Evaluating the Effect of the Number of Naturally Occurring Faults on the Estimates Produced by Capture-Recapture Models

Gursimran Singh Walia
Mississippi State University
gw86@cse.msstate.edu

Jeffrey C. Carver
University of Alabama
carver@cs.ua.edu

Abstract

Project managers can use the capture-recapture models to estimate the number of faults in a software artifact. The capture-recapture estimates are calculated using the number of unique faults and the number of times each fault is found. The accuracy of the estimates is affected by the number of inspectors and the number of faults. Our earlier research investigated the effect that the number of inspectors had on the accuracy of the estimates. In this paper, we investigate the effect of the number of faults on the performance of the estimates using real requirement artifacts. These artifacts have an unknown amount of naturally occurring faults. The results show that while the estimators generally underestimate, they improve as the number of faults increases. The results also show that the capture-recapture estimators can be used to make correct re-inspection decisions.

1. Introduction

Project managers and software developers manage the development process by monitoring the quality of the artifacts developed at each lifecycle stage. In the software engineering community, inspections are widely used to improve the quality of these artifacts, by enabling developers to detect faults early and avoid costly rework later [1]. In practice, however, the evidence suggests that the effectiveness of inspections varies widely [1, 2]. Furthermore, inspections only identify the presence of faults; they cannot certify the absence of faults or provide insight into how many remain post-inspection.

Project managers need objective information to help them decide when enough faults have been found that they can safely stop the inspection process. During a real project, a reliable estimate of the number of faults can aid mangers in determining the need for additional inspections. Among the various approaches available for estimating the number of faults (e.g., fault density, subjective assessment, historical trends, capture-recapture, and curve-fitting), capture-recapture is the most objective and appropriate method [2, 6].

Capture-recapture (CR) is a statistical method that was originally developed by biologists for estimating the size of wildlife populations. CR is used by repeatedly trapping (or capturing) a fixed number of animals, marking them, and releasing them back into the population. If the same animal is trapped during subsequent trapping occasions, it is said to have been recaptured. The size of the population is then estimated using: 1) the total number of unique animals captured across all trapping occasions, and 2) the number of animals that were re-captured. A higher percentage of recaptures indicates a smaller population [8, 14].

Using the same principle, the CR method can be used during the inspection process to estimate the number of faults in an artifact. During an inspection, each inspector finds (or captures) some faults. If the same fault is found by more than one inspector it has been re-captured [2, 4]. The total number of faults is then estimated in a similar manner as in wildlife research, with the animals replaced by faults and the trappings replaced by inspectors. The difference between the estimated total number of faults and the faults already found provides an estimate of how many remain.

While biologists have comprehensively evaluated the CR method and have found it to be useful [8, 14], the use of the CR method for software inspections is relatively new [9]. The majority of the CR studies in software inspections have compared estimates produced by the CR models to the actual number of faults seeded into an artifact. Conversely, there has been little research on the effects of the two main factors used by the CR method for producing its estimates, namely the number of faults and the number of inspectors.

In an earlier paper, we analyzed the effect of the number of inspectors on the accuracy of the CR estimates by varying the number of inspectors while keeping the number of faults constant [12]. In this paper, we examine the effect of the other factor, the *number of faults*, on the accuracy of the CR estimates by varying the number of faults while keeping the number of inspectors constant. In addition, to better understand the impact of real vs. seeded faults, we used fault data from the inspection of real software artifacts that contain natural faults made during their development. We also analyzed the ability of the CR estimators to correctly predict the need for a re-inspection using different numbers of faults found during an inspection. Finally, we discuss how these results can be useful for managing projects.

Section 2 describes the basic principles behind the application of CR method to software inspections. Section 3 discusses the previous studies that provided a motivation for this study. Section 4 describes the study design. The analysis and results are reported in Section 5. Section 6 discusses the threats to validity. Section 7 discusses the relevance of the results to managers. Section 8 provides the conclusions and future work.

2. Use of capture-recapture (CR) for defect estimation in software inspections

The use of the CR method in biology makes certain assumptions that do not always hold for software inspections. The assumptions made by CR method in biology include: 1) a closed population (i.e. no animal can enter or leave), 2) an equal capture probability (i.e. all animals have an equal chance of being captured), and 3) marks are not lost (i.e. an animal that has been captured can be identified) [15]. When using the CR in software inspections, the *closed population* assumption is met (i.e., all inspectors review the same artifact and it is not modified) and the assumption that *marks are not lost* is met (i.e. it can be determined if two people report the same fault). However, because some faults are easier to find than others and because inspectors have different abilities, the *equal capture probability* assumption is not met [2, 9].

To accommodate these different assumptions, four different CR models are built around the two sources of variation: *Inspector Capability* and *Fault Detection Probability*. Table 1 shows the four CR models along with their source(s) of variation. Each CR model in Table 1 has a set of estimators, which use different statistical approaches to produce the estimates. The estimators for each CR model used in this study are shown in Table 2. The mathematical details of
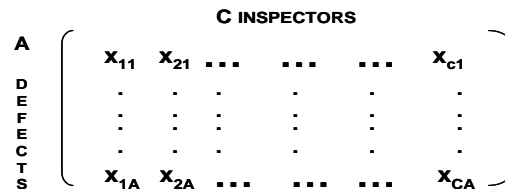
Table 1. Capture-recapture models [2, 10]

Model	Variation Source
M_o	Inspectors have same detection ability, and faults are equally likely of being detected.
M_t	Inspectors differ in fault detection abilities, but faults are equally detectable.
M_h	Inspectors are equally able, but all faults differ in their probability of being found.
M_{th}	Inspectors differ in fault detection ability, and faults differ in detection probability.

Table 2. CR estimators [3, 9, 15]

Models	Estimators
M_o	Unconditional Maximum Likelihood Estimator (M_o-UMLE)
	Conditional Maximum Likelihood Estimator (M_o-CMLE)
	Estimating Equations Estimator (M_o-EE)
M_t	Unconditional Maximum Likelihood Estimator (M_t-UMLE)
	Conditional Maximum Likelihood Estimator (M_t-CMLE)
	Estimating Equations Estimator (M_t-EE)
M_h	Jackknife Estimator (M_h-JK)
	Sample Coverage Estimator (M_h-SC)
	Estimating Equations (M_h-EE)
M_{th}	Sample Coverage Estimator (M_{th}-SC)
	Estimating Equations Estimator (M_{th}-EE)

estimators are beyond the scope of this paper but can be found in provided references. The input data used by all the CR estimators is organized as a matrix with rows that represent faults and columns that represent inspectors as shown in Figure 1. A matrix entry is 1 if the fault is found by the inspector and 0 otherwise.

Figure 1. CR data input matrix

3. Empirical studies of capture-recapture

Most CR research related to software inspections has focused on the basic theory and evaluation of CR models, with very little focus on the influencing factors involved [9]. The theory of CR for software inspections

was introduced by Eick, et al. in an early study on the use of CR models for software inspections by applying them to real defect data from AT&T. They applied the maximum likelihood estimator for the M_t model to estimate the number of faults remaining in requirement and design artifacts. The estimates produced by CR were similar to the subjective opinion of the inspectors. A major result from this study was the recommendation (based on the inspection results) that an artifact should be re-inspected if more than 20% of the total faults remain undetected [4, 5]. This recommendation has been used by all subsequent CR studies.

Weil and Votta used the CR method in the same AT&T environment but added an additional model and estimator - the Jackknife (JK) estimator for the M_h model, and compared it with the M_t-MLE estimator. They found that both estimators produced inaccurate estimates when their assumptions were violated. They also proposed a grouping method to improve these estimators but found that it only improved the accuracy of the M_t-MLE estimator [15].

Briand, et al., reported the first evaluation study that included one or two estimators from each of the four CR models. Using requirement artifacts inspected by NASA professionals, this study investigated the effect that the number of reviewers and the number of faults had on the estimates. The major results from this study showed that the CR models generally underestimate and recommended M_h-JK as the best estimator. The results also showed that the accuracy of the estimators improves with more inspectors and faults, finding that a minimum of four inspectors and six faults are needed to achieve satisfactory estimates. There was no improvement in accuracy beyond four inspectors and six faults [2]. Our current work builds on these early findings. A limitation of Briand, et al.'s, study was that their recommendations were based on only six inspectors using artifacts seeded with fifteen to twenty faults. Therefore, this study builds on their efforts to do a more detailed investigation using artifacts with real fault data and bigger data set. Similarly, Emam, et al., evaluated the CR estimators using only two inspectors and found M_h to be the best CR model. They also advocated the use of subjective opinion with the CR estimates to make decisions on the need for re-inspection during real development [6, 7].

Therefore, most of the CR studies have utilized relatively small data sets. For that reason, we decided to investigate these issues with a larger data set. First, we conducted an empirical study of effect that the number of inspectors had on the accuracy of the CR estimators using data drawn from an inspection performed by 73 Microsoft professionals. The results

from that study contradicted earlier findings that a minimum of four inspectors are needed to achieve satisfactory estimates and provided a detailed analysis that, depending on the estimator, a minimum of fifteen to forty inspectors are needed to obtain estimates within 5% to 20% of the actual fault count, taking into account both accuracy and precision [12].

We performed another study with the goal of evaluating the ability of the CR estimators to estimate the fault count of artifacts containing faults made during their development (as opposed to seeded faults). Each artifact was inspected twice, which allowed the analysis of the CR estimator's ability to decide about the need for re-inspection. The results showed that the estimates after second inspection were more accurate than the estimates after first inspection, and the CR estimates were accurate in determining the need of re-inspection after each inspection cycle [13]

The major results from the analysis of 10 years of research on the use of CR in software inspection as summed up by Petersson, et al. [9], and additional results from Walia, et al. [12, 13], are: a) CR models generally underestimate the fault count; b) M_h-JK is the most accurate but least precise estimator, c) the CR estimates improve with more input data, but there has not been much investigation of the effect of the *number of faults* on the performance of the CR models.

4. Study design

Previous empirical studies of CR in software inspections have evaluated the accuracy of the estimators. The common finding from these evaluation studies is that the CR models generally underestimate the true fault count, but accuracy improves with more input (i.e., more inspectors and more defects). The impact of these two factors on the estimation accuracy is expected to be positively correlated. However, this relation has not been extensively investigated. This study is a follow-up from our earlier study that investigated the impact on estimates when the *number of inspectors* is increased [12]. This study investigates the impact of the *number of faults* on the accuracy of CR estimates by keeping the number of inspectors constant and varying the number of faults.

As mentioned earlier, the data used in most previous CR studies was drawn from the inspection of artifacts with seeded faults, rather than naturally occurring faults. Therefore, this study tries to understand the effect of the number of faults on the CR estimates using real artifacts developed by students in a senior-level capstone software engineering class (i.e. they were created to guide the later implementation of the system)

with naturally occurring defects. The effect of the number of faults on the re-inspection decision is analyzed to gain additional insights into how the CR method can be used on other projects.

4.1 Study goals

The major goal of this study is to understand the effect of the number of faults on the estimates produced by CR models. To achieve this goal, this study focuses on two important research questions:

Question 1: How does the performance of the CR estimators improve as a larger percentage of faults are discovered?

This question focuses on the general trends in the improvement of the performance of estimators as more faults are found. Answering this question provides details into what percentage of faults must be found before the CR models provide satisfactory estimates. Knowledge of these general trends and analyzing the improvement of the fault count estimates in their organizations will help project manager in determining the quality of an artifact under review.

Question 2: How is the re-inspection decision ability of the CR estimators affected by increasing the number of faults?

This question focuses on the estimate of the post-inspection faults relative to the number of faults found during an inspection without knowing the true faults count. The ability of the CR estimators to accurately estimate the number of post-inspection faults is relevant for making re-inspection decisions. Answering this question provides project managers useful insights to make re-inspection decisions in real development.

4.2 Data set

The data was drawn from an earlier inspection study conducted at Mississippi State University (MSU). The goal of the original study was to investigate the impact of errors (mistakes) committed during the development of the requirements document [13]. Only the information relevant to CR analysis is provided here.

4.2.1 Software artifacts and inspectors. Inspection data from two real requirement artifacts is used in this study. The artifacts were developed by sixteen senior-level undergraduate students, majoring in either computer science or software engineering who were enrolled in the Senior Design Course at MSU during the Fall 2005 semester. The sixteen subjects were divided into two 8-person teams that developed the requirement document for their respective system as shown in Table 3. The course required the students to interact with real customers to elicit, and document requirements that they would later implement. Each artifact was then inspected by the same set of developers who created it [13].

4.2.2 Software inspection process. First, all the subjects received training on a fault checklist. Then, each inspector inspected the artifact using that fault checklist and logged any faults identified. Then, the subjects were trained on how to abstract errors from faults, how to classify the errors, and how to use the errors to re-inspect the requirements document. Then, each inspector re-inspected the artifact using the errors. The same inspection process was followed by the subjects in each team, and the artifacts were not modified or corrected between inspections (i.e., the same artifact was re-inspected). The total number of faults found after two inspections in each artifact is shown in the last column in Table 3.

4.3 Evaluation procedure

To understand the impact that the number of faults has on the CR estimates, virtual inspections were created by keeping the team size constant at 8 and varying the number of faults from 1 to 55 for artifact A and from 1 to 105 for artifact B. The data from the original study was organized into an 8 (inspectors) X 55 (defects) matrix for artifact A and an 8 (inspectors) X 105 (defects) matrix for artifact B.

To create virtual inspections the appropriate number of rows (equal to the fault count being studied) were

Table 3. Requirement artifacts and inspectors used in this study

Artifact	Name	Description	# of Pages	# of Requirements	Number of Inspectors	Total Faults
A	Starkville theatre system	Management of ticket sales and seat assignments for the community theatre	27	14	8	55
B	Management of apartment and town properties	Managing apartment and town property, assignment of tenants, rent collection, and locating property by potential renters	42	16	8	105

selected randomly from the total pool of faults. For example, for artifact A, to create the virtual inspection for a fault count of twenty, twenty rows were randomly selected from the 8 X 55 matrix to produce a new 8 X 20 matrix. Similarly, for artifact B, to create the virtual inspection for a fault count of thirty, thirty rows were randomly selected from the original 8 X 105 matrix producing a new 8 X 30 matrix. Using this approach, 10 virtual inspections (i.e. 10 matrices) were created for each fault count. i.e., 10 virtual inspection teams are derived from 8 review teams. The automated tools CAPTURE [15] and CARE-2 [3], were then used to calculate the estimates.

4.4 Evaluation criterion

From the ten estimates produced for each artifact and each fault count, the median value is calculated. The performance of the CR estimators is then evaluated using three metrics: accuracy (bias), precision (variability), and failure rate.

The **accuracy (bias)** is measured as the relative errors (R.E) of an estimate. It is calculated as:

$$R.E = (Estimated\ number\ of\ defects - Actual\ number\ of\ defects) / (Actual\ number\ of\ defects)$$

A R.E of zero means absolute accuracy (zero bias), a positive R.E means an overestimation, and a negative R.E means an underestimation. The accuracy of the CR estimator is measured by calculating the median relative error for each fault count. According to Eick, et al. and Briand, et al., the accuracy of an estimate is considered satisfactory when the R.E is within 20% of the actual value [2, 4, 12, 13].

The **precision** of an estimator is measured by calculating the variability of the R.E. estimates for each fault count. R.E variability around the central tendency i.e. (median value) is measured using the inter-quartile range of the 25th percentile to 75th percentile.

The **failure rate** of an estimator is defined as the number of times an estimator fails to produce any

result. Because each estimator makes different assumptions about the data and they all operate on the same data matrix, some estimators can fail if the data fails to meet some of its basic assumptions.

5. Analysis and results

This section reports the major results relative to the two research questions defined in Section 4.1.

5.1 Effect of fault count on CR estimates

Figures 2 and 3 show the median R.E. for each CR estimator for all fault counts (each line connects the estimates from the same estimator) for artifact A and artifact B respectively. To calculate the RE values, the actual number of faults is assumed to be the total number of faults found after two inspections (i.e., 55 faults for artifact A and 105 faults for artifact B). These figures show that:
a) The estimators failed to produce an estimate when the fault count was less than five.
b) The estimators severely underestimate when the fault count is small,
c) The accuracy of estimators improve with a higher fault count, and
d) Some estimators improve faster (i.e., obtain accurate results with fewer faults) than other estimators.

To compare, the relative performances of the estimators with respect to fault count, we need to determine the cut-off points (minimum fault count) required to obtain an estimate that falls within 20% of the actual value. The cut-off point is one larger than the largest number of faults for which the median estimate falls within 20% (i.e., from that point forward, the R.E. decreases as the fault count increases). For example, for artifact A, for M_O-EE estimator, from 37 to 55 faults the median estimate is always within 20% of the actual value (i.e., 20% of 55 = 44 faults).
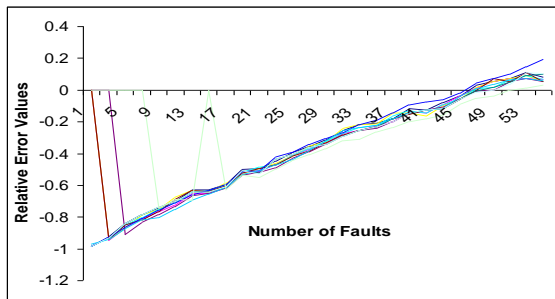

Figure 2. Median relative error values for different fault counts for artifact A
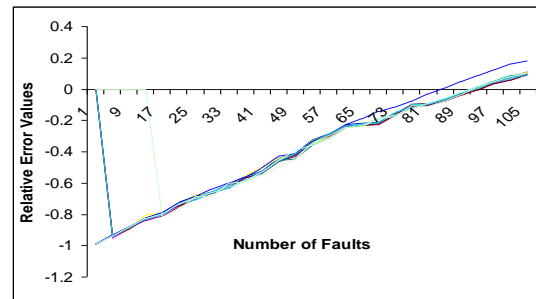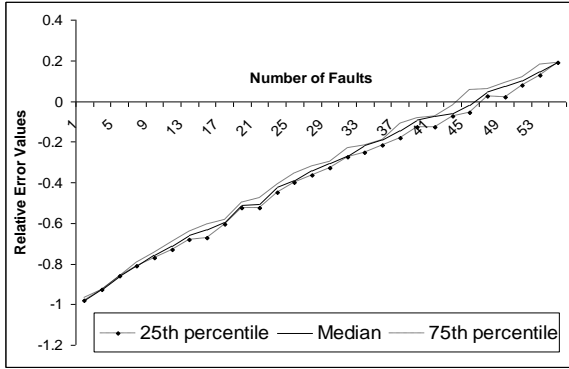

Figure 3. Median relative error values for different fault counts for artifact B

Figure 4. Variability in the estimates for different fault counts for artifact B

Table 4. Percentage of faults required for different levels of estimation accuracy

Artifact A		Estimators	Artifact B	
-10%	**-20%**		**-20%**	**-10%**
76%	67%	M_o-CMLE	68%	73%
76%	67%	M_o-UMLE	69%	77%
78%	66%	M_o-EE	69%	74%
76%	66%	M_t-CMLE	69%	74%
80%	67%	M_t-UMLE	68%	78%
80%	66%	M_t-EE	69%	78%
80%	66%	M_h-SC	64%	75%
69%	62%	M_h-JK	59%	70%
80%	66%	M_h-EE	68%	74%
80%	67%	M_{th}-SC	64%	75%
82%	69%	M_{th}-EE	68%	79%

The R.E. in the estimate is only an indicator of accuracy. In practice, an estimator needs to be both accurate and precise. To understand precision, the variability of the R.E. for each fault count is calculated as the size of the interquartile range (i.e., the spread of the middle 50% of the data). Then, to combine accuracy and precision, three values are calculated for each fault count: a) the median estimate, b) the seventh largest estimate (75th percentile), and c) the third largest estimate (25th percentile). Together b) and c) define the interquartile range. Figure 4 shows this analysis for the M_h- JK estimator on artifact A, with the median R.E. estimate appearing between the upper (75th percentile) and lower bound (25th percentile) on the estimates. Similar graphs were produced for each estimator on each artifact.

The criterion for selecting the minimum fault count will now consider the R.E. at three points (median, 75th percentile, and 25th percentile). The result of this analysis that combines accuracy and precision is shown in Table 4.

Based these results, some general observations can be made:
a) For artifact A, depending on the estimator, inspectors need to find anywhere between 62% and 67% of the total faults for the R.E. estimate to be within 20% of the actual value; and between 69% and 82% of total faults for the R.E. estimate to be within 10% of the actual value,
b) Similarly for artifact B, inspectors need to find anywhere between 59% and 69% of total faults for the R.E. estimate to be within 20% range; and between 70% and 79% of total faults for the R.E. estimate to be within 10% range,
c) The M_h-JK estimator improves the fastest compared with the other estimators (i.e., it requires

fewer faults to achieve an accurate and precise estimate),
d) The variability (precision) is not affected much by the increase in the fault count, and
e) The EE estimators for M_o, M_t, and M_{th} models had a high failure rate (i.e., did not produce an estimate).

Therefore, a significantly large percentage of faults have to be found before the CR models can provide satisfactory estimates. In terms of the relative performance, the Jackknife estimator (JK) is the best estimator. Project managers can perform a similar analysis after each inspection to identify any trend in fault count estimate to gain insights into the quality of an artifact under review. For example, if there is a continuous significant increase in the estimated fault count with increase in the faults found (as opposed to a point after which no improvement is visible), then it is likely that there are substantially more faults remaining in the artifact. The lack of an increase in the estimates indicates that a large percentage of faults have been found.

5.2 Effect of the number of faults on the re-inspection decision of artifacts

The results in Section 5.1 compared the error in the CR estimates relative to the actual fault count (i.e., total faults found after two inspections). However, during real software development, a project manager does not know the actual fault count. Therefore, they have to make a re-inspection decision based only on the number of faults found during an inspection and the estimate of the remaining faults.

This section re-calculates the R.E. values for each fault count, where the actual number of faults is set
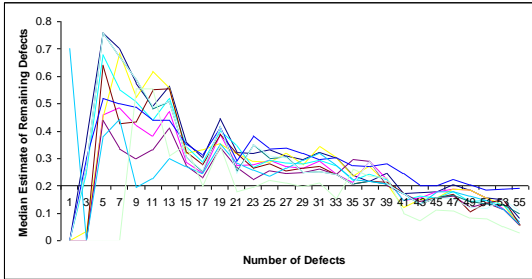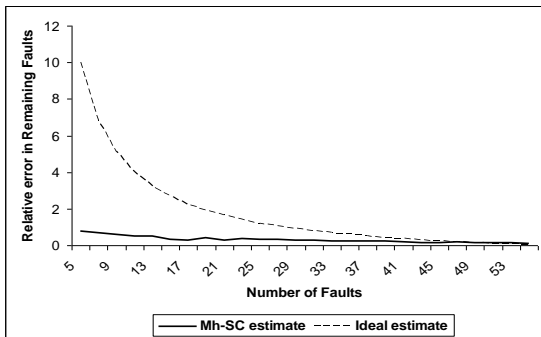
Figure 5. Relative estimate of remaining faults for all defect counts for artifact A
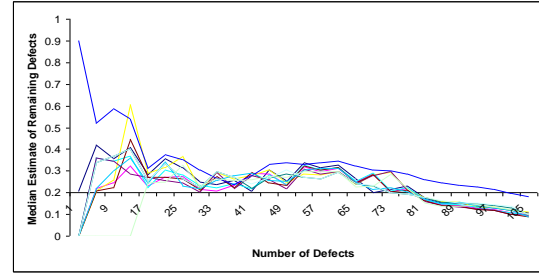

Figure 6. Relative estimate of remaining defects for all fault counts for artifact B

equal to the particular fault count being studied. For example, for artifact A and fault count equal to 37, M_h-JK estimator produced a median estimate of 49 faults. Therefore, the R.E. = (49-37)/37 = 0.32. Meaning that an estimated 32% more faults are remaining. Figure 5 and Figure 6 shows the relative estimates of the remaining faults from all the CR estimators at each fault count for artifacts A and B respectively.

To analyze the effect of the fault count on the re-inspection decision, the estimate of remaining faults is compared against 20% criterion (i.e., if the estimate is greater than 20%, a re-inspection should be performed, otherwise no). The general observations from these figures are:

a) All the estimators (except EE and UMLE) failed for a fault count of less than five. The EE and UMLE estimators failed for fault counts up to 13,

b) For artifact A, the CR estimators (except M_{th}-EE and M_h-JK) suggest the need for re-inspection when the fault count is between 5 and 40. The M_h-JK indicates a need for re-inspection when the fault count is between 3 and 49.

c) For artifact B, the SC estimator for M_h and M_{th} suggest the need for re-inspections when the fault count is less than 76. The remaining estimators (except M_h-JK) indicate the need for re-inspection

when the fault count is less than 86. M_h-JK suggests the need for re-inspection when the fault count is less than 98.

Because the estimator fail when the fault count is less than five, for faults counts greater than five, we evaluated the correctness of the decision to re-inspect artifacts A and B for all the CR estimators. The process for evaluating the correctness is to compare the R.E. in the estimate produced by the CR estimators to the R.E. in the estimate using an ideal estimate (i.e. the CR estimators are perfect). The calculation of these two R.E values is described as:

a) R.E in the remaining faults produced by CR estimators:
    R.E = (estimate – actual) / actual; where **actual** = fault count being analyzed and **estimate** = fault count estimated by CR estimators

b) Ideal R.E in the remaining faults:
    R.E = (estimate – actual) / actual; *where **actual** = same as in a) and **estimate** = 55 for artifact A and 105 for artifact B.

For example, for artifact A, R.E in the estimate produced by the M_h-SC estimator at a fault count of 29 = *(36.25 – 29) /29 = 0.25* i.e., 25%, and the ideal R.E. = (55 – 29) /29 =0.89 i.e., 89%.

Figure 7 shows this analysis graphically for the M_h-

Figure 7. Comparison of estimate and actual relative error for Mh-SC on artifact A
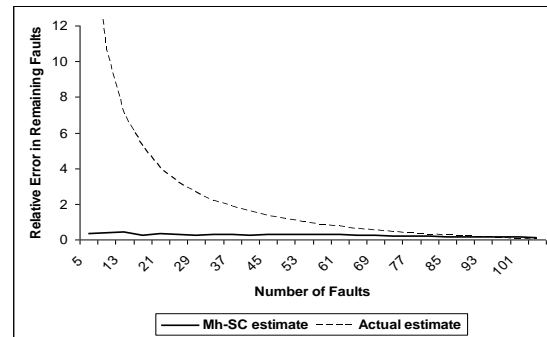
Figure 8. Comparison of estimate and actual relative error for Mh-SC on artifact B

SC estimator at all fault counts on artifact A from five to 55 faults. Similarly, Figure 8 shows the analysis for M_h-SC on artifact B. The solid line represents the R.E in the estimates produced by the M_h-SC estimator and the dotted line represents the ideal R.E. values.

The dotted lines in Figure 7 and Figure 8 touch the 20% axis at fault count of 45 for artifact A and 86 for artifact B (i.e., after this point the estimates are always within 20% of the actual). Comparing the R.E. in the estimates produced by CR estimators vs. the ideal R.E yields following observations:

a) The CR estimators severely underestimate the actual fault count, however, at most fault counts the CR estimators can accurately predict the need for a re-inspection (by predicting that more than 20% of the faults remain).

b) Regarding a correct or incorrect re-inspection decision:
   a. For artifact A, the estimators correctly suggest the need for re-inspection for 5-40 faults, and the JK estimator makes the correct suggestion for 3-45 faults.
   b. For artifact B, estimators correctly suggest the need for re-inspection for 5-76 faults, and JK estimator estimates the need correctly for 3-86 faults.
   c. However, the JK estimator incorrectly suggests the need for re-inspection for 47-49 faults for artifact A and for 87-98 faults for artifact B.

This result is shown in Table 5. The result show that the Jackknife (JK) estimator is able to accurately predict the need for re-inspection for fault counts up to 44 (in artifact A) and 86 (in artifact B). However, it estimates that there are still some faults remaining for fault count up to 49 faults (in artifact A) and 97 (in fault B). One caveat to this analysis is that we assumed the actual fault count to be equal to the number of faults found after two inspections. There might actually be some more faults in the document. Therefore, we suggest Jackknife as the best estimator.

5.3 Use of the results to managers on their software projects

The results in Sections 5.1 and 5.2 provided insights into the: 1) the percentage of faults that have to be found to obtain an accurate CR estimate, and 2) the ability of different CR estimators to correctly predict the need for re-inspection. This section provides some additional insights into how project managers can use these results to manage projects in their organization.

Table 5. Correctness of re-inspection decisions for artifacts A and B

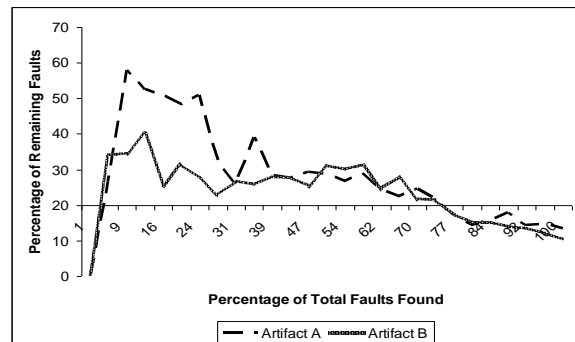| Artifact A		Artifact B	
Incorrect decision	Correct decision	Correct decision	Incorrect decision
Less than 5 faults			Less than 5 faults
41-44 faults	7-41 faults	9-76 faults	77-85 faults
47-49 faults for M_h-JK	3-45 faults for M_h-JK	3-86 faults for M_h-JK	86-97 faults for M_h-JK

In real development the number of faults in an artifact is unknown. Therefore, a project manager has to make re-inspection decisions using only the knowledge of the faults that have been found up to that point. To understand the general trend in the estimates from the CR estimators, Figure 9 plots the percentage of remaining faults for increasing percentages of total faults for artifact A (solid line) and B (dotted line).

The percentage of remaining faults is calculated as:
[(*Estimated number of faults – Number of faults found*) / *Number of faults found*]* 100.

Each line connects the median value of the percentage of remaining faults from all the CR estimators (except M_{th}-EE estimator because of its failure rate). Some general observations from Figure 9 are summed as follows:

a) The estimators fail to produce an estimate when less than only 1% of the faults have been found;

b) The estimators consistently (and accurately) predict the need for a re-inspection (i.e., more than 20% of the faults remain) up to the point when approximately 75% of the total faults have been found (this figure is similar for both artifacts).

Therefore, a project manager should allow a re-

Figure 9. Remaining fauts at different percentage of total faults for both artifacts

inspection as long as the estimators predict the need for one. Each organization can perform a similar analysis to ensure these trends are valid. Understanding the trend in the estimates will help project managers decide on the portion of faults that have been discovered and how long the inspection process should continue. However, a project manager has to make a trade-off between the cost involved in re-inspection and benefits of finding more faults.

6. Threats to validity

Some validity threats were addressed. The artifacts used in this study were real software artifacts that were later used to guide implementation of a real system. The faults were naturally occurring and inserted while developing the artifacts rather than being artificially seeded.

However, there were some threats to validity that were not addressed. First, the actual number of faults in each document is unknown and might be higher than the assumed fault count (i.e., the total number of faults found after two inspections). The effect of this threat on the results is discussed in next section. A second threat was the artifacts used in this study were developed by student teams in a senior-level capstone course and may not be representative of industrial requirement documents. Also, the nature of faults committed by students may differ from faults made by professionals.

7. Discussion of results

This section relates the results from Section 5 to the main research questions posed for this study. Then it provides a discussion of the relevance of the results to software organizations.

7.1. Findings and recommendations

RQ1: How does the relative performance of the CR estimators improve as a larger percentage of faults are discovered?

The increase in the number of faults directly improves the accuracy of estimators with little improvement in precision (or variability). Based on the results from Section 5.1, the CR estimators (except M_{th}-EE) can be used on artifacts that contain five or more faults to produce some estimate. However, the CR estimators require inspectors to find a significantly large percentage of total faults (between 60% to 70%) in order to achieve a satisfactory estimate (i.e., an estimate within 20% of the actual fault count).

Considering the fact that the actual number of faults might be more than assumed for this study, the percentage of faults that have to be found might be even higher. Therefore, the CR estimators severely underestimate the actual fault count when a smaller percentage of the total faults have been found.

In terms of the relative performances of the CR estimators, the JK estimator shows the fastest improvement in accuracy compared with the other estimators. As a result, it requires fewer faults to achieve same level of estimation accuracy.

RQ2: How is the re-inspection decision ability of the CR estimators affected with an increase in the fault count?

Even though the estimator severely underestimates at different fault counts, it correctly predicts the need for re-inspection very consistently. Based on the results from Section 5.2, the CR estimates should not be used for fault counts of less than six for deciding on the need of re-inspections (because of their high rate of failure). In addition, the M_{th}-EE estimator fails to produce an estimate using count of as much as thirteen faults. Therefore, M_{th}-EE estimator is not recommended.

The results showed that the CR estimators helped made the correct re-inspection decision at most (but not all) of the higher fault counts. In addition, the M_h-JK estimator provided accurate re-inspection decision at more fault counts than the other estimators. Therefore, the JK estimator is recommended as the best estimator. Based on the findings from this study, an important recommendation is that a project manager should not trust the actual number of estimate faults remaining. However, he/she can trust the suggestion of need for re-inspection decision based on the 20% threshold.

Table 6 compares the important findings from this study that confirm some findings while contradict some other findings from previous CR studies in software engineering.

7.2. Relevance to software organizations

A project manager needs to decide whether or not to re-inspect an artifact in real time. To accurately use the CR models, it is imperative to know the relative performance of the different CR estimators and their ability to correctly predict the need for re-inspection based only on the faults found in previous inspections. The results in this paper provide insights into the relative performance of the different CR estimators with respect to varying fault counts and how the CR estimates can help manage the inspection process.

Table 6. Comparison of findings

	Previous Studies	Our Study
1	The CR estimators underestimate but improve with more faults and inspectors [2, 9, 14]	<u>CONFIRM</u>. CR estimators produce no estimate for less than 5 faults. We report the percentage of faults that have to be discovered for varying levels of estimation accuracy.
2	No big improvement in accuracy for more than six faults [2]	<u>CONTRADICT</u>. We found linear improvement in the estimates beyond six faults due to the larger fault data used
3	M_h-JK overestimates if the overlap of faults is small [14, 10, 14]	<u>CONTRADICT</u>. M_h-JK can produce satisfactory estimate for as few as 3 faults
4	M_h-JK is the best estimator [2, 8, 10]	<u>CONFIRM</u>. The JK estimator showed the fastest improvement. It is the best estimator
5	The UMLE, EE estimators have high failure rates [9]	<u>CONFIRM</u>.
6	The CR estimators can accurately predict the need for re-inspection [13]	<u>CONFIRM</u>. The CR estimators can be trusted to make a re-inspection decision using the 20% threshold

8. Conclusion and future work

The results in this paper show project managers and how to monitor the quality of artifacts by using CR estimates to understand the percentage of faults that remain in an artifact. This study is by no means a complete investigation of the factors that influence the CR analysis. However, we have investigated the effect of the *number of inspectors* (in an earlier study) and the *number of faults* (in this study) in isolation. Our immediate next efforts will be to understand the combined effect of these two factors by varying both the inspection team size and number of faults together.

9. Acknowledgements

10. References

[1] Ackerman, A., Buchwald, L., and Lewski, F., "Software Inspections: An Effective Verification Process*." IEEE Software*, 1989. **6**(3): 31-36.

[2] Briand, L.C., Emam, K.E., Freimut, B.G., and Laitenberger, O., "A Comprehensive Evaluation of Capture Recapture Models for Estimating Software Defect Content*." IEEE Transactions on Software Engineering*, 2000. **26**(6): 518-539.

[3] Chao, A. and Yeng, H.C., *Program CARE-2 (for Capture-Recapture Part.2), http://chao.stat.nthu.edu.tw*

[4] Eick, S., Loader, C., Long, M., Votta, L., and Weil, S.V. "Estimating Software Fault Content Before Coding*". In Proceedings of the 14th International Conference on Software Engineering*. 1992. Australia.

[5] Eick, S., Loader, C., Weil, S.V., and Votta, L. "How Many Errors Remain in a Software Design after Inspection*". In Proceedings of the 25th Symposium on the Interface*. 1993.

[6] El-Emam, K., Laitenberger, O., and Harbrich, T., "The Application of Subjective Estimates of Effectiveness to Controlling Software Inspections *" Journal of Systems and Software*, 2000. **54**(2): 119-136.

[7] El-Emam, K. and Laitenberger, O., "Evaluating Capture-Recapture Models with Two Inspectors*." IEEE Transactions on Software Engineering*, 2001. **27**(9): 851-864.

[8] Otis, D., Burnham, K., White, G., and Anderson, D., "Statistical Inference from Capture Data on Closed Animal Population*." Wildlife Monograph*, 1978. **64**: 1-135.

[9] Petersson, H., Thelin, T., Runeson, P., Wohlin, C. "Capture-Recapture in Software Inspections after 10 Years Research-Theory, Evaluation, and Application." *The Journal of Systems and Software*, 72(2):249-264, 2004.

[10] Runeson, P. and Wohlin, C., "An Experimental Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections*." Empirical Software Engineering: An International Journal*, 1998. **3**(4): 381-406.

[11] Thelin, T., Petersson, P., and Runeson, P., "Confidence Intervals for Capture-Recapture Estimations in Software Inspections*." Journal of Information and Software Technology*, 2002. **44**(12): 683-702.

[12] Walia, G. S., Carver. J., Nagappan, N. "The Effect of the Number of Inspectors on the Defect Estimates Produced by Capture-Recapture Estimators". In Proceedings of International Conference in Software Engineering, May 10-18, 2008. Leipzig, Germany.

[13] Walia, G. S., Carver. J. "Evaluation of Capture-Recapture Models for Estimating the Abundance of Naturally Occurring Defects." To Appear in the Proceedings of the 2nd International Symposium of Empirical Software Engineering and Measurement, October 9-10, 2008. in Kaiserslauten, Germany.

[14] Weil, S.V. and Votta, L., "Assessing Software Designs Using Capture-Recapture Methoda*." IEEE Transactions on Software Engineering*, 1993. **19**(11): 1045-1054.

[15] White, G.C., Anderson, D.R., Burnham, K.P., and Otis, D.I., *Capture-Recapture and Removal Methods for Sampling Closed Populations, Los Alomos National Laboratory, 1982.*