National Research Council Canada
Conseil national de recherches Canada
Institute for Information Technology
Institut de Technologie de l'information

ERB-1084

NRC·CNRC

A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content*

Briand, L.C., El-Emam, K., Freimut, B.G., and Laitenberger, O.
March 2001

Also published in IEEE Trans. Software Eng. 26(6): 518-540; 2000. NRC 44115.

Canada

NRC 44189

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de Technologie de l'information

A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content

Briand, L.C., El-Emam, K., Freimut, B.G., and Laitenberger, O.
March 2001

Abstract

An important requirement to control the inspection of software artifacts is to be able to decide, based on more objective information, whether the inspection can stop or whether it should continue to achieve a suitable level of artifact quality. A prediction of the number of remaining defects in an inspected artifact can be used for decision making. Several studies in software engineering have considered capture-recapture models, originally proposed by biologists to estimate animal populations, to make a prediction. However, few studies compare the actual number of remaining defects to the one predicted by a capture-recapture model on real software engineering artifacts. Thus, there is little work looking at the robustness of capture-recapture models under realistic software engineering conditions, where it is expected that some of their assumptions will be violated. Simulations have been performed but no definite conclusions can be drawn regarding the degree of accuracy of such models under realistic inspection conditions, and the factors affecting this accuracy. Furthermore, the existing studies focused on a subset of the existing capture-recapture models. Thus a more exhaustive comparison is still missing. In this study, we focus on traditional inspections and estimate, based on actual inspections' data, the degree of accuracy of relevant, state-of-the-art capture-recapture models, as they have been proposed in biology and for which statistical estimators exist. In order to assess their robustness, we look at the impact of the number of inspectors and the number of actual defects on the estimators' accuracy based on actual inspection data. Our results show that models are strongly affected by the number of inspectors and, therefore, one must consider this factor before using capture-recapture models. When the number of inspectors is too small, no model is sufficiently accurate and underestimation may be substantial. In addition, some models perform better than others in a large number of conditions and plausible reasons are discussed. Based on our analyses, we recommend using a model taking into account that defects have different probabilities of being detected and the corresponding Jackknife estimator. Furthermore, we attempt to calibrate the prediction models based on their relative error, as previously computed on other inspections. Although intuitive and straightforward, we identified theoretical limitations to this approach, which were then confirmed by the data.

Keywords: Inspections, Capture-Recapture Models, Robustness, Fault Content Estimation

1. Introduction

Software inspection is a proven approach that enables the detection and removal of defects in software artifacts soon after these artifacts are created (Ackermann et. al., 1989; Jones, 1996; Gilb and Graham; 1993). Inspections not only contribute towards software quality improvement, but can also lead to budget and time benefits through early defect detection.

A high quality inspection must ensure that most of the detectable defects in a software artifact are, indeed, detected. In practice, however, it has been shown that the effectiveness[3] of inspections can vary widely (Briand et. al., 1998). This variation is due to the fact that companies may not have implemented an optimal inspection solution given their environment and development situation. Other plausible causes are differences in artifacts being inspected and available resources to perform inspections. Whenever possible, a company should strive to optimize known inspection success factors, such as using more experienced inspectors or making sure the artifact is read at an appropriate rate (Porter et. al. 98). However, regardless of the specific situation at hand, one needs to make a decision whether the inspected artifact is of sufficient quality or whether a re-inspection is warranted. Making such a decision as objectively as possible is therefore a practical problem to address.

Three approaches have been suggested for deciding whether to re-inspect an artifact. The first one is to let inspection participants make the re-inspection decision at the end of the inspection meeting (Strauss and Ebenau, 1993). A recent study

[3] In this paper effectiveness is defined as the proportion of total defects in a document that are detected during inspections.

suggests that subjective estimates of inspection effectiveness have good accuracy (median relative error of zero) and can be applied to make the re-inspection decision (El Emam et al., 2000). However, it was noted that complete reliance on subjective estimates dilutes the transparency of the decision making process, and therefore objective re-inspection criteria are also necessary.

The second approach requires a comparison of inspection results with representative benchmarks, e.g., a document is re-inspected for a second time if the number of defects is significantly different from the historical average (Eick et. al.,1992). Too many defects would indicate a poor document, and too few defects a poor inspection. With this approach, however, a high-quality document may be re-inspected, and a poor-quality document may not be re-inspected if the inspection is performed poorly. Therefore, such an approach is reliable only when defect densities are roughly constant across inspected artifacts.

The third approach is similar to the second one but uses upper and lower thresholds on the number of defects found per unit of size (Vander Wiel and Votta, 1993). The lower limit is set to detect poor quality inspections and the upper limit for detecting low-quality documents. Following this approach, however, raises the risk that inspectors are tempted to only find a passing number of defects regardless of the document's quality.

Although more objective than the first one, the second and third approach require historical data to define a benchmark or quality-thresholds. In practice, such data may not be available or may be difficult to obtain. These difficulties make the case for exploring a fourth approach.

The fourth approach is to use the number of detected defects in the software artifact to estimate how many defects are remaining. This estimate is then used as the basis for deciding whether to reinspect. Since it is impossible to count the total number of defects in a system before it has been in operation for a while, it is necessary to build estimation models of the number of defects in a software artifact. This estimation problem is similar to the problem of estimating animal abundance in biology and wildlife research. For example, knowing the population size of deer is essential for deciding on the number to be released for shooting.

A solution to this problem in biology is to use capture-recapture models (White et. al., 1982; Otis et. al., 1978): animals are captured, marked, and released on several trapping occasions. If an animal bearing a mark is captured on a subsequent trapping occasion, it is said to be recaptured. Based on the number of marked animals that are recaptured one can estimate the total population size using statistical models and their estimators. When many marked animals are recaptured, one can argue that the total population size is small and vice versa. Other solutions have been proposed recently in software engineering, that is, graphical methods (Wohlin and Runeson, 1998) and an additional capture-recapture estimator (Ebrahimi, 1997). They are not investigated here as we have restricted ourselves to the set of estimators proposed in biology. These estimators have undergone extensive simulation studies in biology but have only been partially investigated in software engineering.

The capture-recapture principle in biology can be transferred to inspections: each inspector draws a sample from the population of defects in the inspected software artifact. In this way, an inspector is equivalent to a particular trapping occasion in biology. A defect discovered by one inspector and rediscovered by another is said to be recaptured. Based on estimators similar to the ones used in biology, the total number of defects in the software artifact can be estimated.

Thus far, few empirical studies on software engineering artifacts have performed a comprehensive comparison of the relevant, state-of-the-art capture-recapture models for software inspection. In addition, very seldom is the *actual* number of defects in a real software engineering artifact compared to the one predicted by a capture-recapture model. Such comparisons are important because each capture-recapture model makes certain assumptions that are violated in an inspection context[4]. Hence, empirical studies are necessary to assess whether one or several of the models are robust to these violations in representative software engineering conditions, i.e., provide useful results despite violating some or all of the underlying assumptions.

In this paper, the performances (i.e., the accuracy and the failure rate) of relevant, state-of-the-art capture-recapture models and their corresponding estimators are empirically evaluated using actual software engineering data from the

[4] Some of the assumptions are also violated when the models and their estimators are applied in biology.

inspection of requirements documents. For this evaluation, the number of inspectors and the number of total defects are considered. Based on this evaluation's results, recommendations are made on which models and estimators to use and under which circumstances.

Briefly, our results indicate that, overall, the capture-recapture models and their evaluated estimators tend to underestimate the number of defects. In particular, using a small number of inspectors (defined as less than four) will lead to rather inaccurate estimates. Using calibration to improve the models' accuracy has some theoretical limitations that were confirmed by our data. Furthermore, the recommended estimator (referred to as Jackknife Estimator) is based on a model taking into account that defects have different probabilities of being detected.

This paper is organized as follows. Section 2 provides a survey of capture-recapture models used in biology, and their application in software engineering. Section 3 describes how different capture-recapture models were evaluated in our study. In Section 4 the results of the evaluations are presented, as well as recommendations on which capture -recapture models and estimators to use based on our findings. Section 5 discusses the overall results, and concludes the paper with directions for future work.

2. Review of Capture-Recapture Models and Their Application to Software Inspections

This section provides the basic concepts of capture-recapture models, a discussion of the applicability of existing models to software inspections, and a review of the work that has already been performed in a software engineering context.

2.1 Basic Concepts of Capture-Recapture Models

In biology, capture-recapture studies are used to estimate the size of an animal population. In doing so, animals are captured, marked, and then released on several trapping occasions. The number of marked animals that are recaptured allows one to estimate the total population size based on the samples' overlap. As an example of such an estimation procedure, consider the following (see for example Dudewicz, 1988): suppose one wants to estimate the size N of a population that does not change over time, i.e., no animals enter or leave the population through birth, death, immigration, or emigration. A number of n_1 animals are captured on a first day. These animals are marked somehow and released into the population. After allowing some time for the marked and unmarked animals to mix, a second trapping occasion is performed on a second day. On this day, n_2 animals are captured. This sample of n_2 animals consists of m_2 animals bearing a mark (animals captured on both days) and n_2-m_2 animals without a mark (newly captured animals). Assuming that the ratio of marked to total animals in the second sample is equal to the ratio of marked to total animals in the entire population, the so-called Lincoln-Peterson Estimator for the number of animals in the population can be derived (Seber, 1982 and White et. al., 1982):

$$\hat{N} = \frac{n_1 \times n_2}{m_2} \qquad \text{Equation 1}$$

The idea behind using capture-recapture models for software engineering inspections is to let several inspectors read the same document, i.e., draw independent samples from the population of defects. Based on the overlap of defects amongst inspectors, one can estimate the number of defects remaining in a software artifact. By subtracting the number of defects that was actually found during inspection from the estimated number of defects, one can calculate the number of remaining defects. Taking into account this number of remaining defects, one can decide on a more objective basis whether the software artifact has to be reinspected.

2.2 Applicability of Capture-Recapture Models for Inspections

Capture-recapture models make certain assumptions that may differ between biology and software inspections. Thus, before using the models it is necessary to investigate the various assumptions in biology and assess their validity for inspections. The basic assumptions of the Lincoln-Peterson Estimator are:

(a) *Number of trapping occasions*: *only two trapping occasions are performed.* The number of trapping occasions is analogous to the number of inspectors in inspections. For inspections, however, one often wants to include more than two inspectors.

(b) *Closure: no animals must leave or enter the population during the study.* The number of animals in the population is equivalent to the number of defects in a software artifact. For inspections the assumption of closure is valid since all inspectors usually inspect the same software artifact and thus face the same population of defects.

(c) *Capture probability: all animals are equally likely to be caught in each trapping occasion.* The capture probability is analogous to the defect detection probability in inspections. Therefore, all defects are assumed to have the same detection probability. This assumption may be violated, for example, when there are some defects that are easier to detect than others.

For practical purposes, the first two assumptions can be easily addressed. For instance, more than two trapping occasions can be managed by calculating a weighted mean (Begon, 1979). Also, the Lincoln-Peterson Estimator can still be applied when the assumption of closure does not hold: when animals enter or leave the population, only an estimate for the population on the second trapping occasion is provided. It is the third assumption that requires most attention.

Various models and corresponding estimators have been developed and proposed in biology to alleviate the effects of these assumptions (see Pollock, 1991 for an overview). The most important models one can consider for inspections have been described by Otis et. al. (1978) and White et. al. (1982). They present a set of closed models that can deal with more than two trapping occasions, and that allow for a varying capture probability. The idea behind this set of models is that sources of variation are modeled which relax the assumption of equal capture probabilities.

The first source of variation is called *time response*. In biology, it models the fact that on different days the capture probabilities of animals might vary. For example, small mammals tend to stay in their dry homes during rainy weather. Therefore, the probability of capturing a small mammal is higher for days with fine weather than for days with rainy weather. For inspections this can be used to model inspectors with different abilities to detect defects. For example, experienced inspectors find more defects than inexperienced inspectors and therefore have a higher probability of detecting defects.

The second source of variation is called *heterogeneity*. In biology, it models the fact that different animals vary in their capture probability. For example, older animals are less mobile than younger ones and stay more often in their homes. Therefore, the probability of capturing an old animal is smaller than of capturing a young animal. For inspections this can be used to model defects that differ in their detection probability. For example, defects that are difficult to detect have a lower detection probability than defects that are easy to detect.

In addition to these two sources of variation, Otis et al. and White et al. consider variations due to *behavioral* or *trap response*. This reflects the fact that an animal may change its behavior due to the process of being captured and marked. For example, when using baited traps, the probability to get caught for the first time is less than the probability for subsequent captures. This is because animals can get fascinated by traps, so marked animals are more likely to get caught than unmarked animals (Otis et. al., 1978). In an inspection context, this may be usable to model the fact that defects captured by more than one inspector have usually a higher probability of being detected. However, the estimators for this source of variation depend on the order of trapping occasions (i.e., inspectors). Since no ordering of inspectors seems reasonable in the context of inspections, these estimators are not considered adequate.

With the two relevant sources of variation one can take into account the fact that both inspectors and defects can affect the defect detection probability. Inspectors may have different detection capabilities due to variation in their ability to detect defects (due to experience or education) and defects may have different detection probabilities when there are defects that are easier to detect than others.

Based on these two sources of variation, four capture-recapture models can be formulated. Model M0 assumes that none of these sources of variation is included, Model Mt and Mh account for exactly one source of variation, and Model Mth accounts for both sources of variation. When the analogy is made to inspections, these models make the following assumptions about inspectors and defects:

(a) Model M0 - No variation: This model assumes that every defect has the probability p of being detected by every inspector. Thus, all defects have the same detection probability, and all inspectors have the same detection capability.

(b) Model Mh - Variation by heterogeneity: This model assumes that every defect j has the probability p_j of being detected, which is the same for every inspector. Thus, different defects can vary in their detection probability, but all inspectors have the same detection capability. For instance, in Vander Wiel and Votta (1993) it is reported that inspectors often classify defects as easy or hard to detect. The Mh type of model accounts for this source of variation.

(c) Model Mt - Variation by time response: This model assumes that every inspector i has the probability p_i of detecting every defect. Thus, all different defects have the same detection probability, but the inspectors have different detection capabilities. Hence, with this source of variation accounted for, a model allows for inspectors with differing "general ability". Note that this "general ability" affects all defects[5].

(d) Model Mth - Two sources of variation are combined: time response and heterogeneity. This model assumes that every defect j has the probability p_j of being detected and that every inspector i has the probability p_i of detecting defects. The probability p_{ij} that inspector i detects defect j is computed as $p_{ij} = p_i\ p_j$. This allows for different detection probabilities for the different defects and inspectors.

These four models build a partial ordered hierarchy. Model M0 is a special case of Model Mt, Model Mh, and Model Mth. Both Model Mt and Model Mh are special cases of Model Mth.

Figure 1 graphically illustrates the assumptions of the four models. In these graphs, the bars on (x,y) denote the probability that inspector x detects defect y.

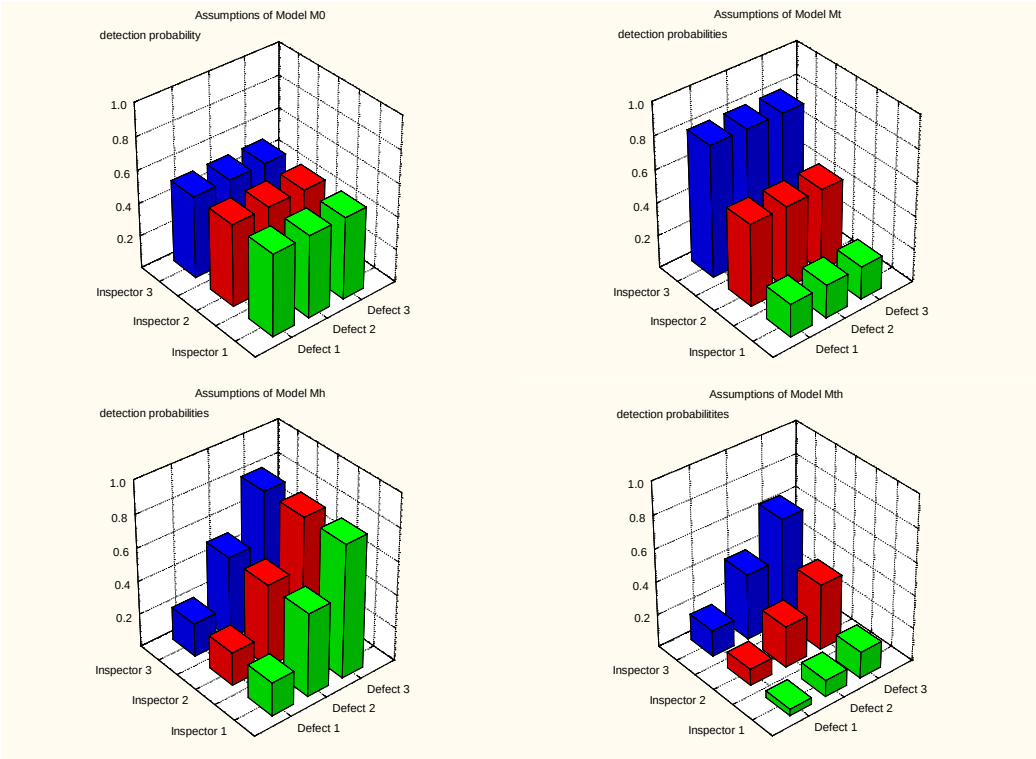


Figure 1**:** Graphical depiction of capture-recapture models assumptions in a software inspection context

[5] The Lincoln-Peterson Estimator is an estimator for this kind of model

When applying capture-recapture models for estimating the number of defects, suitable estimators are necessary. While the model defines the assumptions made about detection probabilities, the corresponding estimator is a formula that actually performs the estimation based on the model's assumptions. In order to derive these estimators, the models' assumptions have to be cast into a stochastic form. Using estimation techniques, such as maximum likelihood estimation (MLE), estimators can be derived. Since there are different estimation techniques, several estimators can be derived for a given capture-recapture model each of which requires a particular set of information to be collected in an inspection.

The estimators investigated in this paper are summarized in Table1:

Model	Estimator	Notation
M0	Maximum Likelihood Estimator (Otis, 1978)	M0(MLE)
Mt	Maximum Likelihood Estimator (Otis, 1978), Chao's Estimator (Chao, 1989)	Mt(MLE) Mt(Ch)
Mh	Jackknife Estimator (Burnham and Overton, 1978), Chao's Estimator (Chao, 1987)	Mh(JE) Mh(Ch)
Mth	Chao's Estimator (Chao et. al., 1992)	Mth(Ch)

Table 1: Relevant capture-recapture models and considered estimators.

The most general information needed by those estimators is displayed in Figure 2. For each defect one must note which inspectors detected this defect.

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1k} \\ x_{21} & x_{22} & \dots & x_{2k} \\ \dots & \dots & \dots & \dots \\ x_{D1} & x_{D2} & \dots & x_{Dk} \end{pmatrix}$$

x_{ij}=1 if and only if, inspector i detected defect j

Figure 2: Data Matrix with most general data required by capture-recapture estimators

However, once a suitable estimator has been identified, only those data required by this estimator have to be collected instead of collecting the whole data matrix as in Figure 2. The minimum set of data for each estimator is listed in the table below.

Estimator	Data Requirements
M0(MLE), Mt(MLE)	For each inspector: the number of defects detected by that inspector
Mt(Ch)	For each defect: the number of inspectors detecting that defect For each inspector: the number of defects exclusively detected by that inspector
Mh(JE), Mh(Ch)	For each defect: the number of inspectors detecting that defect
Mth(Ch)	For each inspector: the number of defects detected by that inspector For each defect: the number of inspectors detecting that defect

Table 2: Detailed data required by individual capture-recapture estimators

Presenting the complete formulae for all the estimators would be beyond the scope of this paper. However it is worth mentioning that, although the derivation of the estimators may be complex, their implementation and use are straightforward. In some cases the estimate can even be obtained by using a simple pocket calculator.

The models and their corresponding estimators have been evaluated in the biological context with respect to their accuracy and robustness to assumption violations by means of Monte-Carlo Simulation. These results, however, cannot be directly applied to inspections as realistic parameter ranges may differ significantly, e.g., capture probabilities, and there may be very different levels of compliance with models' assumptions.

For inspections, we expect that the inspection data violates some of the assumptions of the models. For example, we have to assume that some inspectors find more defects than others (indicating time response) and that some defects are more difficult to find than others (indicating heterogeneity). Even Model Mth, though it includes all relevant sources of variation, is not an ideal representation for inspections. This is due to the fact that the structure of the model implies that the detection capability p_i of the inspector affects all defects *similarly*. Thus, this model does not cover the case in which one or more inspectors focus on finding specific types of defects, as it is assumed, for example, with Perspective-based reading (Basili et al., 1996). Despite all these likely assumption violations, the question is to determine how robust are the models in typical software engineering conditions.

The number of animals and the number of trapping occasions in the Monte Carlo Simulations performed in biology are different from the numbers we would expect in an inspection context. Therefore, the properties of these estimators should be investigated using more realistic numbers of defects (animals) and inspectors (trapping occasions). This would allow us to determine how robust the capture-recapture estimators are to assumption violations present in representative inspection data.

2.3 Studies of Capture-Recapture Models in Software Engineering

2.3.1 Existing Studies

The first use of capture-recapture models in software engineering was due to Mills (1972). He proposed an estimation of defects in a system based on defect discoveries in the testing phase. His approach was to seed pseudo defects before testing. During testing, a tester detects pseudo defects and real defects. Applying the Lincoln-Peterson Estimator to the number of seeded pseudo defects, the number of detected pseudo defects, and the number of detected real defects gives an estimate for the number of total defects. However, using the Lincoln-Peterson Estimator requires that the seeded and real defects have the same detection probabilities.

Based on Mill's approach, several people have used capture-recapture models with seeded defects for estimating software reliability. However, according to Musa et al. (1984), this fails to be accurate due to the difficulties in seeding the software with defects similar to those naturally occurring. They argue that seeded defects are much easier to find.

Basin (1973) introduced a similar approach. Like Mills' approach, his estimation was based on the Lincoln-Peterson estimator. But instead of seeding defects and using one tester, Basin used two testers. The defects detected by the first tester were regarded as "marked defects" for the Lincoln-Peterson Estimator.

Eick et. al. (1993) describe the first application of capture-recapture methods for software inspections. They use capture-recapture models during design inspections to predict defect content. Like Basin's approach, no artificial defects were seeded into the inspected document. Instead, prior to each inspection meeting, the inspectors search independently for defects. Eick et. al. used the Model Mt for defect content prediction. Since they had no software artifact with a known number of actual defects, Eick et. al. asked the inspectors for their intuitive opinion about the plausibility of Mt's estimates. The result was that the estimations were consistent with the inspectors' intuition, i.e., a software artifact with a low estimated number of defects was perceived to contain a low number of defects by the inspectors.

The problem with Model Mt is that it assumes that all defects have equal defect detection probabilities. But since in software engineering defects vary with respect to their defect detection probability, Vander Wiel and Votta (1993) compared the Model Mt with a model which allows for different detection probabilities of defects, i.e., the Model Mh. They performed a Monte-Carlo simulation to investigate the accuracy of two estimators for these two models. They observed that Mt performed better than Mh and can be improved by grouping defects into classes wherein the assumption of equal detection probability for different defects is more justifiable.

Based on these findings Wohlin et. al. (1995) propose two classification techniques, referred to as “filters”. A filter groups defects into classes to improve the accuracy of the models. Wohlin et. al. propose a “percentage filter” and an “extreme filter” that are defined based on experience. With the percentage filter, given a percentage value x, the defects are divided into two classes. The first class contains defects that are found by more than x% of the inspectors The second one contains all defects found by less than x% of the inspectors. For the extreme filter, all defects found by exactly one inspector are put into one defect class, and the remaining in a second class.

These filters were tested in an experiment. However, instead of using software artifacts, the inspectors read a document with grammatical and spelling errors. Wohlin et al. (1995) report that when estimating without grouping, the number of defects was underestimated. Applying the percentage filter (x=40%) improved the estimates compared to the ones without the filter but still underestimated the known number of defects. Applying the extreme filter lead to overestimation. However, for both filter techniques, thresholds or parameters have to be provided (e.g., the percentage x for the percentage filter) whose values affect the performance of the filter technique. An approach to decide upon these parameters is presented by Runeson and Wohlin (1998). In this article they present an empirical approach to set parameters and improve the extreme filter based on experience. They conducted a C-code inspection experiment to evaluate the experienced-based approach. The results of the study show that the experience-based approach provides significantly better results than estimates without the filter technique for the Mt-model with the Maximum Likelihood Estimator.

Recently, two new estimation approaches have been proposed in addition to the capture-recapture models used in biology. First, Ebrahimi (1997) proposed a capture-recapture model that is an extension of the Model Mt. Model Mt, like all other capture-recapture models, assumes the inspectors to work independently from each other. This independence assumption might be violated when inspectors collaborate (resulting in more defects found in common) or specialize on certain defect types (resulting on fewer defects found in common). Ebrahimi’s estimator allows us to account for these cases where inspectors do not look for defects in an independent manner.

Second, Wohlin and Runeson (1998) proposed two alternative approaches which are distinct from the concept of capture-recapture models. Their idea is to determine for each defect the number of inspectors detecting that defect, to sort these data according to some criterion, and finally to fit a curve through these data points. One of these approaches, referred to as the Detection Profile Method, did not show significantly better results than the MLE for Model Mt. An improvement of this Detection Profile Method, along with a selection procedure between this method and the biological capture-recapture models, has been presented in Briand et al. (1998b)

These two recent estimation approaches will not be considered in our analysis as we have restricted ourselves to the capture-recapture models defined in biology. One reason for that is that we do not expect Ebrahimi’s estimator to be relevant in our study as the inspectors neither collaborated nor specialized on certain defect types. In our case, Ebrahimi’s estimator is expected to provide a similar result to that of MLE for Model Mt (Ebrahimi, 1997).

2.3.2 Open Research Issues

The current state of knowledge about capture-recapture models for software inspections can be improved by expanding on the scope of previous studies. Specifically, four issues are addressed in this study:

1. Performing a comprehensive comparison of relevant, state-of-the-art models and different estimators for these models.
2. Using actual software engineering data.
3. Investigating the impact of the number of inspectors on the performance of models.
4. Investigating the impact of the total number of defects in the document on the performance of the models.

Evaluating Capture-Recapture Models and Different Estimators

In the preceding section the four capture-recapture models relevant for inspections have been introduced. So far research on software inspections has only considered two of these models, with one estimator for each (the Maximum Likelihood Estimator (MLE) for Model Mt and the Jackknife Estimator for Model Mh).

First of all, it is interesting to look at the existing set of relevant models and estimators. Especially interesting for inspections is the model that incorporates both different probabilities to detect defects and different detection probabilities for inspectors (Model Mth) because this represents the closest match of assumptions and the situation in many inspection implementations. Furthermore, the estimators derived by Chao for Model Mh (Chao, 1987) and Model Mt (Chao, 1989) have been designed for situations in which many animals were detected only once or twice. These estimators may be appropriate when performing inspections with a few inspectors, which is a typical situation in software engineering, or when the inspectors have low defect detection capabilities.

Type of Data

If we are to make informed decisions about using capture-recapture models in software engineering practice we need to evaluate all the presented models and their estimators with data from inspections that were performed using real software artifacts. Hence, we investigate the accuracy of capture-recapture models and their estimators using data from the inspection of software artifacts, which we regard as representative in terms of defect types, inspectors' skills, and documents' complexity. In our investigation, we considered the number of inspectors and the number of defects as variation factors.

Number of Inspectors

One important element of applying capture-recapture models is the number of inspectors involved. The larger the number of inspectors, the larger the amount of information available for estimation, and consequently the more accurate the estimates are expected to be. However, using a large number of inspectors may not be feasible for practical and economical reasons.

Therefore it is necessary to assess the impact of the number of inspectors on the performance of capture-recapture models. This allows one to make a trade-off between accuracy and the number of inspectors (i.e., the cost of the inspection).

In texts dealing with the biological application of capture-recapture models, a number of five trapping occasions (equivalent to five inspectors in software engineering) is recommended as a rule of thumb, though a number of 7 or 10 was deemed more appropriate (Otis et. al., 1978; White et. al.; 1982). However, no quantitative justification or evidence is provided for this recommendation.

In the inspections literature, the reported number of inspectors that typically participate in an inspection varies. Bisant and Lyle (1989) have found performance advantages in an experiment with two persons: one inspector and the author. Weller presents some data from a field study using three to four inspectors (Weller, 1993). Bourgois presents data showing that the optimal size is between three and five people (Bourgeois, 1996). Such a variation and the current lack of quantitative evidence on the impact of the number of inspectors warrants a thorough investigation.

Number of Defects

Another factor that can influence the accuracy of a capture-recapture estimate is the total number of defects in a document. If there is a small number of defects in the document, the number of defects found by inspectors can only cover a narrow range. Thus, only little information can be used for estimating and the estimate's accuracy might be adversely affected.

In the literature on inspections, the number of defects is not explicitly considered as a factor. Eick et. al. (1992) present the results of 13 inspection meetings, where the number of discovered defects ranges from approximately 15 to approximately 200. For the majority of inspections, however, the number of detected defects was in the range of 20 to 50. It must be kept in mind that these numbers represent the number of discovered defects and not the number of actual defects, which is likely to be higher. In (Eick, 1993) the authors present the data matrix of an inspection with a document containing

47 actual defects. Wohlin et. al. (1995) seeded 38 defects in the documents they used for their experiment and the number of defects was not systematically varied to determine its impact on the estimator's accuracy. Finally, Runeson and Wohlin (1998) used several C programs with a various number of defects. Although the number of defects they report (i.e., 16-35 defects) is within the range that can be considered realistic, they do not perform a sensitivity analysis in which they consider this a factor.

In the biological context of capture-recapture studies, the population is usually much higher, ranging from 50 to several hundreds or even thousands of animals (White et. al., 1982). Often capture-recapture methods are employed especially due to the large size of a population. For example, Begon (1979) reports an example where the number of mosquitos in a specific area was to be estimated. In this context, typical population sizes for simulations evaluating estimators for biological purposes range from 100 to 400.

In this paper the number of actual defects per document is systematically varied between 6 and 24. The number of defects in this range is lower than the numbers of defects reported or used by Eick et al. and Wohlin et al. (1995) but can be considered as realistic, although they might represent a partial range for software inspections.

3. Research Method

For evaluating capture-recapture models in the context of inspections we take inspection data (i.e., which inspector detected which defect), apply the estimators described above, and evaluate the models according to some evaluation criteria while considering factors that might have an impact on the evaluation. Section 3.1 describes the inspection data that we used. Section 3.2 introduces selected evaluation criteria. Section 3.3 describes how the impact of the number of inspectors and the number of defects were investigated. Finally, Section 3.4 describes how to select the most appropriate capture-recapture model under different conditions.

3.1 Data Set

The data used in our evaluation comes from two experiments that evaluated different reading techniques for defect detection in requirements documents. These experiments were performed between 1994 and 1995 at the NASA/Goddard Space Flight Center (NASA/GSFC) (Basili et al., 1996).

3.1.1 Inspection Process

The goal of the experiments was to compare reading techniques, namely the company specific reading approach denoted as ad-hoc and perspective-based reading. Basili et al. focused exclusively on the defect detection step of an inspection process. Each inspector analyzed one particular requirements document according to a specified reading technique. Neither inspection meetings nor corrections to the inspected documents were performed as this was not necessary to achieve the experimental objectives.

3.1.2 Inspectors

The inspectors were software professionals at NASA/GSFC with various levels of experience in the application domain and the development techniques used. The first experiment was conducted with 12 and the second one with 14 professionals. Since in the second experiment one subject was not familiar with NASA flight dynamics applications, data from only 13 professionals were analyzed.

3.1.3 Type of Software Artifacts

The artifacts under study were requirement documents. These were structured according to the IEEE 830-1993 standard (IEEE, 1994) and the different requirements were stated in natural language. Two different sets of requirement documents were used:

- Two generic documents developed for educational purposes. These were the requirements for an automated teller machine (ATM) and a parking garage system (PG). The ATM document was 17 pages long and contained 29 defects. The PG document was 16 pages long and contained 27 defects. The defects were the ones made while developing these documents.
- Two NASA/GSFC documents consisting of functional specifications for satellite ground support software. Both documents were 27 pages long and contained 15 defects (one of the NASA documents initially contained 18 defects). The defects were the ones detected during subsequent development phases.

Before the experiments, the defects were seeded (i.e., reintroduced) in the documents. Since the defects were known in advance, false positives were not an issue in this study. However, a particular defect could manifest itself in several places within the requirements document. As a consequence, two or more inspectors could localize the same defect in different places and describe it differently, which could bias some of the capture-recapture estimates. To resolve this dilemma and to ensure consistent counting rules, two individuals consolidated the defect lists that inspectors filled out in the experiments so as to avoid duplicate defect counts.

To investigate some of the experimental hypotheses about reading techniques, the defects were classified according to different defect types (i.e., omission, ambiguous information, incorrect fact, extraneous, and miscellaneous). However, the defects were not considered differently with respect to their criticality or severity. Hence, we do not need to distinguish among different sets of defects in our study.

3.1.4 Experimental Procedure

Two experiments were performed. Each experiment consisted of two reading sessions in each of which the participants either inspected the document pair NasaA/ATM or the document pair NasaB/PG. In our study, we only consider the data from the first reading session (ad-hoc reading). Moreover, we treat both experiments independently (i.e., we take it that there were eight different documents) since the documents were modified for the second experiment. The modification was performed because Basili et. al found a few inconsistencies in some of the specifications after running the first experiment and, therefore, changed a few sentences to make them less ambiguous. The modification also explains the different number of defects in the NasaANov and NasaAJun document as can be seen in Table 3 [6].

Document Name	Number of Actual Defects	Number of Inspectors
AtmNov	29	6
AtmJun	29	8
PgNov	27	6
PgJun	27	6
NasaANov	18	6
NasaAJun	15	7
NasaBNov	15	6
NasaBJun	15	6

Table 3: Documents used during both experiments

[6] After running both experiments, the generic documents were updated and improved as well. Some more defects were introduced, which explains why the PG document in the replication package has now 30 defects.

3.2 Evaluation Criteria

To evaluate the different models and estimators, one has to determine the accuracy of the estimates. We used as a measure of accuracy the relative error (RE). The RE is defined as follows:

$$\mathrm{RE} = \frac{\text{estimated no. of defects} - \text{actual no. of defects}}{\text{actual no. of defects}} \qquad \text{Equation 2}$$

The RE allows one to distinguish between overestimation (too many defects are estimated, yielding a positive RE) and underestimation (too few defects are estimated yielding a negative RE). An RE of zero indicates that the estimate is perfectly accurate.

When dealing with the accuracy of estimators, two properties should be investigated: Bias and Variability.

- *Bias of the Relative Error*
  The bias of an estimator's RE tells us how accurate the estimates are on average. Bias can be expressed as the central tendency across the population of estimates, e.g., the mean or the median. A drawback regarding the mean is that it is sensitive to extreme values or outliers. A first look at the maximum RE values (see Figure 3) for each estimator shows that some estimators (especially Chao estimators) indeed have large maximum values. Therefore, bias is defined here as the median RE. As discussed further below, calibration might help improve the models' bias.
- *Variability of the Relative Error*
  The variability of an estimator's RE tells us whether a large variation around the central tendency can be expected. The inter-quartile range and the presence of extreme outlier values were used as measures of variability.

In addition to these performance evaluation criteria, it has to be taken into account that estimators may not provide estimates under all conditions. Since the most accurate estimator cannot be used alone when it fails to yield an estimate in a large number of cases, an important performance measure that should be looked at is how often an estimate cannot be obtained. Therefore, we define an additional criterion:

- *Failure Rate*
  The failure rate is defined as the percentage of cases where no estimate is produced.

3.3 Investigation Strategy for Studying the Performance of Capture Recapture Models

In the preceding sections two issues were identified that might have an impact on an estimate's accuracy: the number of inspectors and the actual number of defects. Since only a fixed number of inspectors and defects for each document were available, these factors were varied by creating "virtual inspections". A virtual inspection is created by randomly selecting a set of defects and a set of inspectors. The general approach is, for a given number of inspectors, to select from the total number of inspectors and defects in order to define one instance of a virtual inspection.

Combining inspections with a varying number of inspectors

To calculate the accuracy for inspections with k inspectors, the number of defects was estimated for all documents for all possible combinations of k inspectors. For example, if for a document a total of six inspectors was available and inspections with three inspectors were investigated, 20 virtual inspections were then formed. For each of these virtual inspections an estimate was obtained with each estimator. This is repeated for each of the eight documents.

The number of inspectors (previously denoted as k) was systematically varied between two and six. This range covers the numbers of inspectors reported in Section 2.3.2 and can therefore be considered as representative of real-world inspections. The bias for a given document is defined as the median RE for all combinations of k inspectors for that document. A single number characterizing the bias for each estimator had to be computed as well. This was computed as the estimator's median

RE for all combinations of k inspectors for all documents. The differences between these two figures is shown in Table 4, considering two documents with an actual number of three inspectors but combining inspection teams with two inspectors.

No.	Virtual Inspection	Bias for given document and estimator	Bias for given estimator
1	Document A , Inspectors 1,2		
2	Document A , Inspectors 1,3	Bias for estimator and Document A:	
3	Document A , Inspectors 2,3	Median (1,2,3)	Bias for estimator:
4	Document B , Inspectors 1,2		Median (1,2,3,4,5,6)
5	Document B , Inspectors 1,3	Bias for estimator and Document B:	
6	Document B , Inspectors 2,3	Median(4,5,6)	

Table 4: Estimation of bias for 2 documents and 3 inspectors

Combining inspections with a varying number of defects

In addition to the number of inspectors involved in an inspection, the number of defects that are in the inspected document is important. If the number of defects is too small, too little information for estimating might be available. In order to investigate the impact of the number of defects on accuracy, inspections with varying numbers of defects were combined. In contrast to the combination procedure with varying numbers of inspectors, it is much more difficult to compile all combinations for a given number of defects. This is due to the fact that performing all combinations from 15 or 28 defects would result in a very large number of estimations. Nevertheless, in order to obtain a relevant number of representative inspections 50 combinations were randomly selected from all possible combinations.

Unfortunately, the documents had different numbers of defects. While the two generic documents had 27 and 29 defects, respectively, the two NASA documents had only 15 and 18 defects. Therefore, the NASA documents could not be used for simulating inspections with more than 15 defects. For the NASA documents the number of defects was systematically varied between six and 12 defects in steps of two defects and for the generic documents the number of defects was systematically varied between six and 24 defects in steps of two defects.

This simulation approach assumes that the detection probability of a specific defect is independent of the number of defects in the document. This is a valid assumption since the defects were injected in a way that ensures their independence.

When simulating inspections a specific number of inspectors had to be considered. Two representative numbers of inspectors, namely four and six inspectors, were selected for simulating a varying number of defects. The median RE was then computed for all selected 50 combinations, given a number of defects and inspectors across all documents.

3.4 Selection of the Best Model

The objective of this selection is to make a recommendation on the most appropriate capture-recapture model(s) to use under different conditions described by the factors in Section 2.3.2 using the evaluation criteria presented in Section 3.2. The decision procedure for selecting the best model(s) consists of two steps. First, we characterize the conditions under which it is not recommended to apply capture-recapture models. Second, for the remaining situations, the most appropriate model is identified.

Conditions of Applicability

We first identify a threshold for the minimal number of inspectors and defects that are to be used in order to yield a reasonable degree of accuracy in terms of both the bias and variability of estimates. Although such a threshold is somewhat subjective, a bias of more than +/- 20% was considered too large to be practical. In addition, models that are occasionally producing extreme predictions (i.e., extreme outliers) were considered as too risky for practical use.

Statistical Testing and Multiple Comparison Procedure

Capture-recapture models are then compared through statistical testing. First, for each model we try to select one estimator. Second, we compare the different models in order to identify those sources of variation that are most important in an inspection context.

Selecting one estimator per model

Since we have two estimators for Model Mh and two estimators for Model Mt, we first compare estimators for the same model. If they are equally accurate then they are considered both for the subsequent analysis. If one is better than the other, then only the best one is considered.

One possible approach for testing would be to use the RE values for all virtual inspections. However, this is not appropriate since virtual inspections containing the same inspectors cannot be considered statistically independent. For example, a virtual inspection with inspectors 1, 2, and 3 will yield similar results to a virtual inspection with inspectors 1, 2, and 4. Therefore, we determine for *each document and estimator* the bias and use this value for testing. Thus, the sample size for statistical testing is determined by the number of documents.

Moreover, we do not consider the bias of the estimator for a given document but the *absolute* bias. The absolute values are used here because we make in this study no formal distinction between over- and under- estimation, i.e., a comparable cost is assumed for both situations. This is due to the fact that we do not perform our study in a specific development environment and it is therefore not possible to come up with a specific loss function for over- and under- estimation. However, in practice, underestimation of the number of remaining defects may be substantially more harmful than overestimation since it leads to insufficient effort spent on inspections and poor quality artifacts.

The statistical test between two estimators for the same model is performed by using a two-tailed t-test for paired samples (Lapin, 1978) on the *absolute* bias for the eight documents. In some cases, however, the t-test provided non-significant results although the differences between the mean absolute biases were quite large. In these cases, the sample was investigated with respect to outliers. If such outliers could be identified, the Wilcoxon-test was performed, since this test is more powerful under these circumstances (Siegel and Castellan 1988).

Selecting the most appropriate model

After deciding which of the estimators for Model Mt and Mh are preferable, we compare the different models as to which sources of variation are most important in an inspection context. For this comparison we formulated two ordered hypotheses for the number of inspectors and the number of defects. As noted earlier, the four models that we evaluate account for possible sources of variation. It can be expected that models considering more sources of variation will have a better accuracy. This means that Model M0 is expected to perform worst, and Model Mth is expected to perform best. Therefore, an ordered hypothesis is that adding more sources of variation will lead to improved estimates. This hypothesis can be split into two ordered sub-hypotheses: i) Mth is better than Mt which is better than M0, and ii) Mth is better than Mh which is better than M0.

Each of these ordered hypothesis is evaluated separately. For each group of ordered hypotheses (e.g., M0<Mh<Mth) at least two comparisons are made, and sometimes four (e.g., when two estimators for Model Mh are considered). If we assume for each comparison an alpha-level of 0.1 then, considering four comparisons, there is a probability of 0.344 that at least one hypothesis would be rejected even if it is true (Rice, 1987). This is larger than the acceptance level of incorrectly rejecting the true null hypotheses of 0.1. Therefore, the difference between the comparisonwise error (in the example above 0.1) and the experimentwise error (in the example above 0.344) has to be considered. The comparisonwise error is defined as the probability of making a Type I error on a particular comparison (e.g., M0(MLE) vs. Mh(JE)), while the experimentwise error is defined as the probability of making at least one Type I error when conducting a set of pairwise comparisons (e.g., all comparisons belonging to the ordered hypotheses M0<Mh<Mth) (Zwick, 1986). In (Zwick, 1986) it is recommended that if the conclusions depend on the simultaneous correctness of the set of inferences, to control the experimentwise error rate, and if researchers are concerned about the correctness of individual comparisons, to control the comparisonwise error rate.

When comparing the respective influence of various sources of variations in the capture-recapture models investigated, we depend on the simultaneous correctness of a set of inferences. Consider the ordered hypothesis M0<Mh<Mth. In order to draw the conclusion, that heterogeneity improves over Model M0 and is not improved by time response, a specific pattern must be present in the test results. Namely, for *both* h-type estimators the comparison with M0(MLE) must be significant, while for both comparisons with Mth(Ch) the tests must be non-significant. Therefore, controlling the experimentwise error rate is appropriate.

Thus, we select an experimentwise alpha = 0.1 as the significance threshold in order to retain sufficient statistical power, i.e., a high probability (at least 80%) of identifying a difference between estimators when it there is one. Since the direction of the effect in the pairwise comparisons is hypothesized (i.e. more sources of variation improve the estimates), a one-tailed paired t-test is used for these. In order to determine the comparisonwise alpha values, we employ the Dunn-Bonferroni Method as a multiple comparison procedure, which has advantages over other multiple comparison methods when one-sided tests are to be performed (Zwick, 1986). The usage of Dunn-Bonferroni results in a comparisonwise alpha value of 0.05 (comparisonwise significance threshold) for all ordered hypotheses with two comparisons and an alpha value of 0.025 for the ordered hypothesis with four comparisons.

4. Results

In the following section we investigate the data obtained from the combined virtual inspections as described in Section 3.3.: In Section 4.1 the bias and variability of the obtained estimates is analyzed.. In Section 4.2 the approach described in Section 3.4 is used to identify the most accurate model(s). Section 4.3 investigates the failure rate for the estimators. Section 4.4 concluded the section with an investigation, whether the bias of the estimates can be improved by means of calibration.

4.1 Evaluation of Bias and Variability

To compare models for a given number of inspectors or defects the bias of the RE as well as its variability are interesting. Both properties can be conveniently displayed on boxplots. In such boxplots it is possible to show the median (here: the bias), as well as the range and interquartile range of a given data set. The boxplots in this paper use the notation

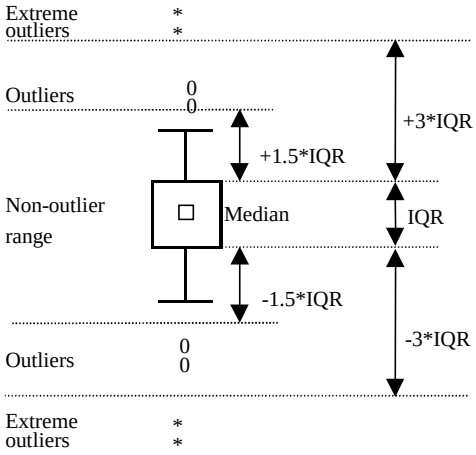


Figure 2: Semantics of boxplots

displayed in Figure 2.

The number of virtual inspections used for determining a boxplot will be displayed by a number below the corresponding boxplot. In some instances, large values were present in the estimations, that would request a large range in single boxplots. In order to display all plots in an equal and reasonable scale, we scaled the boxplots in a way that all relevant information was visible within the plots' limits. Values beyond these limits are denoted as figures beside the upper part of the corresponding plot.

Varying Number of Inspectors

Figure 3 shows the boxplots for all models and for each number of inspectors when considering the RE values for all virtual inspections. Therefore, the median values characterize the bias for each estimator.

The interpretation of these results can be summarized as follows:

(a) There is a general trend towards underestimation. In general, the Chao estimators for Model Mh and Mth show relative errors that are closest to 0. However, they tend to generate rare but extreme outliers especially for low numbers of inspectors. This effect limits their practical use if we have no means to control for these extreme overestimations. One possibility is to use several other models/estimators and compare their fault content estimate with the Chao estimators' estimates. If the latter are much larger (say > 50% larger) then they should be considered

with care. Also, a comparison with some organizational defect content baseline might help detect unrealistic or extreme estimates.

(b) For less than four inspectors and according to our evaluation criterion (-20% < RE < 20%), no estimator yields satisfactory results. Although Mh(Ch) shows a relative error closer to 0, it exhibits large variability with a high maximum value. For models with low RE variability, calibration could be considered an approach to improve the accuracy of an estimate. This can be done, for example, by adding a constant percentage defect overhead to each model's estimate. This calibration procedure will be investigated in Section 4.4.

(c) As the number of inspectors increases, large overestimation tends to decrease monotonically. Thus, we expect the Chao's Estimators to provide satisfactory results (RE ≤ 20%) for more than six inspectors.

(d) For two inspectors, the estimators Mh(JE) and Mt(MLE) show the lowest RE variability compared to the other estimators. Though they usually underestimate, they might be good candidates for calibration. The estimator with the smallest median RE value is Mh(Ch). However, it shows a large variability. The estimator Mth(Ch) does not provide an estimate. This is due to the fact that the estimator has a (k-2) term in one of its denominators where k is the number of inspectors. Surprisingly, Mt and Mh performed even worse than M0, the simplest model.

(e) For three inspectors, the Jackknife Estimator for Mh performed better in terms of RE bias and variability. However, the median RE is still large, (i.e. -27%). Therefore, without calibration, this estimator might not be usable in practice for three inspectors.

(f) For four and five inspectors, the Mh(JE) shows a low median RE. However, it has a relatively large RE variability. Although yielding a low median RE, Chao estimators show very large maximum values. The MLE for Model Mt yields a lower RE variability, but tends to underestimate significantly more than Mh. Estimators Mh(Ch) and Mth(Ch) seem to be the least likely to lead to underestimation.

Like for 4 and 5 inspectors, the Chao estimators for Mt (Mt(Ch)) and Mth (Mth(Ch)) show the best results for six inspectors. Mth(Ch) shows good results in terms of median RE and variability but has a large maximum RE value. Mt(Ch) has a smaller maximum but shows poorer values for median RE and RE variability. The MLE for Model Mt and M0 underestimates to a great extent. Yet, they show the best behavior in terms of RE variability. Therefore, they might be possible candidates for calibration.

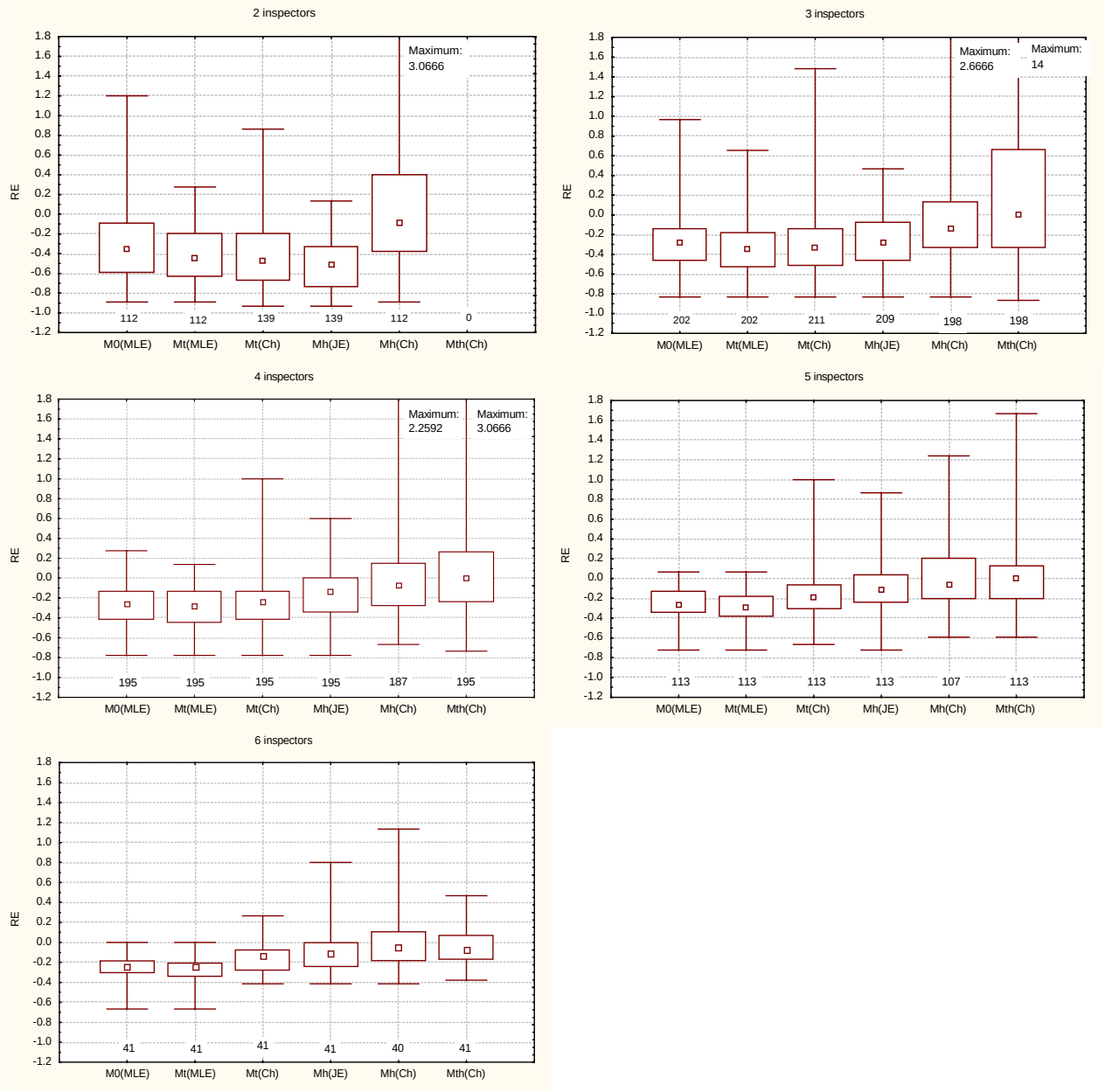


Figure 3: Accuracy for varying numbers of inspectors
(figures below boxplots denote number of virtual inspections used)

Varying Number of Defects

Similarly to Figure 3 for a varying number of inspectors, Figure 4 shows the RE bias and variability for various selected numbers of defects, assuming four and six inspectors.

The interpretation of these results shows that

(a) The larger the number of defects in the documents, the lesser the tendency for large outlying estimates.

(b) With an increasing number of defects, the median RE does not seem to be much affected.

(c) Estimators considering heterogeneity (Models Mh and Mth) show better median RE values than the estimator for Model M0. For six inspectors this behavior is more obvious than for four inspectors.

(d) The two estimators for Model Mt are not better in terms of the median RE than the estimator for Model M0.

(e) Among the estimators considering heterogeneity, Mh(JE) is less preferable in terms of the median RE than the Chao Estimators. Between Mth(Ch) and Mh(Ch) there is no large difference in terms of median RE.

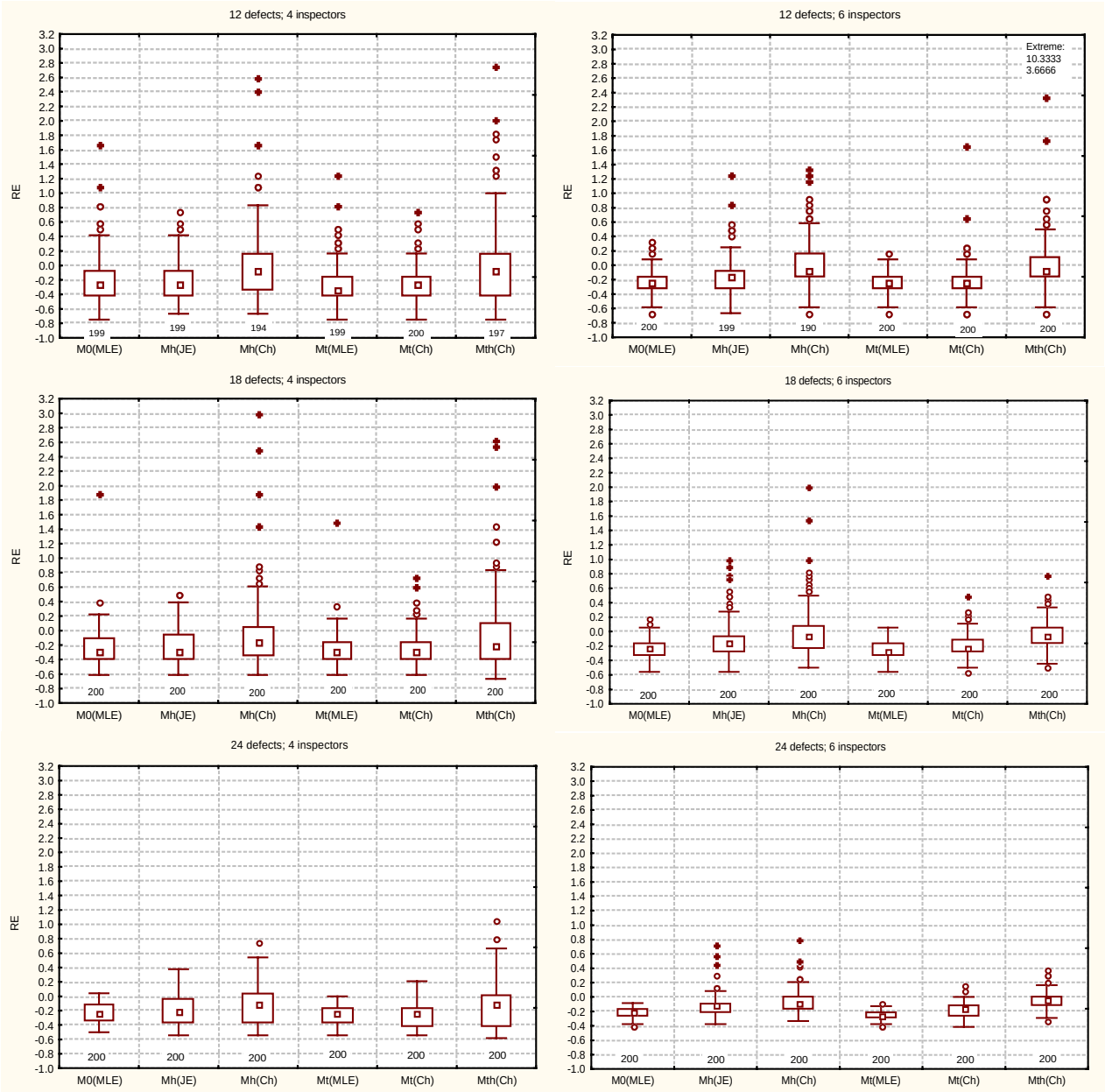


Figure 4: Accuracy for varying number of defects
(figures below boxplots denote number of virtual inspections used)

4.2 Selection of the Best Model

This section follows the selection procedure described in Section 3.4. First, the conditions under which capture-recapture estimates are too inaccurate should be clearly identified. Next, statistical testing is performed in order to compare various estimators and assess how statistically significant their differences are.

Determining thresholds for the number of inspectors and defects

Following the selection procedure in Section 3.4, we want to determine the minimum numbers of defects and inspectors under which models do not perform well. For this purpose, Figure 5 shows the bias for each estimator as a function of the number of inspectors. Furthermore Figure 6 shows the bias as a function of the number of actual defects (for four and six inspectors, only for generic documents since these documents have the largest number of defects). This bias is obtained by taking the median relative error from all inspector or defect combinations and across all documents.

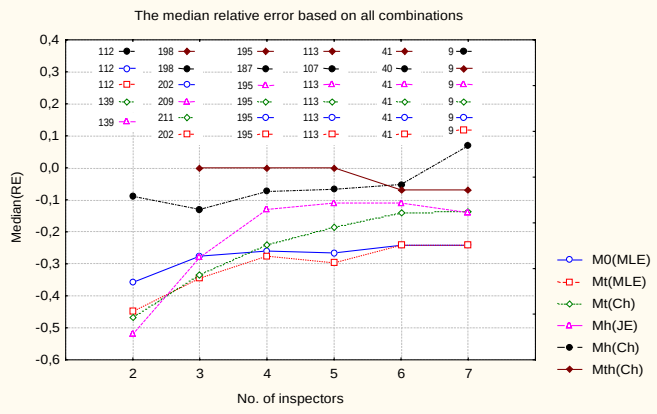


Figure 5: The bias of the estimates as a function of the number of inspectors

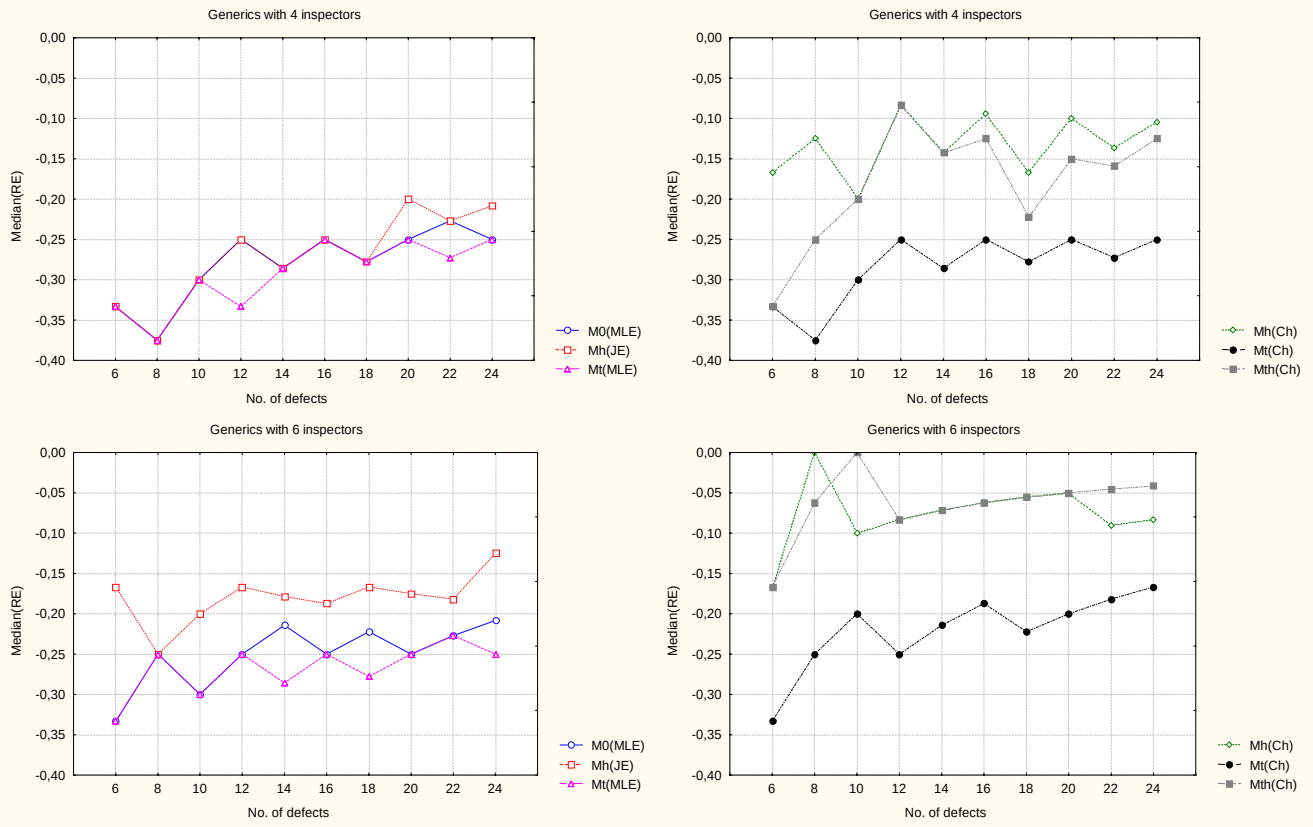


Figure 6: The bias of the estimates as a function of the number of defects

Number of Inspectors

The median RE as a function of the number of inspectors is shown in Figure 5 for the various models. The values above the graph indicate the number of combinations that were generated. It shows that, for most models and over all documents, the median RE decreases fast below 4 inspectors and does not change significantly above that level.

If a median relative error greater than 20% is considered unacceptable, most estimators are not usable for two or three inspectors. Only Mth(Ch) and Mh(Ch) show reasonable values for bias. Unfortunately, Figure 3 reveals that these estimators tend to exhibit occasional large overestimation. Based on these observations, it can be concluded that for inspections with less than four inspectors, capture-recapture models are not very accurate. For four or more inspectors, estimators considering heterogeneity (Mh(JE), Mh(Ch), Mth(Ch)) provide the best results in terms of median relative error (Figure 5). Figure 3 shows that the MLEs for Models M0 and Mt indeed underestimate significantly compared to the h-type models. Yet, they show the lowest RE variability. These estimators' median RE might therefore be significantly improved through calibration.

Number of Defects

In order to visualize the effect of the number of defects on accuracy, only data from the generic documents are used. This is due to the fact that these documents offer the widest range of defect numbers. If the estimates from the NASA documents were also included, this might have distorted the results as for a small number of defects (i.e., 6 to 12) more combinations from both document types (NASA, Generic) would have been used. However, when all documents (i.e., NASA and Generics) are considered for low numbers of defects (i.e., 6 to 12 defects) the results are consistent with those that are presented here for the generic documents.

For each estimator, the median relative error is computed from the approx. 200 virtual inspections and the overall results are shown in Figure 6. In this graph, for each estimator, the median relative error is plotted as a function of the number of defects.

In these plots, the following observations can be made:

- The largest improvements in terms of median relative error can be obtained when changing from 6 defects to 12 defects. This is especially true for Chao's estimators (the plots for these estimators are in Figure 6 on the left side).
- When there are more than 12 defects, there is still an upward trend. However, the improvements are not that large. For example, there is an obvious upward trend for Mth(Ch). However, the payoff of this upward trend is rather low in absolute terms: For twelve defects the median relative error is -0.0833 which means that we underestimate by one defect, for 24 defects the median relative error is -0.0416, which also means that we underestimate by one defect.
- For Mh(JE): When using six inspectors, the median relative error is not strongly affected by the number of defects. From 12 to 22 defects, the accuracy remains nearly constant.

As a result of this investigation we can conclude, that there is no large improvement in median relative error to be expected once there are twelve or more defects in a document. For a small number of defects (i.e., six to twelve) it depends on the estimator whether more actual defects increase the accuracy. For Chao's Estimators big improvements can be made when changing from six to twelve defects. For all other estimators no big improvements can be made when having twelve instead of six defects.

Results of statistical testing

Ideally, statistical testing should be performed for both the number of inspectors and the number of defects. However, here testing is used only for the number of inspectors. For the number of defects, too few documents with a large number of defects were available and this would have resulted in too small a sample size for statistical testing.

For these results, we focus on four and five inspectors only. We did not consider two or three inspectors since we concluded that for less than four inspectors the capture-recapture models become unusable in practice. We did not consider

six inspectors since for six inspectors the number of combinations used to obtain the bias was rather small, and hence this bias is expected to be unstable. This suggests that any conclusions drawn from an analysis using six inspectors from our data set would be quite inconclusive.

Selecting one estimator per model

For four inspectors, there was no difference between the two estimators for Model Mh (using a t-test). We therefore select the Jackknife Estimator since Chao's Estimator shows cases of larger variability in the sense of large overestimation for four inspectors (see Figure 3). Chao's estimator for Model Mt, however, was better than the MLE (p=0.05) and was selected as a representative estimator for Model Mt.

For five inspectors, there was again no difference (p=0.84) between Chao's Estimator and the Jackknife estimator for Model Mh. The variability of Chao's Estimator for 5 inspectors is comparable to the variability of the Jackknife Estimator and we therefore keep both estimators for Model Mh. Chao's Estimator could be selected for Model Mt since it shows a smaller bias than the MLE (p=0.02). However, its variability is larger (Figure 3) and, in practice, selecting one estimator or another may come down to a tradeoff between bias and variability.

Selecting the most appropriate model

The results used for the following discussion are shown in Figure 7 and Figure 8. In these diagrams the nodes are the models and their estimators, and the edges indicate a comparison using the t-test. The p values for each comparison are also given. The figures below each estimator denote the mean of the absolute bias values over the 8 documents.

Figure 7**:** Test Results for 4 Inspectors

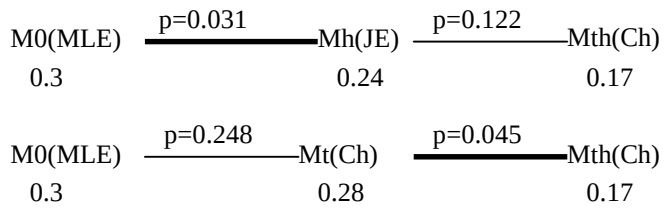
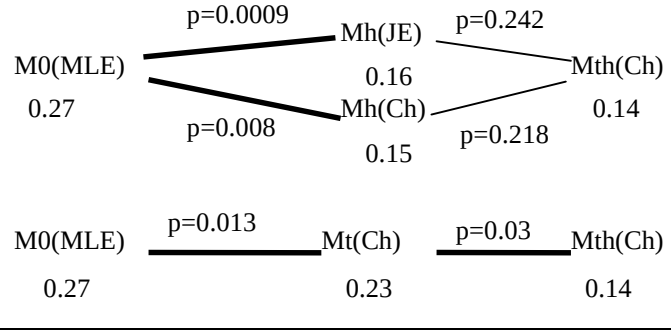


Figure 8**:** Test Results for 5 Inspectors



The comparisons of different models for four inspectors are shown in Figure 7. All means in that figure are in the expected direction (i.e., the models that account for more sources of variation tend to have lower mean absolute bias). Therefore, the only question is whether these differences are statistically significant.

As can be seen, the heterogeneity (Mh(JE)) source of variation improves the mean absolute bias over M0. However, adding time response to Mh to become Mth does not significantly improve the estimates. On the other hand, Model Mt

(with Mt(Ch)) does not perform better than Model M0, while Mth performs better than Model Mt, which can be interpreted as if adding heterogeneity to time response improves the estimates. Overall, this indicates that considering heterogeneity results in improved estimates whereas time response does not. Therefore, we can recommend Model Mh with the Jackknife Estimator for four inspectors.

For five inspectors (see Figure 8) the results indicate that both of the heterogeneity (Mh(JE) and Mh(Ch)) and the time response (Mt(Ch)) sources improve over the basic Model M0. However, taking both sources of variation into account does not improve over the h-type models, but improves over Model Mt. This indicates that, i) each source of variation will lead to improved estimates but that the effect of time response is rather low compared to the effect of heterogeneity, and ii) that the time response source of variation does not bring much more beyond heterogeneity. For five inspectors, we can therefore recommend Model Mh with the Jackknife or the Chao Estimator. For five inspectors, Mth(Ch) shows some large variability in the sense of overestimation (see Figure 3) and is therefore not recommended for five inspectors.

Overall, the pattern of results for four and five inspectors indicates that inclusion of the time response source of variation does not substantially improve the absolute estimation bias. It is interesting to note that, when looking at Figure 6, these results seem consistent for varying numbers of defects.

The results we obtained, however, are based on data coming from a very mature environment (NASA/GSFC) where differences in ability across inspectors (accounted for by Mt-type of models) did not appear to be very relevant. If current practice is to select experienced professionals for performing inspections, then the risk of the above assumption being wrong would be minimal. However, in cases where inspections are used for training junior engineers (for example, this is one of the cited benefits in Doolan (1992)), the above assumption would not hold.

As a general recommendation then, under the condition that experienced inspectors are involved, one ought to use Model Mh with the Jackknife estimator although the Chao Estimator might also be helpful for five and more inspectors, where risks for extreme overestimation are lower.

For the situation that a mixture of experienced and inexperienced inspectors are involved in the inspection, it should be investigated in the future whether models incorporating time response would be more beneficial than in this study.

4.3 Failure Rate

A general observation is that all estimators fail to provide estimates more often for a low number of inspectors. Thus, apart from the poor accuracy, the high failure rate also limits the use of capture-recapture models for a low number of inspectors.

Presenting the results of all estimators and all combinations are beyond the scope of this paper. Thus we concentrate on the Jackknife and Chao estimators for Model Mh as these were the recommended model and estimators. Failures of the Jackknife estimator may occur for a low number of inspectors and defects when there is no overlap in defects detected amongst the inspectors. However, during the simulations for four or more inspectors, this never occurred since the likelihood of no overlap is small (see Table 7). During the simulations for six to 12 defects only 1.1% of all estimations failed. This supports further the utility of the Model Mh when using the Jackknife Estimator.

Mh(Ch) for more than three inspectors failed approximately 3.9% of the time (across all documents) when simulating different numbers of inspectors. However, more failures occurred for small numbers of total defects (less than 12 defects): 20.2% on average. This estimator fails when there are no defects that have been detected by exactly two inspectors. This is more likely to occur if there are few inspectors, (e.g., two inspectors with no overlapping defects), or when there are many inspectors and that all defects are detected by three or more inspectors. Given that we found no differences in absolute bias between the Jackknife estimator and Chao's, the Jackknife estimator is preferable since its failure rate has been found to be lower.

	All defects		4 Inspectors	
Estimator	< 4 inspectors	>= 4 inspectors	<12 defects	>= 12 defects
Mh(JE)	0.5%	0%	1.1%	0%
Mh(Ch)	12.8%	3.9%	20.2%	14%
Mth(Ch)	6.2%	0%	15.7%	6.5%

Table 5: Failure rates as a function of the number of inspectors and defects

4.4 Can Calibration Be Used to Improve the Estimates?

In the preceding sections it was concluded that for a small number of inspectors (two or three) the estimates are too inaccurate to be practically usable. This is a severe limitation to the practical application of capture-recapture models since inspections are often performed with a small number of inspectors. Therefore the utility of calibration for improving the estimates should be investigated. An intuitive calibration approach is to calculate the relative error based on on historical data and adjust future predictions by compensating for this error. However, we will show that the intuitive approach, although very simple, has fundamental problems.

In the following, we first present the calibration approach and then discuss its theoretical limitations. Finally, we perform a rigorous, empirical evaluation of it.

4.4.1 Calibrating Estimates

One solution for improving the accuracy of an estimate is to take the uncalibrated defect estimate $\hat{N}$ and compute a calibrated estimate $\tilde{N}$ by multiplying a constant factor to the initial estimate:

$$\tilde{N} = k\hat{N} \qquad \text{Equation 3}$$

Ideally, the factor *k* should be chosen in a way that $\tilde{N}$ equals N. The problem is now to devise a method to estimate this parameter. An initial or uncalibrated estimate $\hat{N}$ can be obtained by applying an estimator such as the MLE for Model Mt. This uncalibrated estimate can be expected to be biased with a relative error RE:

$$RE(\hat{N}) = \frac{\hat{N} - N}{N} \qquad \text{Equation 4}$$

where N equals the actual number of defects and $\hat{N}$ equals the estimated number of defects. This equation can be changed to:

$$N = \left(\frac{1}{1 + RE(\hat{N})}\right) \times \hat{N} \qquad \text{Equation 5}$$

This equation simply means that, if the relative error of an estimate is known, the actual value can be calculated by means of the relative error and the estimate. In our context, however, the actual relative error (RE) of the estimate is unknown. Therefore, we must use an estimate of RE and in this case we take the bias of the estimator, i.e., the median relative error. With that, we obtain

$$\tilde{N} = \left(\frac{1}{1 + med(RE(\hat{N}))}\right) \times \hat{N} \qquad \text{Equation 6}$$

where $\tilde{N}$ is the calibrated estimate.

4.4.2 Problems of the Simple Calibration Approach

The objective of calibration is to obtain a calibrated estimate $\tilde{N}$ which is equal or near to the true value N. Thus, the relative error of the calibrated estimate should be equal or near zero, regardless of the initial estimate's relative error. In other words, the relative error of the calibrated estimates should fulfill:

$$RE(\tilde{N}) \approx 0 \qquad \text{Equation 7}$$

When considering the simple calibration approach presented above, a linear relationship between the RE's of the calibrated estimates and the RE's of the uncalibrated estimates can be derived:

$$RE(\tilde{N}) = \frac{\tilde{N} - N}{N} = \frac{k\hat{N} - N}{N}$$

$$<=> \qquad RE(\tilde{N}) = kRE(\hat{N}) + (k-1) \qquad \text{Equation 8}$$

Thus, there is a linear relationship between the relative error of the original estimate and the calibrated estimate. Due to this relationship, the relationship between both estimates' variances can be expressed as follows:

$$Var(RE(\tilde{N})) = Var(kRE(\hat{N}) + (k-1)) = k^2 Var(RE(\hat{N})) \qquad \text{Equation 9}$$

Based upon these two relationships, the following results can be expected from the simple calibration method:

- Equation 8: Uncalibrated estimates with a positive relative error or a relative error of about zero will get worse.
- Equation 8: Uncalibrated estimates with a large negative relative error will show a reduced bias.
- Equation 9: The variability of the calibrated estimate will increase quadratically with k (note that we expect k>1, since $\hat{N}$ is generally supposed to underestimate). Therefore, the worse the underestimation, the larger the increase in the $RE(\tilde{N})$ variance when calibrating. In other words, calibration creates a major problem when it is the most needed! This is illustrated in Figure 9.

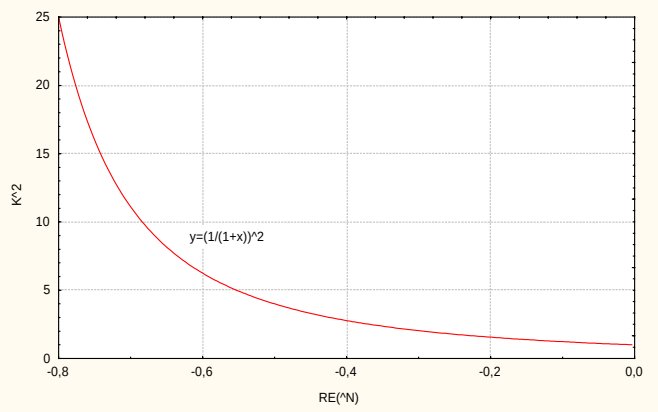


Figure 9: Relationship between bias of uncalibrated estimator and variability of calibrated estimator

4.4.3 Evaluating Calibration

To assess the criticality of the identified calibration problems in practical situations, we performed a n-fold cross-validation. For this, one needs *n* documents. From these n documents n-1 documents are used in order to calibrate estimates on the remaining document. This way, calibration can be validated for each document in a realistic manner. In this case, the median RE for calibrating one document was determined from the remaining n-1 documents.

All combinations of inspectors as described in Section 3.3 were used. For calibrating the combinations of a given document d, the following procedure was used: The median RE used for calibrating document d was obtained by taking the median relative error from all combinations of all documents except d. Using Equation 5, all combinations for document d were calibrated. For each of these calibrated estimates, the resulting relative error was calculated.

Estimators used for calibration are the MLE for Model M0 and Mt and the Jackknife Estimator for Model Mh, since these showed a low RE variability and were, therefore, good a priori candidates for calibration. In the following these calibrated estimators will be denoted as MtCAL, M0CAL, and MhCAL, respectively.

4.4.4 Calibration Results

Tables 8 to 9 show, for small numbers of inspectors (two to three) and each calibrated estimator, the average calibration statistics for all documents. The median RE across all combinations of all documents is shown in the second column. The third column shows the average values of k^2 across documents, where k^2 was estimated based on the median RE of each document. The average variance of the uncalibrated and calibrated estimates are then shown in the fourth and fifth columns. Finally, the last column shows the average increase in variance through calibration, which can be seen as an empirical estimate of k^2. We can see that the change in variance predicted by equation 9 is confirmed by the data. For example, for two inspectors and Mh(JE) calibration increases the variance by a factor of 3.7021, which is close to the predicted factor of 3.6853. The increase in variance is larger for estimators with a lower cross-documents median RE. For example, for two inspectors, Mh(JE) has a lower median RE (-0.5172) than M0(MLE) (-0.3575) and therefore the increase in variance is larger (k^2 decreases from 3.7021 to 2.248). In addition, we can see that the increase in variance is worse for smaller numbers of inspectors, since the average bias of the original estimate is less in those cases. The question is now whether this kind of calibration can be used in practice to obtain improved estimates, although it results in larger estimation variances.

Estimator	Med (RE)	Mean(k^2)	Mean Variance (uncalibrated)	Mean Variance (calibrated)	Mean (calVar/uncalVar)
M0(MLE)	-0.3575	2.1998	0.1284	0.2672	2.2480
Mt(MLE)	-0.4482	3.0279	0.0500	0.1497	3.0502
Mh(JE)	-0.5172	3.6853	0.0320	0.1227	3.7021

Table 6: Calibration results for 2 inspectors

Estimator	Med (RE)	Mean(k^2)	Mean Variance (uncalibrated)	Mean variance (calibrated)	Mean (calVar/uncalVar)
M0(MLE)	-0.2758	1.8101	0.0648	0.1194	1.8143
Mt(MLE)	-0.3448	2.1689	0.0405	0.0875	2.2577
Mh(JE)	-0.2758	1.7424	0.0381	0.0694	1.7722

Table 7: Calibration results for 3 inspectors

Boxplots for each of the calibrated and uncalibrated estimates are plotted. These boxplots are shown in Figure 10. In this figure we can assess the overall impact of calibrated estimates in comparison with uncalibrated estimates for two and three inspectors. This range of inspectors was chosen because it was previously concluded that uncalibrated estimates are too inaccurate for small inspection teams.

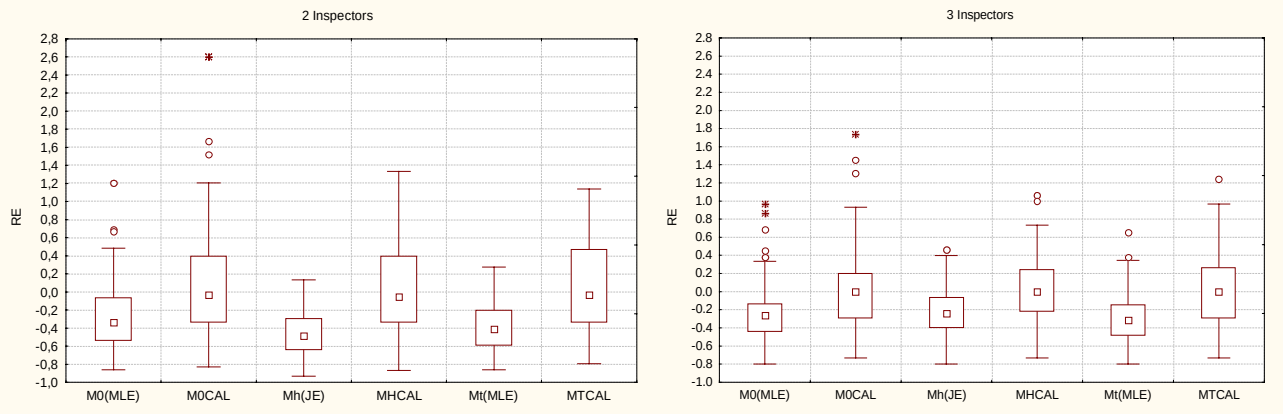


Figure 10: Overview of calibration results

From these plots the following observations can be made:

- As expected, overall, calibration improves the median relative error.
- In many cases the median relative errors are very close to zero. Over- and underestimation occur almost in equal proportions.
- Calibration of Model M0's MLE and Model Mh's Jackknife works better in terms of interquartile range and median relative error than Model Mt's MLE.
- As predicted, calibrated estimators have a higher variability than their corresponding uncalibrated estimators.
- For two inspectors the calibrated estimates have a large variability. The 25% and 75% percentiles (delimiting the interquartile range) are beyond the +/-20% relative error interval.
- For three inspectors and using M0CAL and MhCAL, more than 50% of all estimates are within +/-20%.

As a general conclusion, it can be said that calibration does not work for two inspectors, since the variability of the calibrated estimates is likely to be too large for the models to be practically useful. For three inspectors, calibration seems to yield satisfactory results: the median relative error is close to zero, the interquartile range lies within +/-20%.

However, these are only general results since we have only looked at the overall effect of calibration, neglecting the differences between single documents. This is important since we want to improve the estimates for inspections of *one particular* document by means of calibration. Thus, in order to decide whether one document can really be calibrated using the median relative error of the remaining documents, the results for single documents should be investigated.

In Figure 11 the same data is used as in Figure 10 but shows the results for single documents. For each document, the effect of calibration is shown for the MLEs for Model M0 and Mt and the Jackknife Estimator for Model Mh.

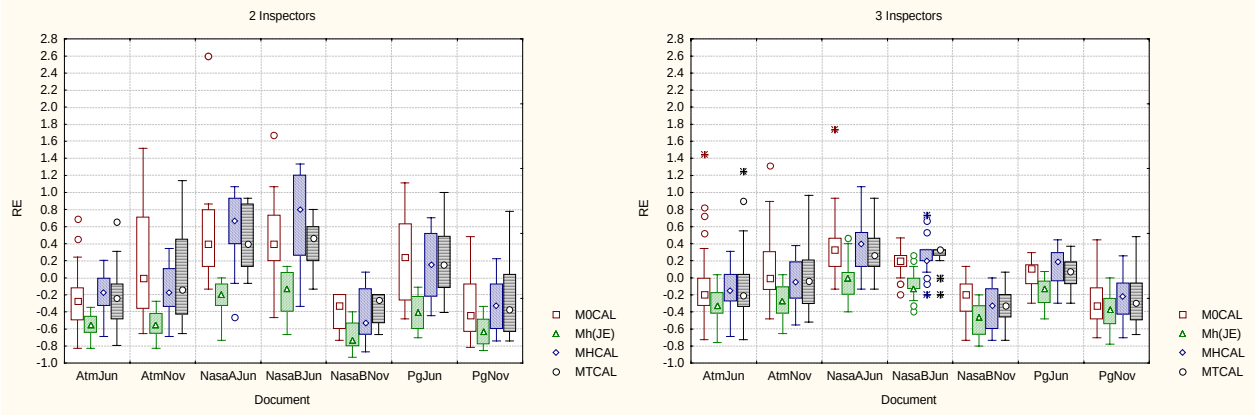


Figure 11: Overview of calibration results per document

When looking at the documents individually, the following two observations can be made. First, when the uncalibrated estimate underestimates severely, the calibrated estimate still underestimates significantly (e.g., the bias of the calibrated estimates is outside the +/–20% interval). Second, when the uncalibrated estimate shows no or moderate underestimation, the calibrated estimate will show significant overestimation. Thus, although in Figure 10 improvements seemed substantial due to grouping effects, a more detailed look at the results shows that calibration is not beneficial for two or three inspectors.

With respect to the evaluation approach for calibration, this result shows that it is important to look at the impact on documents individually when evaluating calibration.

It is also worth noticing the difference between the MLEs for Model M0 and Model Mt. Otis et. al. (1978) indicate that little can be gained from using M0 instead of Mt since M0 is a special case of Model Mt. Our results show, that there is usually little difference between M0 and Mt. If there is a difference, M0 shows the better values in terms of median relative error.

Based on the results above, we can conclude that the kind of calibration proposed here does not work. This is due to two facts: First, the variability of estimates increases significantly, especially for low numbers of inspectors, and second, the median relative error for single documents has a linear relationship with the relative error of the uncalibrated estimates, which makes it unpractical. As a result of this, the median relative error does not often lie within +/- 20%.

4.5 Comparison with Existing Findings in Software Engineering

Eick et al. (1993) provide a recommendation regarding which model works best in which context. They have used Mh and Mt but did not compare them. Vander Wiel and Votta (1993) came to the conclusion that the MLE for Model Mt generally underestimates. The results in this paper do confirm this. However, we could not confirm that the Jackknife estimator for Model Mh tends to exhibit severe overestimation as Vander Wiel and Votta reported. This may be explained by the fact that the violation of the assumptions are more severe for a low number of defects and actual inspection data than a large number of defects and data from simulations.

Wohlin et. al. assessed the MLE for Model Mt in (Wohlin et. al.,1995) in their experiment on textual documents. Consistent with our and Vander Wiel and Vottas (1993) results they found this estimator to underestimate. In (Runeson and Wohlin, 1998) they assessed the MLE for Model Mt with data from their experiments on C-code inspections. They found that in this experiment the MLE constantly produced overestimates, a fact they attributed to the fact that nearly 100% of the defects were already found in the inspection.

4.6 Comparison with Existing Findings in Biology

All the estimators considered here have been analyzed by means of simulations for application in biology (Otis et. al. 1978; Chao et. al. 1992; Chao 1989: Chao 1987). Although the simulated conditions do not fully match the conditions present for inspections, results from biology can further support several of the results shown in this paper.

The major difference in the simulated conditions between biology and inspections is the size of the samples. In our evaluation we considered documents with 15 to 29 defects. Simulations performed in biology usually consider 100 to 400 animals, which are significantly larger sample sizes than what can be inspected in an inspection context. However, we believe it is still worth identifying common findings across all existing studies, so that we can gain more confidence in the external validity of our results.

In order to assess the robustness of an estimator against heterogeneity or time response, the following approach is taken in biology: A data matrix is simulated using catching probabilities fulfilling the assumptions of Model Mh or Model Mt. This matrix is used to obtain an estimate. For example, to assess the impact heterogeneity has on Mt(MLE), data fulfilling the requirements for Model Mh is created and estimated using Mt(MLE). Using a large number of simulations provides enough information to assess the impact of sources of variation on the estimation bias.

Following the procedure above, the results reported in biology are:

- In the presence of severe heterogeneity, estimators for Model Mt systematically underestimate. This confirms our

observation that estimators for Models Mt and M0, which is a special case of Model Mt, underestimate. Furthermore, in the presence of time response, which is interpreted here as varying inspectors' abilities in the inspection context, Mh(JE) provides good estimates, especially if many animals are caught a relatively large number of times. Thus, in a typical software engineering context, Mh(JE) is more likely to be robust to violations than Mt(MLE) or M0(MLE).

- Mh(JE) is preferable to Mth(Ch) when the number of trapping occasions is small and the capture probabilities are low. Mth(Ch) is only recommended in the presence of severe heterogeneity or if the data are sufficiently abundant. Thus, in the conditions we face (small number of trapping occasions), Mth(Ch) is unlikely to be useful.

These simulation results confirm our findings that Model Mh is more usable than Model Mt and that Model Mth does not improve significantly over Model Mh. Such results further supports of the external validity of our results.

Another important finding is that the general factors having an impact on the estimate are the detection probabilities and the number of inspectors. Thus, in practice, capture-recapture estimates should work best if inspectors are better trained or have more experience (higher detection probabilities) and if the number of inspectors is large enough

5. Discussion and Conclusions

We presented in this paper an evaluation of the relevant, state-of-the-art capture-recapture models and estimators using actual software engineering defect data collected during inspections. Capture-recapture models allow us to estimate the total number of defects in a software artifact. For using capture-recapture models in practice, the performance of various models and estimators must be evaluated taking into account various factors. Below we present an overall summary of our evaluation findings and recommendations:

- *Number of Inspectors*
  Our results indicated that the number of inspectors does have an impact on the relative error (RE), RE variability, and failure rate of capture-recapture models. Specifically, it is suggested that capture-recapture models ought not be used with less than 4 inspectors unless shown to work in a particular environment. However, a recent investigation (Miller, 1998) has shown that some of the models may work reasonably well with 3 inspectors. This may be due to differences in number of defects detected and implies that capture-recapture models should be investigated even in the case of inspections with 3 inspectors. An important issue is whether a high number of inspectors can always be used in practice. Practical problems may arise when trying to use a large number of inspectors, e.g., setting up inspection meetings.

- *Accuracy of Models*
  For 4 and 5 inspectors, we found that Model Mh with the Jackknife estimator was most appropriate based on absolute bias, failure rate, and tendency for extreme overestimation.

- *Sources of Variation*
  Intuitively, one would expect the Model Mth to perform best among the models since it is the most general model. In this study, however, we did not find compelling evidence that models with two sources of variation (i.e., considering variations across defect detection probabilities for defects and inspectors) perform much better than models with one source of variation. Assuming 4 or more inspectors, the models work best when they consider that defects have different probabilities of being detected and when the probability to detect defects is constant for all inspectors (i.e., Mh type models). The median relative error for these models (so called Mh type of models) is between 5% and 16%. There are several possible reasons why the Mth-type of model did not perform best. One reason might be that the differences between the inspectors were not large enough so that the time response assumption did not bring much in addition to the assumption of heterogeneity. Another explanation might be that, in an inspection context, insufficient amounts of data are available for the use of Mth.

- *Relative Error Variability*
  In most of the cases, we have observed that the models underestimate the total number of defects. It may be possible to calibrate the model if the variance of the estimates is not too large. Opportunities for model calibration should be

investigated in the future to alleviate the models' bias. Models Mt and Mh appear to offer more potential for calibration. We demonstrated that an intuitive calibration approach based on the relative error of historical estimates does not work well in practice since it increases the variability of the estimates and does not lead to an improved bias as one would expect. Further effort should be devoted to identify an alternative calibration approach.

- *Failure Rate*
  We observed that, for some models, the failure rate was high for a low number of inspectors. This finding limits their practical usage. In general, several models should be used to prevent failure to obtain an estimate.

However, it is of importance to identify appropriate capture-recapture models in the specific environment where they would be used and for the specific number of inspectors to be used. Alternatively, one may decide on an appropriate number of inspectors based on the models' results. If quality control is really important, it may be worth having more inspectors involved in inspections, even if this means spreading the available effort across a larger number of inspectors. Since the current literature recommends 3 to 5 inspectors for optimal inspection efficiency (Weller, 1993), (Bourgoise, 1996), selecting an appropriate number of inspectors for achieving a satisfactory prediction accuracy does not seem unrealistic.

Ideally, the selection of capture-recapture models should be based on cost. All the other factors described above can be included in a cost model. With the help of a cost model the cost of overestimation (e.g., a re-inspection is unnecessarily performed) and underestimation (e.g., defects slip through inspections) can be simulated and assessed taking into account the expected RE, the RE variability, and the failure rate of a capture-recapture model. However, this requires that data about the cost of inspections and defects be readily available. In this study, since we are making general comparisons of all the models, we implicitly consider underestimation equivalent to overestimation, as far as cost is concerned.

In a specific inspection environment the choice of a model should be dependent on the specific cost of over- and underestimation. The combination of cost models with capture-recapture models would improve evaluations such as those performed in this paper.

Furthermore, it would be informative to further investigate whether calibration for models with low variance can improve their bias.

Acknowledgements

The research presented in this paper was performed in the framework of Bernd Freimut's master's thesis (Diplomarbeit). The thesis was awarded with the Software Engineering Award of the Ernst-Denert-Foundation in Germany.

We would like to thank the participants of the original experiments performed at NASA/GSFC for providing the data used in this study.

We also would like to thank the reviewers of this paper whose comments significantly improved this paper.

References

A. Ackerman, L. Buchwald, and F. Lewski (1989) "Software Inspections: An Effective Verification Process". In IEEE Software, Vol. 6, No. 3, pp. 31-36.

V. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sørumgård, M. V. Zelkowitz (1996) "The Empirical Investigation of Perspective-Based Reading". In *Empirical Software Engineering: An International Journal*, vol. 1, no. 2, pp. 133-164.

S. L. Basin (1973): *Estimation of Software Error Rates via Capture-Recapture Sampling*. Technical Report, Science Applications, Inc.

M. Begon (1979): *Investigating Animal Abundance*. Edward Arnold Publishers.

D. Bisant and J. Lyle (1989) "A Two-Person Inspection Method to Improve Programming Productivity". In *IEEE Transactions on Software Engineering*, vol. 15, no. 10, pp. 1294-1304.

K. V. Bourgeois (1996) “Process Insights from a Large-Scale Software Inspection Data Analysis”. In *Crosstalk. The Journal of Defense Software Engineering*, pp. 17-23.

L. Briand, K. El Emam, O. Laitenberger, and T. Fussbroich (1998) “Using Simulation to Build Inspection Efficiency Benchmarks for Development Projects”. In *Proceedings of the 19th International Conference on Software Engineering*, pp. 340-349.

L. C. Briand, K. El Emam, and B. G. Freimut (1998b), A Comparison and Integration of Capture-Recapture Models and the Detection Profile Method, in Proceedings of *the 9th International Symposium on Software Reliability Engineering*, pp. 32—41.

K. P. Burnham and W.S. Overton (1978) “Estimation of the Size of a Closed Population when Capture Probabilities Vary Among Animals”. In *Biometrika*, vol. 65, pp. 625–633.

A. Chao (1987) “Estimating the Population Size for Capture-Recapture Data with Unequal Catchability”. In *Biometrics*, vol. 43, pp. 783–791.

A. Chao (1989) “Estimating Population Size for Sparse Data in Capture-Recapture Experiments”. In *Biometrics*, vol. 45, pp. 427–438.

A. Chao, S.M. Lee, and S.L Jeng (1992) “Estimation Population Size for Capture-Recapture Data when Capture Probabilities Vary by Time and Individual Animal”. In *Biometrics*, vol. 48, pp. 201–216.

E. Doolan (1992) “Experience with Fagan’s Inspection Method”. In *Software - Practice and Experience*, vol. 22, no. 2, pp. 173-182.

E. Dudewicz (1988) *Modern Mathematical Statistics*. John Wiley & Sons, Inc.

N. Ebrahimi (1997), “On the Statistical Analysis of the Number of Errors Remaining in a Software Design Document after Inspection”, In *IEEE Transactions on Software Engineering*, vol. 26, pp.529--532.

S. Eick, C. Loader, M. Long, L. Votta, and S. Vander Wiel (1992) “Estimating Software Fault Content Before Coding”. In *Proceedings of the 14th International Conference on Software Engineering*, pp. 59–65.

S. Eick, C. Loader, S. Vander Wiel, and L. Votta (1993) “How Many Errors Remain in a Software Design After Inspection?” In *Proceedings of the 25th Symposium on the Interface*. Interface Foundation of North America.

K. El Emam, O. Laitenberger, and T. Harbich (2000) “The application of subjective estimates of effectiveness to controlling software inspections”. To appear in the Journal of Systems and Software.

M. Fagan (1976) “Design and Code Inspections to Reduce Errors in Program Development.” In *IBM Systems Journal*, vol. 15, no. 3, pp. 182-211.

T. Gilb and D. Graham (1993), *Software Inspections*, Addison-Wesley.

IEEE Standards Collection, Software Engineering, Std 830-1993, 1994.

C. Jones (1996) “Software Defect Removal Efficiency”, *IEEE Computer*, vol. 29, No. 4, pp. 94-95.

O. Laitenberger and J.-M. DeBaud (1997), Perspective-based Reading of Code Documents at Robert Bosch GmbH, Information and Software Technology, vol. 39, pp. 781-791.

L. Lapin (1978), *Statistics for Modern Business Decisions*, Second Ed., Hartcourt Brace Jovanovich, Inc.

J. Miller (1998), Estimating the number of remaining defects after inspection, International Software Engineering Network Technical Report ISERN-98-24, University of Strathclyde.

H. Mills (1972) *On the Statistical Validation of Computer Programs*. Technical Report Report FSC-72-6015, IBM Federal Systems Division.

J. Musa, A. Iannino, and K. Okumoto (1987), *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill.

D. Otis, K. Burnham, G. White, and D. Anderson (1978) “Statistical Inference from Capture Data on Closed Animal Populations”. In *Wildlife Monographs*, (62):1–135.

A. A. Porter, H. Siy, A. Mockus, and L. Votta (1998) , Understanding the Sources of Variation in Software Inspections, ACM Transactions on Software Engineering and Methodology, vol. 7, pp. 41-79.

K. Pollock (1991) “Modeling Capture, Recapture, and Removal Statistics for Estimation of Demographic Parameters: Past, Present, and Future”. In *Journal of the American Statistical Association*, 86(413), pp. 225-238.

J. Rice (1987), *Mathematical Statistics and Data Analysis*. Duxbury Press.

P. Runeson and C. Wohlin (1998) “An Experimantal Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections”. Empirical Software Engineering, vol. 3, no. 3, pp. 381-406.

G. Schick and R. Wolverton (1978) “An Analysis of Competing Software Reliability Models”. In *IEEE Transactions on Software Engineering*, vol. 4, no. 2, pp. 104–120.

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

Siegel and Castellan (1988), *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, 1988.

S. Strauss and R. Ebenau (1993), *Software Inspection Process*, McGraw Hill, 1993.

E. Weller (1993) “Lessons from Three Years of Inspection Data.”, *IEEE Software*, vol. 10, no. 5, pp. 38-45.

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

S. Vander Wiel and L. Votta (1993) “Assessing Software Designs using Capture–Recapture Methods”, *IEEE Transactions on Software Engineering*, vol. 19, no. 11, pp. 1045–1054.

C. Wohlin, P. Runeson, and J. Brantestam (1995) “An Experimental Evaluation of Capture-Recapture in Software Inspections”., *Software Testing, Verification and Reliability*, vol. 5, pp. 213–232.

C. Wohlin and P. Runeson (1998), “Defect Content Estimations from Review Data”, Proceedings of the 20th International Conference on Software Engineering, pp. 400-409, .

R. Zwick (1986) “Testing Pairwise Contrasts in One-Way Analysis of Variance Designs”. In *Psyschoneuroendocrinology*, vol. 11, no. 3, pp. 253-276.