frequently be incorrect, since in such circumstances the document will more frequently require reinspection. Also, it can be seen that the fall in default decision accuracy occurs sooner for the least effective inspection team ($\omega = 0$) when compared to the most effective inspection team ($\omega = 0.75$). This is because the default decision will be correct for a larger range of thresholds for the more effective teams.

In Figure 5 we can see the *relative accuracy* ($A_2$) for different thresholds of defect density. There is essentially no difference between the subjective estimates and the default decision making criterion (accuracy is at or close to zero) for the less stringent thresholds. As noted above, this is because the default decision is frequently correct, so not much improvement can be gained over that. At the more stringent thresholds it is relatively easy to achieve an improvement over the default decision, and therefore the relative accuracy can be dramatically large.

It is noted that the relative accuracy value rarely if ever dips below zero. This means that using subjective estimates will, in general, not be worse than the default decision, and in cases where the threshold is stringent, considerably better. This indicates a good ability of the subjective estimates to *discriminate* between instances where a reinspection is required from those where it is not. One can then draw the conclusion that using subjective estimates to control document quality can be implemented irrespective of the threshold that is chosen.

The best accuracy results are obtained the lower the value of $\omega$ [5]. This indicates that if the best inspector (i.e., the one who finds the greatest number of defects) makes the estimate, the accuracy tends to be systematically better.

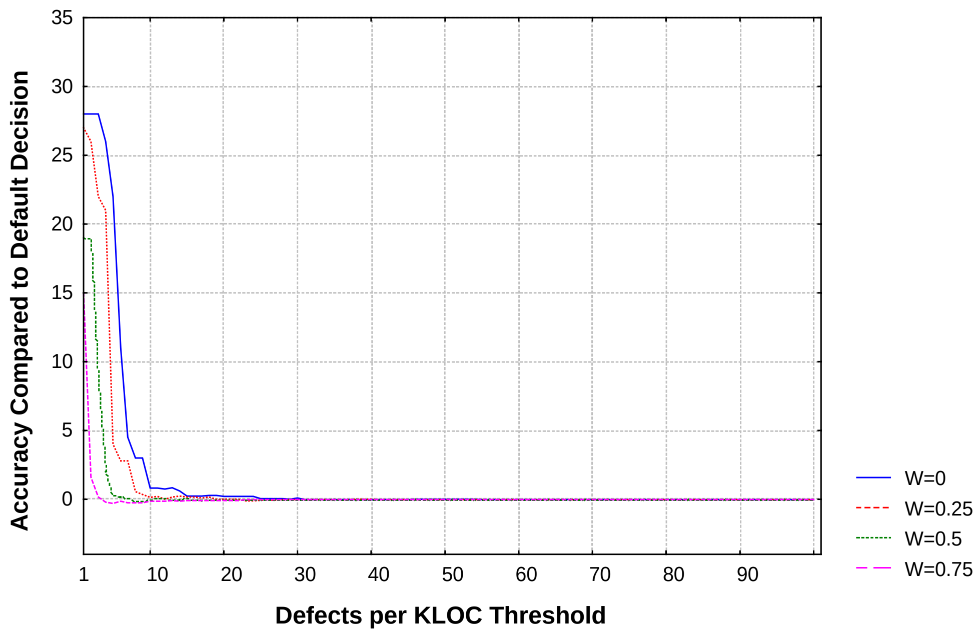


Figure 5: Variation of Accuracy with the defect density threshold for different numbers of defects found by the remainder of an inspection team.

In Figure 6 we can see the results for controlling the effectiveness of the inspection process.  The x-axis represents the threshold effectiveness, expressed as a percentage. At the less stringent thresholds (i.e., low effectiveness) there is

[5] The reason for this can be seen in Figure 4, whereby at larger $\omega$ the default decision is correct most frequently for longer, and therefore presents a bigger hurdle.

no difference between using the subjective estimates and the default decision criterion. However, at the more stringent thresholds, the subjective estimates tend to perform substantially better.

Similar to the conclusion that we drew above, we can state that using subjective estimates for controlling the effectiveness of inspections will almost always at least perform as well as the default decision, and in the cases with more stringent thresholds, will perform better.

The best accuracy results are obtained the lower the value of $\omega$ . This indicates that if the estimator is the best inspector (i.e., the one who finds the greatest number of defects), the accuracy tends to be systematically better.

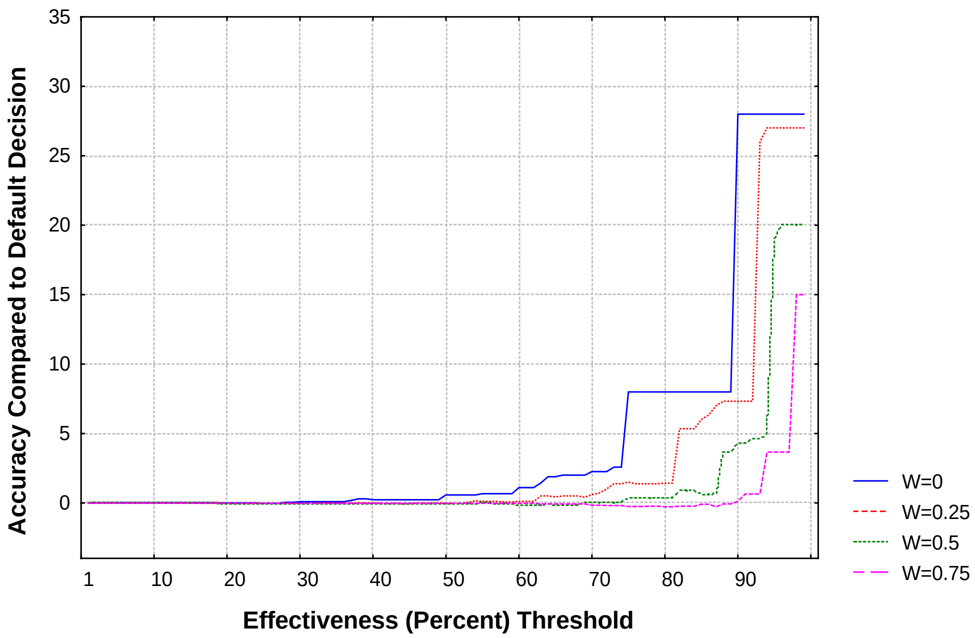


Figure 6: Variation of accuracy with the effectiveness threshold for different numbers of defects found by the remainder of an inspection team.

Based on the above results, it is a reasonable suggestion that practitioners use subjective estimates to control product and process quality, irrespective of the thresholds that are chosen, if more objective DCET's are not available. This has the advantage in that the costs of using such a technique are negligible, and above we demonstrate that it has advantages in terms of decision accuracy. Furthermore, the best results will be obtained if the estimator is the best inspector on the inspection team.

## 5.3 Discussion

In this section we discuss the results and the issues that can arise when applying subjective estimates in practice.

### 5.3.1 Inspection Coverage

Some companies may not be in a position to inspect all of their development products due to schedule or effort constraints. The question that arises in this situation is whether to perform a reinspection on one document (if necessary) or inspect another document that has not been inspected so far. However, this is not a decision of the members of an inspection team. It is rather one that the project management must make . If the management committs itself to developing high quality software, it must spend the effort for a reinspection once the inspection team has decided on it. The subjective estimates of the inspection team members therefore should not be biased by the question of whether to perform reinspections at all.

### 5.3.2 Personal Bias and Political Pressure

An obvious danger with subjective estimates is that they are liable to personal bias and political pressure. For example, consider the situation where a team is behind schedule and is under very high pressure to pass a document to the next phase. This may tempt the exaggeration of subjective effectiveness to pass the document. However, actions that defeat the purpose of a quality assurance technique, such as inspection, can be performed whether the reinspection decision is subjective or based on objective information. This argument is further supported by Hughes (Hughes, 1996) who states that, in the realm of cost estimation, “all estimating methods may be liable to personal bias and political pressure”. For instance, when using a parametric estimation model, the estimator may be pressured to modify the input subjective cost drivers if the estimated cost is unacceptable (say, it is too high and the project manager is very eager to have his/her project approved by senior management).

Nevertheless, below we discuss managerial actions that may discourage such biases.

### 5.3.3 Estimate Accountability

A recent study on cost estimation practices found evidence supporting the claim that when estimators are held accountable for their estimates, the accuracy of the estimates increases (Lederer and Prasad, 1998). It would therefore seem reasonable that, should subjective estimates of effectiveness be introduced into an organization, the estimators ought to be held accountable for document quality and inspection process quality. Such a practice would likely dilute the negative impact of personal bias and political pressure.

### 5.3.4 Collusion Amongst Inspectors

Collusion amongst inspectors occurs if inspectors discuss their defects or work jointly during the preparation phase of the inspection. Dealing with collusion has been a concern in the DCET literature (Basu and Ebrahimi, 1998; Ebrahimi, 1997; Eick et al., 1993).

If there is collusion where the subjective estimator is involved then we can consider two cases:

- If the other inspector(s) find many defects that s/he did not find, then this may push the estimator to reduce his/her estimate of personal effectiveness.
- If the other inspector(s) find few defects that s/he did not find, then this may prompt the estimator to raise his/her estimate of personal effectiveness.

Since we recommend that the "best" inspector take the role of the estimator, the second case is more likely. Therefore, should collusion exist, we would expect an overestimation of subjective effectiveness. The evaluation of the impact on relative accuracy ought to be an item for future research. It should be noted that during our study the design precluded collusion.

### 5.3.5 Subjective Estimates vs. Objective DCETs

Do we suggest that one do away with objective DCETs and simply use expert judgement in making the reinspection decision? Certainly the case seems attractive. Subjective estimates are low cost and do not require additional data collection to operationalise.

Such a suggestion, however, would not be corroborated by the results of this study since we did not compare the decision accuracy of subjective estimates with those from objective DCETs. Given the encouraging performance of subjective estimates as indicated in the current study, it would certainly be of value to perform this comparison in future research.

### 5.3.6 Subjective Estimates and Decision Transparency

A recent study (Lederer and Prasad, 1998) found a positive relationship between user commitment and participation in IS development and the use of more objective methods for cost estimation. This may be interpreted that the use of objective methods provides visibility and transparency in the estimation process, and may also provide justification for decisions that are made. It is expected that such benefits would be dampened with subjective estimates.

This finding and its interpretation would suggest some inherent disadvantages when using subjective estimates of effectiveness as the sole basis for making the reinspection decision, even if accountability is enforced. It is therefore recommended that subjective estimates of effectiveness be applied primarily in three contexts:

- *As a starting point for controlling software inspections.* The ease with which such a practice can be institutionalized and the evidence we present as to its efficacy supports this recommendation.
- *Evaluating objective DCETs.* In many practical situations the real number of defects or the real decision is unknown. This makes it difficult to evaluate an objective DCET in a particular environment. However, given the good decision accuracy performance of subjective estimates, they can be used to evaluate objective DCETs.
- *Bayesian DCETs.* Our results would certainly recommend the use of subjective estimates as informative priors to Bayesian capture-recapture models. This is likely to result in improved decisions when compared to the more common (non-Bayesian) models.

### 5.3.7 Relevance of Thresholds

Another point that should be emphasized as a consequence of this study is that the performance of any method for making the reinspection decision, whether subjective or objective, is inherently linked to the threshold that is used for controlling document and inspection process quality. Therefore, future studies should either evaluate the methods across all reasonable thresholds, or select a number of thresholds as the basis for the evaluation.

## 5.4 Threats to validity

It is the nature of any empirical study that assumptions are made that later on may restrict the validity of the results. Here, we list these assumptions that impose threats to internal and external validity.

### 5.4.1 Threats to Internal Validity

One important potential threat to the internal validity of our study concerns defect severity. Similar to previous studies that evaluate DCETs, we do not consider the characteristics of the defects that are found. With characteristics we mean, for example, the severity or the criticality of defects. An inspection that reveals few extremely critical defects is different from an inspection that detects many uncritical defects. When only looking at some thresholds one may decide not to reinspect in the first case while making a reinspection in the second one. However, from a purely subjective point of view, it would be better to reinspect the first one to be sure that there are no other major defects and that the defects are corrected without introducing other or even more major defects (in some existing inspection implementations, a follow-up phase is performed as part of the inspection for checking this (Gilb and Graham, 1993)). However, it is more difficult to model the characteristics of the defects in a decision criterion. One approach might be to just consider critical or severe defects for decision making.

A second potential threat to the internal validity of our study is related to the fact that seeded defects were used. Despite us taking special care to alleviate this (for example, by reinjecting previously detected defects rather than making up new defects), there is always the danger that seeded defects are not the same as actual defects. Specifically, they may be easier to detect than actual defects. If an inspector realizes that the defects found are easy, then s/he may be suspicious that the defects are too easy and that the difficult defects were not found yet, hence deflating the effectiveness estimate. However, we did not observe this effect in our data, therefore this potential threat may not be severe.

### 5.4.2 Threats to External Validity

Our study was performed with subjects and code documents from a single organization. While this enjoys greater external validity than doing studies with students in a "laboratory" setting, it is uncertain the extent to which the results can be generalized to other organizations. However, our result are in line with previous findings (Selby, 1985).

# 6 Conclusion

Knowing the number of defects in an inspected document can provide the basis for deciding whether to reinspect a document. Making the correct reinspection decision provides a means for ensuring that inspected documents have a defect density below a prespecified threshold, and that the inspection process has attained a minimal level of effectiveness.

Thus far, software engineers have proposed capture-recapture models and the Detection Profile Method as objective means for estimating the number of defects in a document. Both of these approaches utilize data collected during an inspection to make an estimate.

In this paper we evaluated subjective estimates of effectiveness by the inspectors as a basis for making that decision. We showed how subjective estimates can be used for such a purpose and detailed an empirical evaluation of subjective estimates. The empirical study was performed with 30 professional software engineers. Our results indicate that subjective estimates of defect content have a median relative error of zero, and do indeed consistently outperform the default decision of not reinspecting a document.

Given that subjective estimates are easy to perform, they can provide a cost effective way for making the reinspection decision. This is especially true since we have shown that the subjective estimate of only a single inspector is necessary to make the decision for a team-based inspection.

These results are encouraging for organizations contemplating the use of reinspections for controlling their product and process quality. It can be recommended that subjective estimates be used as a starting point for controlling inspections and for evaluating more objective decision models. From a research perspective, our results suggest that future studies ought to compare the performance of subjective estimates with more objective techniques such as capture-recapture models and the Detection Profile Method, evaluate the use of Bayesian capture-recapture models, and ought to consider the effects of collusion among inspectors. In addition, in our study the estimates did not take into account the confidence of the inspectors in their estimates. Future work ought to consider the inspectors expressing a distribution of their effectiveness rather than a point estimate, much in the same manner as recent studies of subjective estimates of cost (Connolly and Dean, 1997; Hihn and Habib-Agahi, 1991; Host and Wohlin, 1997; Host and Wohlin, 1998).

# 7 Acknowledgements

We thank both the managers and developers at Bosch Telecom GmbH for their dedicated efforts in making this study possible as well as for their participation.

# 8 References


Adams, T., 1999. A Formula for the Re-Inspection Decision. Software Engineering Notes, 24(3), p. 80.

Ardissone, M., Spolverini, M., Valentini, M., 1998. Statistical Decision Support Method for In-Process Inspections. Proceedings of the 4th International Conference on Achieving Quality in Software, p. 135-143.

Basin, S., 1972. Estimation of Software Error Rates via Capture-Recapture Sampling. Technical Report, Science Applications Inc..

Basu, S., Ebrahimi, N., 1998. Estimating the Number of Undetected Errors: Bayesian Model Selection. Proceedings of the 9th International Symposium on Software Reliability Engineering, p. 22-31.

Briand, L., El Emam, K., Freimut, B., Laitenberger, O., 1997. Quantitative Evaluation of Capture Recapture Models to Control Software Inspections. Proceedings of the 8th International Symposium on Software Reliability Engineering, p. 234-244.

Briand, L., El Emam, K., Freimut, B., 1998a. A Comparison and Integration of Capture-Recapture Models and the Detection Profile Method. Proceedings of the 9th International Symposium on Software Reliability Engineering, p. 32-41.

Briand, L., El Emam, K., Bomarius, F., 1998b. COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. Proceedings of the 20th International Conference on Software Engineering, p. 390-399.

Briand, L., El Emam, K., Laitenberger, O., Fussbroich, T., 1998c. Using Simulation to Build Inspection Efficiency Benchmarks for Development Projects. Proceedings of the 20th International Conference on Software Engineering, p. 340-349.

Briand, L., Freimut, B., Laitenberger, O., Ruhe, G., Klein, B., 1998d. Quality Assurance Technologies for the EURO Conversion – Industrial Experience at Allianz Life Assurance. Proceedings of the Quality Week Europe.

Briand, L., El Emam, K., Freimut, B., Laitenberger, O., 1998e. A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content. Submitted for publication.

Chandra Sekar, C., Deming, W., 1949. On a Method of Estimating Birth and Death Rates and the Extent of Registration. Journal of the American Statistical Association, 44(245-248), p. 101-115.

Chernak, Y., 1996. A Statistical Approach to the Inspection Checklist Formal Synthesis and Improvement. IEEE Transactions on Software Engineering, 22(12), p. 866–874.

Connolly, T., Dean, D., 1997. Decomposed Versus Holistic Estimates of Effort Required for Software Writing Tasks. Management Science, 43(7), p. 1029-1045.

Deitel, H., Deitel, P., 1994. C How to program, 2nd ed.. Prentice Hall.

Devnani-Chulani, S., 1997. Incorporating Bayesian Analysis to Improve the Accuracy of COCOMO II and Its Quality Model Extension. Technical Report, University of Southern California, Computer Science Department.

Duran, J., Wiorkowski, J., 1981. Capture-Recapture Sampling for Estimating Software Error Content. IEEE Transactions on Software Engineering, 7(1), p. 147-148.

Ebrahimi, N., 1997. On the Statistical Analysis of the Number of Errors Remaining in a Software Design After Inspection. IEEE Transactions on Software Engineering, 23(8), p. 529-532.

Eick, S., Loader, C., Long, M., Votta, L., Vander Wiel, S., 1991. Investigating the Application of Capture-Recapture Techniques to Requirements and Design Reviews. Proceedings of the SEL Software Engineering Workshop, p. 97-102.

Eick, S., Loader, C., Long, M., Votta, L., Vander Wiel, S., 1992. Estimating Software Fault Content Before Coding. Proceedings of the 14th International Conference on Software Engineering, p. 59-65.

Eick, S., Loader, C., Vander Weil. S., Votta, L., 1993. How Many Errors Remain in a Software Design After Inspection ?. Proceedings of the 25th Symposium on the Interface, Interface Foundation of North America, p. 195-202.

El Emam, K., Garro, I., 1999. Estimating the Extent of Standards Use: The Case of ISO/IEC 15504. Submitted for Publication.

Fagan, M.E., 1976. Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal, vol. 15, no. 3, p. 182-211.

Gilb, T., Graham, D., 1993. Software Inspection. Addison-Wesley Publishing Company.

Heemstra, F., 1992. Software Cost Estimation. Information and Software Technology, 34(10), p. 627-639.

Hihn, J., Habib-Agahi, H., 1991. Cost Estimation of Software Intensive Projects: A Survey of Current Practices. Proceedings of the 13th International Conference on Software Engineering, p. 276-287.

Host, M., Wohlin, C., 1997. A Subjective Effort Estimation Experiment. Proceedings of the Conference on Empirical Assessment in Software Engineering.

Host, M., Wohlin, C., 1998. An Experimental Study of Individual Subjective Effort Estimations and Combinations of Estimates. Proceedings of the 20th International Conference on Software Engineering, p. 332-339.

Hughes, R., 1996. Expert Judgement As An Estimating Method. Information and Software Technology, 38, p. 67-75.

Isoda, S., 1998. A Criticism on the Capture and Recapture Method for Software Reliability Assurance. Journal of Systems and Software, 43, p. 3-10.

Kernighan, B.W., Ritchie, D.M., 1990. Programming in C. Hanser Verlag.

Kusters, R., van Genuchten, M., Heemstra, F., 1990. Are Software Cost Estimation Models Accurate?. Information and Software Technology, 32(3):187-190, 1990.

Laitenberger, O., DeBaud, J.M., 1998. An Encompassing Life-Cycle Centric Survey of Software Inspection. International Software Engineering Research Network (ISERN) Technical Report ISERN-98-14, Fraunhofer Institute for Experimental Software Engineering. To appear in the Journal of Systems and Software, 2000.

Laitenberger, O., El Emam, K., Harbich, T., 1999. An Internally Replicated Quasi-Experimental Comparison of Checklist and Perspective-based Reading of Code Documents. International Software Engineering Research Network, Technical Report ISERN-99-01.

Lederer, A., Prasad, J., 1992. Nine Management Guidelines for Better Cost Estimating. Communications of the ACM, 35(2), p. 51-59.

Lederer, A., Prasad, J., 1998. A Causal Model of Software Cost Estimating Error. IEEE Transactions on Software Engineering, 24(2), p. 137-148.

Miller, J., 1998. Estimating the Number of Remaining Defects After Inspection. International Software Engineering Research Network Technical Report ISERN-98-24.

Mills, H., On the Statistical Validation of Computer Programs. Technical Report FSC-72-6015, IBM Federal Systems Division.

National Aeronautics and Space Administration, 1993. Software Formal Inspection Guidebook. Technical Report NASA-GB-A302, National Aeronautics and Space Administration. http://satc.gsfc.nasa.gov/fi/fipage.html.

Ohba, M., 1982. Software Quality = Test Accuracy X Test Coverage. Proceedings of the 6th International Conference on Software Engineering, p. 287-293.

Otis, D., Burnham, K., White, G., Anderson, D. 1978. Statistical Inference from Capture Data on Closed Animal Populations. Wildlife Monographs, 62, p. 1-135.

Petersson, H., Wohlin, C., 1999. Evaluation of Using Capture-Recapture Methods on Software Review Data. Proceedings of the Conference on Empirical Assessment in Software Engineering.

Runeson, P., Wohlin, C., 1998, An Experimental Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections. Empirical Software Engineering, 3, p. 381-406.

Seber, G., 1982, The Estimation of Animal Abundance and Related Parameters. Charles Griffin & Company Ltd., 2nd. edition.

Selby, R.W., 1985. Evaluations of Software Technologies: Testing, Cleanroom, and Metrics. PhD-Thesis, Department of Computer Science, University of Maryland.

Stephen, C., 1996. Capture-Recapture Methods in Epidemiological Studies. Infection Control and Hospital Epidemiology, 17(4), p. 262-266.

Strauss, S., Ebenau, R., 1994. Software Inspection Process. McGraw Hill.

Thelin, T., Runeson, P., 1999a. Robust Estimations of Fault Content with Capture-Recapture and Detection Profile Estimators. Proceedings of the Conference on Empirical Assessment in Software Engineering.

Thelin, T., Runeson, P., 1999b. Capture-Recapture Estimations for Perspective-Based Reading – A Simulated Experiment. Submitted for Publication.

Vicinanza, S., Mukhopadhyay, T., Prietula, M., 1991. Software-Effort Estimation: An Exploratory Study of Expert Performance. Information Systems Research, 2(4), p.243-262.

Vander Wiel, S., Votta, L., 1993. Assessing Software Designs Using Capture-Recapture Methods, IEEE Transactions on Software Engineering, 19(11), p.1045-1054.

White, G., Anderson, D., Burnham, K., Otis, D., 1982. Capture-Recapture and Removal Methods for Sampling Closed Populations. Technical Report LA-8787-NERP, Los Alamos National Laboratory.

Wickens, T., 1989. Multiway Contingency Tables Analysis for the Social Sciences. Lawrence Erlbaum Associates.

Wohlin, C., Runeson, P., Brantestam, J., 1995. An Experimental Evaluation of Capture-Recapture in Software Inspections. Software Testing, Verification and Reliability, 5, p.213-232.

Wohlin. C., Runeson, P., 1998, Defect Content Estimations from Review Data. Proceedings of the 20$^{th}$ International Conference on Software Engineering, p. 400-409.

# 9 Appendix A: Current Approaches for Evaluating DCETs

The literature on DCETs and their evaluation in software engineering has been rather ambiguous about how the estimates from a DCET can be used in practice. This is disconcerting given that criteria for the evaluation of a DCET's estimation accuracy should be strongly congruent with the decisions that are made using the estimates. In this section we review existing approaches for evaluating a defect content estimation technique's accuracy (whether objective or subjective). It is found that the criteria that have been used thus far are not appropriate, and alternative criteria are proposed.

First we need to define some notation:

| Term | Definition |
|---|---|
| $D_A$ | The actual number of defects in a document. |
| $\hat{D}_A$ | The estimated number of defects in a document. |
| $D_A$ | The unique number of defects found during an inspection. |
| $D_R = D_A - D_F$ | The actual number of remaining defects after the inspection. By definition this value is always positive. |
| $\hat{D}_R = \hat{D}_A - D_F$ | The estimated number of remaining defects after the inspection. For the purposes of our exposition, we assume that $\hat{D}_A > D_F$ |
| $LOC$ | The size of the inspected document in Lines of Code. This can be any size measure and does not have to be Lines of Code, but for the sake of our exposition we use LOC. |

It has been stated that "One approach to optimize the effectiveness of inspections is to reinspect a software document that is presumed to still have high defect content. The reinspection decision criterion could be based on the number of remaining defects after an inspection, which can be estimated with defect content models." (Briand et al., 1997), "The [capture-recapture] method is based on the review information from the individual reviewers and through statistical inference, conclusions are drawn about the remaining number of defects after the review. This would allow us to take informed and objective decisions regarding whether to continue, do rework, or review some more." (Wohlin and Runeson, 1998). An obvious interpretation of such statements would imply that a DCET should be used to only pass a document if the estimated number of remaining defects is less than some constant threshold:

$$\hat{D}_R < D_T \qquad \text{Eqn. 23}$$

where $D_T$ is some threshold defined by the organization as symbolizing a "minimum acceptable quality" in terms of the number of remaining defects in an inspected document. As illustrated above, it has been argued that this allows one to make an objective decision as to whether the document is of sufficient quality, and if not then to reinspect it.

Below we show that the manner in which DCETs have been evaluated in the software engineering literature is inappropriate given the decision criterion in Eqn. 23 because of a discordance between the evaluation and decision criteria, and because current evaluation practices make unrealistic assumptions about inspected documents.

### 9.1.1 Discordance between Evaluation and Decision Criteria

In many previous studies that empirically evaluated DCETs, the relative error was used as an accuracy evaluation criterion, for example, (Briand et al., 1997; Briand et al., 1998a, Petersson and Wohlin, 1999; Runesion and Wohlin, 1998; Thelin and Runeson, 1999a; Thelin and Runeson, 1999b). This was defined as:[6]

$$RE_1 = \frac{\hat{D}_A - D_A}{D_A} \qquad \text{Eqn. 24}$$

This variable is zero if the DCET estimates defect content perfectly, negative if it underestimates, and positive if it overestimates. This is usually aggregated through the mean or median across multiple inspections.

However, it is $\hat{D}_R$ that is used to make the decision according to Eqn. 23, not $\hat{D}$ , therefore a more sensible criterion for evaluating a DCET that evaluates the accuracy of the variable that is used in actual decision making would be:

$$RE_2 = \frac{\hat{D}_R - D_R}{D_R} = \frac{\left(\hat{D}_A - D_F\right) - \left(D_A - D_F\right)}{D_A - D_F} = \frac{\hat{D}_A - D_A}{D_A - D_F} \qquad \text{Eqn. 25}$$

As will be noted, Eqn. 24 is equal to Eqn. 25 only if $D_F = 0$, which almost always is not the case. The consequence is that $|RE_1| < |RE_2|$, and hence the evaluations will seem better with Eqn. 24 than if Eqn. 25 is used.[7]

[6] Other studies, such as (Wolin and Runeson,1998), used the absolute relative error.

[7] This point was also alluded to in [Vander Wiel and Votta, 1993].

### 9.1.2 Unrealistic Assumptions About Inspected Documents and Estimation Models

As noted in the body of the paper, in making the reinspection decision, one wishes to control two aspects of quality:

- **The quality of the document.** By ensuring that the *documents* that pass the inspection attain a minimal quality level.
- **The quality of the inspection.** By ensuring that the *inspection process* has achieved a minimal quality level.

It is shown below that following the decision criterion in Eqn. 23 achieves neither of these objectives.

The decision criterion to control the quality of the document can be based on estimated residual defect density:

$$\frac{\hat{D}_R}{LOC} < Q_D \qquad \text{Eqn. 26}$$

where $Q_D$ is some threshold document quality level. The above decision criterion stipulates that a document must have a residual defect density that is lower than some accepted level. This accepted level would be defined as a constant for a project or an organization. This way, an organization can ensure that a document that passes inspection will be of a certain quality. For this purpose, the estimated residual defect density, $\frac{\hat{D}_R}{LOC}$, is used. The question then is what is the difference between the decision criterion in Eqn. 23 and the criterion that controls document quality defined in Eqn. 26 ?

It can be seen that Eqn. 23 is equal to Eqn. 26 only if $D_T = Q_D \times LOC$. If $D_T$ is a constant, then $LOC$ would also have to be a constant. The implication is that using the decision criterion in Eqn. 23 will consistently result in the same decisions as Eqn. 26 only if all documents that are inspected are of the same size. This is unlikely to be true in many cases as the documents that go through an inspection will not be of a uniform size.

The problematic aspect can be illustrated through an example. Let us assume that we have two documents, A and B. A is 600 LOC and B is 50 LOC. Further, let us assume that after an inspection, the DCET is used to predict 5 remaining defects in both documents, and that our threshold quality level, $Q$ , is 0.01 defects per LOC. In the former, the estimated defect density is approximately 0.008 while for the latter it is 0.1. According to the decision criterion in Eqn. 23 *both* documents would either pass or be reinspected. However, to control document quality, we should pass document A but reinspect document B.

The skeptic can argue that it may be clear, in an intuitive sense, that 5 remaining defects in a 50 LOC document would seem problematic to the inspection participants. While this may be true, it indicates that the decision is no longer objective, which defeats the whole point of using an objective DCET.

To control the quality of the inspection process, one can use its estimated effectiveness as follows:

$$\frac{D_F}{\hat{D}_A} > Q_P \qquad \text{Eqn. 27}$$

where $Q_P$ is some threshold inspection process quality level that is constant for the project or organization. The above decision criterion stipulates that an inspection must achieve an effectiveness that is larger than some minimal accepted level. This way, an organization can ensure that its inspections achieve a certain quality. We can then write the decision criterion of Eqn. 27 as:

$$\frac{D_F}{\hat{D}_A} = \frac{\hat{D}_A - \hat{D}_R}{\hat{D}_A} > Q_P \qquad \text{Eqn. 28}$$

It can be seen that Eqn. 23 is equal to Eqn. 28 only if $D_T = \hat{D}_A(1 - Q_P)$. If $D_T$ is a constant, then $\hat{D}_A$ would also have to be a constant. The implication is that decisions made using Eqn. 23 will be consistently the same as using Eqn. 28 only if DCETs always predict the same number of defects in all documents, which will not be the case, and if it was the case then the DCETs would be useless. Further, let's say we achieve the ultimate outcome of devising a DCET that has 100% accuracy (i.e., $\hat{D}_A = D_A$), then the assumption that would need to be made is that all documents that go through inspections have the same defect content before the inspection (i.e., are of the same quality). This is unlikely to be true in many cases as the documents that go through an inspection will not be of a uniform quality beforehand.

The problematic aspect can be illustrated through an example. Consider a stipulation that all inspections should attain at least 0.6 effectiveness. This means that 60% of the defects in a document are found during the inspection. Let's assume we have two documents, A and B. Document A has 30 defects in it before the inspection, and document B has 20 defects. During the inspections, 20 defects were found in document A, and 10 defects in document B. In both cases the number of remaining defects is 10. If we have a perfect DCET, we will know that the number of remaining defects is 10, and so we will make the same decision for both documents as to whether they should be reinspected using the criterion in Eqn. 23. However, the effectiveness of the inspection of document A is approximately 0.66, and for document B is 0.5. So, to control inspection process quality, we would reinspect document B to attain our threshold level of effectiveness. Therefore, using the criterion in Eqn. 23 does not actually control the effectiveness of the inspection process.

The above exposition has shown that the implied decision criterion from the literature does not necessarily control document quality nor inspection process quality. In order to do so, all documents that are inspected would have to be of the same size and of the same quality, or that DCETs always make the same prediction. Clearly, then the criterion in Eqn. 23 is not a good one. Below we consider some alternatives.

### 9.1.3 Possible Alternatives to Relative Error

We can propose that DCETs should be used to explicitly control one or both of the qualities in practice by using the criteria in Eqn. 26 and Eqn. 28. From the perspective of the previous literature that evaluates DCETs, we can consider alternative definitions of relative error that are congruent with the decision criteria in Eqn. 26 and Eqn. 28. For controlling document quality, we can define:

$$RE_3 = \frac{\left(\frac{\hat{D}_R}{LOC}\right) - \left(\frac{D_R}{LOC}\right)}{\left(\frac{D_R}{LOC}\right)} = \frac{\hat{D}_R - D_R}{D_R} = RE_2 \qquad \text{Eqn. 29}$$

Therefore, the previously used definition of relative error (Eqn. 24) provides accuracy results that are not congruent with the decision criterion in Eqn. 26.

For controlling process quality, we can define:

$$RE_4 = \frac{\left(\frac{D_F}{\hat{D}_A}\right) - \left(\frac{D_F}{D_A}\right)}{\left(\frac{D_F}{D_A}\right)} = \frac{D_A - \hat{D}_A}{\hat{D}_A} = \frac{RE_1}{1 + RE_1} \qquad \text{Eqn. 30}$$

This then would be the appropriate definition of relative error when one applies DCETs for making the decision as defined in Eqn. 28. As can be seen, this is inconsistent with the definition of relative error in Eqn. 24.

The above formulation would seem to indicate that the commonly used definition of relative error (defined in Eqn. 24) does not actually indicate an accuracy that is useful for controlling document quality and inspection process effectiveness.

### 9.1.4 Summary

We have shown that the commonly used criterion for evaluating DCETs, the relative error (as defined in Eqn. 24), does not actually give you an indication of whether the DCET can be used for controlling document quality nor inspection process quality when used in conjunction with the decision criterion of Eqn. 23. Furthermore, the implied practical decision criterion in the DCET literature (Eqn. 23) will not achieve this control if used directly.

## 9.2 Appropriate Criteria for Evaluating Accuracy

To resolve the difficulties unearthed above with the commonly used evaluation criterion, one can use *variable* thresholds for decision making. A variable threshold combines the specifics of a particular inspection with the expected product quality level $Q_D$ or process quality level $Q_P$.

We reformulate Eqn. 26 as:

$$\frac{\hat{D}_A - D_F}{LOC} < Q_D \qquad \text{Eqn. 31}$$

$$\hat{D}_A < (Q_D \times LOC) + D_F \qquad \text{Eqn. 32}$$

Therefore, the estimated number of defects in the document can be applied directly in the decision criterion. Also, the decision criterion maintains the desired document quality level by defining a different threshold depending on the document size. Similarly for inspection effectiveness, we can reformulate Eqn. 27 as:

$$\hat{D}_A < \frac{D_F}{Q_P} \qquad \text{Eqn. 33}$$

The above results also indicate that the currently used relative error criterion (Eqn. 24) is congruent with the manner in which DCETs are used in practice for controlling document quality and process effectiveness only if the decision criteria defined in Eqn. 32 and Eqn. 33 are used.

# 10 Appendix B: The Checklist

| Item no. | Where to look | How to detect |
|---|---|---|
| 1 | Data declaration | Are all variables declared before being used? |
| 2 | | If variables are not declared in the code module, will it be sure that these variables are global ones? |
| 3 | | Are the initializations of declared variables correct? |
| 4 | | Are the types of variables correct? |
| 5 | Data referencing | Are variables referenced that have no value (i.e., which have not been initialized)? |
| 6 | | Are the indices of arrays within the specific boundaries? |
| 7 | | Are all constants used correctly |
| 8 | Calculation | Are there any calculations with variables of different types? |
| 9 | | Does a calculation lead to underflow or overflow values? If yes, are these cases considered? |
| 10 | | Does a calculation involve a division by 0? |
| 11 | | Is it possible that a variable is assigned a value beyond its scope? |
| 12 | | Is the priority of operators considered (multiplication > addition)? |
| 13 | | Are the rules of arithmetik considered (2*A/2 must not be A!)? |
| 14 | Comparison | Are there any comparisons between variables of different type? |
| 15 | | Are the operators for comparison used correctly? |
| 16 | | Are the boolean operations correct? |
| 17 | | Does the boolean operations depend on evalution order? |
| 18 | | Are the values of the boolean operations correct (0 and 1)? |
| 19 | | Are there any comparisons between floating numbers and is this correct? |
| 20 | Control Flow | Does each loop terminate correctly? |
| 21 | | Is it possible that a loop is never executed (e.g., entry condition is never fulfilled)? |
| 22 | Interfaces | Are the number and types of variables in each function interface correct? |
| 23 | | Does a function modifies a variable although it should be call by value? |
| 24 | | Do the global variables have the same definition in each function they are used? |
| 25 | Memory | Are pointers used correctly (referencing and de-referencing)? |
| 26 | | Is enough memory allocated if necessary and de-allocated afterwards? |
| 27 | | Is too much or too little memory allocated? |

# Document Information

| | |
|---|---|
| Title: | The Application of Subjective Estimates of Effectiveness to Controlling Software Inspections |
| Date: | April 12, 1999 |
| Report: | IESE-031.99/E |
| Status: | Final |
| Distribution: | Public |