C. Wohlin and P. Runeson, "Defect Estimations from Review Data", Proceedings 20th International Conference on Software Engineering, pp. 400-409, Kyoto, Japan, April 1998.

Defect Content Estimations from Review Data

Claes Wohlin
Dept. of Communication Systems
Lund University
Box 118
S-221 00 Lund, Sweden
+46-46-222 3329
claesw@tts.lth.se

Per Runeson
Dept. of Communication Systems
Lund University
Box 118
S-221 00 Lund, Sweden
+46-46-222 9325
perr@tts.lth.se

ABSTRACT
Reviews are essential for defect detection and they provide an opportunity to control the software development process. This paper focuses upon methods for estimating the defect content after a review and hence to provide support for process control. Two new estimation methods are introduced as the assumptions of the existing statistical methods are not fulfilled. The new methods are compared with a maximum-likelihood approach. Data from several reviews are used to evaluate the different methods. It is concluded that the new estimation methods provide new opportunities to estimate the defect content.

Keywords
Reviews, inspections, defects, faults, process control, experiment

1 INTRODUCTION

Process control and improvement are essential in any development activity. To be in control and to enable improvements imply that we have to measure and use the measures to take informed decisions about how to continue. This poses two general questions: “What measures should we collect?” and “How do we use the measures to control the process?” It is infeasible to answer these two questions generally, but if we focus on one particular aspect we can make a contribution. The objective in this paper is to try to make such a contribution.

It is well known that rework is a major problem as it means that we have done some work which either was unnecessary or wrong. Thus, it is important to be able to capture any problem we may have at an early stage, or close to the introduction of a problem, instead of continuing to build a software product on an erroneous input. For example, to base the code on a faulty design. One way of coping with this problem is the introduction of inspections, reviews and walk-throughs. These are methods that consistently are reported to be cost-effective and they can be used in any step in the software process. There is, however, still some debate of how they should be conducted, for example, checklists vs perspective-based reading. The methods are discussed in, for example, [1, 2, 6, 7, 8]. We will use the term review throughout this paper to denote the general activity of static analysis through reading.

Reviews are used as a way of controlling quality, in terms of defects, but it is mostly done rather informally. The procedure is informal in the sense that we do not normally have any method or model to objectively judge the quality of the document or code being reviewed. A capture-recapture method has been proposed to overcome this problem [5]. The method is based on the review information from the individual reviewers and through statistical inference, conclusions are drawn about the remaining number of defects after the review. This would allow us to take informed and objective decisions regarding whether to continue, do rework or review some more. The capture-recapture approach is based on applying a statistical method to the collected data. Three methods have been applied for this purpose: the maximum-likelihood estimator, the jackknife estimator and the Chao estimator. The two first methods are described in more detail below. The maximum-likelihood method has been applied in, for example, [5, 13, 14, 15], and the jackknife method has been compared with the maximum-likelihood method in [3, 14]. The Chao estimator has been examined quite recently and some results are presented in [3].

It is by no means simple to get accurate estimates from noisy data using statistical methods, based on assumptions of the behaviour of the real world. This is no reason for not applying statistical methods, on the contrary we believe they should be applied but we must be aware of the shortcomings.

The objective of this paper is to complement the statistical approach to estimation with two methods, which are based on sorting the actual data and then fit a mathematical function. After having analysed the data collected during reviews, we have seen a pattern in terms of that some reviewers are better than others and also that some defects are easier found than others [13]. These observations are by no means surprising, but unfortunately we have to introduce assumptions which violate these observation to apply either the maximum-likelihood method or the jackknife method. Instead of imposing assumptions, we would like to propose to accept these observations, and then plot the data and use functions that fit the data to

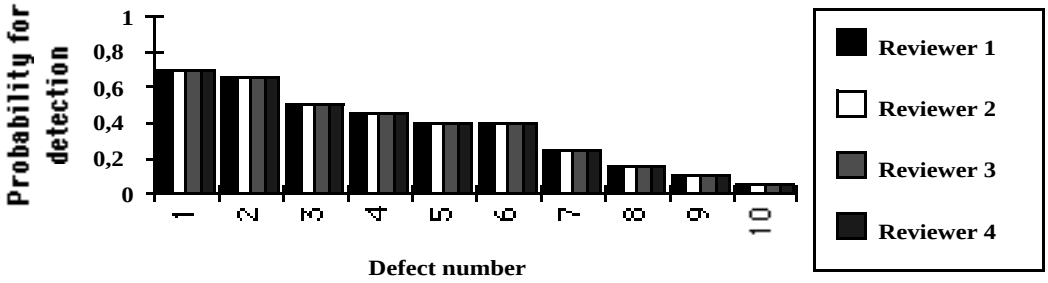

FIGURE 1: Jackknife main assumption: all reviewers have the same detection profile, but different defects may have different detection probabilities.

perform estimations. This approach is similar to how software reliability is estimated using software reliability growth models, see for example [16] and [11].

This paper presents two new methods to estimate the number of remaining defects after a review, and hence to control the software process. The output of the estimation is one important input to decide about a proper continuation after a review. Furthermore, it is important to understand when the different approaches of estimation are best suitable. The two new methods are evaluated for two data sets, one from three reviews of a textual document and the second from reviews of five different C programs.

The paper is organized as follows. In Section 2, the different estimation methods are presented briefly. An evaluation of the maximum-likelihood method vs the two new estimation methods is then presented in Section 3. Finally, a statistical evaluation and comparison of the methods is presented in Section 4 and in Section 5 some conclusions are presented based on the empirical study.

2 DEFECT ESTIMATION METHODS

2.1 Introduction

2.1.1 Existing Methods for Defect Estimation

In [5, 14, 15], it is reported that both the maximum-likelihood method and the jackknife method seemed to consistently underestimate the number of defects. This result is contradicted by [13], where the maximum-likelihood method overestimates the number of defects with approximately 10% in average. An experience-based approach is also presented in [13]. This method is based on historical data, and it does in average as good as the maximum-likelihood method, but it seems a little less sensitive to variations in the data. Based on the difficulty to achieve good and consistent estimates, two new methods are proposed and evaluated in this paper.

Before presenting the new methods, we have to understand the existing methods. A major problem with both the jackknife and the maximum-likelihood method is that the assumptions of the models are not fulfilled in a real world situation.

The main assumption in the jackknife is: All reviewers have the same probability for detecting a specific defect. This implies that the probability for detecting different defects may vary. In summary, this means that all reviewers have the same detection profile, see Figure 1.

The jackknife method is further discussed in [14], and it is not further considered here as its behaviour is similar to the maximum-likelihood method, which is used in this paper as baseline in comparison with the two new methods proposed.

The maximum-likelihood method is briefly introduced in Section 2.3, but let us first consider the main assumption in this method to understand the difference in comparison with the jackknife method. The main assumption of the maximum-likelihood method is: All defects have the same probability for being detected, but the reviewers may have different profiles, see Figure 2.

A further problem with the above estimation methods is that they assume that defects are found more or less ad hoc, and that the reviewers work independently. The main reason for the problem is that structured ways of reading the material in the review may affect the probabilities of finding different defects and thus violate the assumptions of the estimation methods. This should of course not lead us to do ad hoc reviews just to fulfil the assumptions of the estimation methods. On the contrary, we should develop new estimation methods which are less sensitive to structured methods in reading as for example checklist based reading [8] and perspective-based reading [2]. Thus, the introduction of more structured ways of reading when doing reviews emphasizes the need for solving the problems with the assumptions of the above methods.

2.1.2 New Estimation Methods

The above methods are based on statistical analysis of the data. Another possible way to address the problem is to sort the data and plot them i.e. we plot the data according to some criterion, and then based on the plot we try to draw conclusions about the total defect content. The two new methods proposed here are based on the observation that if the data are plotted in a certain way then we can approximate the plot with a curve, and it can be used to estimate the total number of defects.

The novel idea behind the new approaches is that we start from a plot of the actual data, and through the plot we are

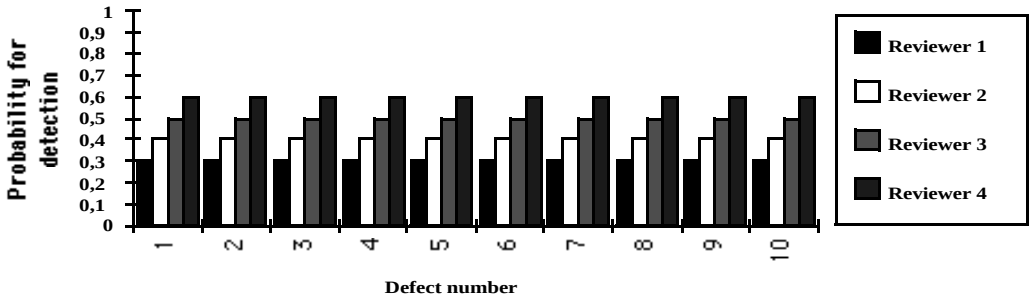

FIGURE 2: M-L main assumption: all defects have the same detection probability for a specific reviewer. Example in the figure: four reviewers with personal probabilities 0.3, 0.4, 0.5 and 0.6 respectively.

able to better understand the data. Thus, we are also able to understand how the actual data deviate from the assumptions in using, for example, a capture-recapture approach.

The two methods identified are:

- Detection profile method

  In this method, the data are plotted with defect number on the x-axis and the number of reviewers that found a particular defect on the y-axis. The ordering of the defects on the x-axis is done based on the number of reviewers that found a specific defect. The data are plotted in a bar graph, and it is assumed that the data can be approximated with an exponential function which then is used to estimate the total number of defects, and hence the number of remaining defects as the reviews are done. This method is described in more detail in Section 2.4 and Figure 3, including assumptions and illustration of the method.

- Cumulative method

  This approach is based on the cumulative plot of all defects found by the reviewers. The data are plotted with the defects on the x-axis, and with the cumulative number of defects found by the reviewers on the y-axis. This means that the first bar gives the number of reviewers that found the defect found by most reviewers, the second bar adds the number of reviewers that found the next defect to the first bar and so on. The y-axis is simply the cumulative number of defects found. It is assumed that the bars can be approximated with an exponential curve. The exponential curve is then used to estimate the total number of defects, and hence the number of remaining defects. This method is described in more detail including assumptions and an example in Section 2.5 and Figure 4.

The three methods (maximum-likelihood, Detection profile and Cumulative) are presented briefly subsequently. To evaluate the different methods, experimental data where the number of defects is known, either through extensive reviews or seeding of the defects, are needed for a full evaluation. An evaluation based on reviews of a textual document is presented in Section 3.2 and a second evaluation based on reviews of C-code is presented in Section 3.3.

2.2 Notation

The following notation is introduced to form a common basis for the three defect estimation methods.

General notation:

- j - reviewer number,
- J - total number of reviewers,
- k - defect number,
- K - total number of defects found,
- n - total number of unique defects found prior to the review meeting,
- n_j - number of defects found by reviewer j,
- m_k - number of reviewers finding defect k (also interpreted as occurrences of defect k).

Factors for the maximum-likelihood method:

- r - number of defects found at the review meeting,
- N - initial number of defects prior to the review.

Factors for the Detection profile method:

- A - number of reviewers to find all defects,
- b - a factor in the Detection profile method (describes how the exponential function decreases).

Factors for the Cumulative method:

- M_k - cumulative number of defect occurrences after k defects have been counted, $M_k = \sum_{i=1}^{k} m_i$ (it could alternatively be interpreted as the cumulative number of reviewers having found defects)
- C - total number of defects to detected,
- d - a factor in the Cumulative method (describes how the exponential function increases).

2.3 Maximum-Likelihood Estimation

The maximum-likelihood estimation method is based on the following assumptions.

- All defects are found by a specific reviewer with equal probability. Differences between reviewers are allowed, but each reviewer is assumed to have the same probabil-

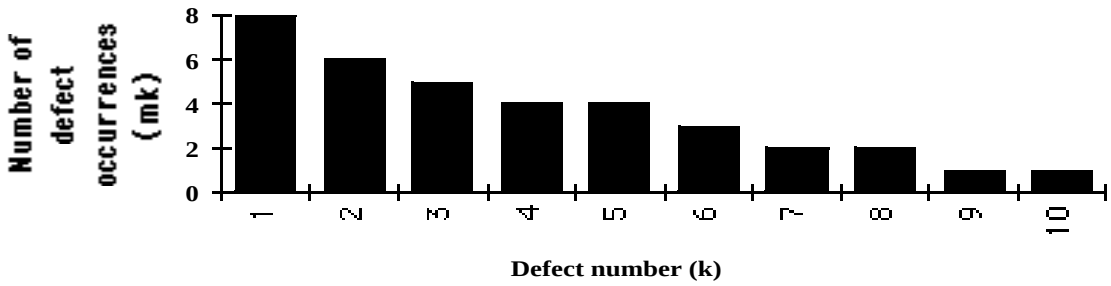

FIGURE 3: Number of defect occurrences (or the number reviewers that found the defects). The defects are sorted in order according to the number of occurrences.

ity of finding all the defects.

- The reviewers work independently.

It is, of course, also assumed that the number of reviewers is at least two, but the method does not assume an upper limit.

It is possible to derive a formula which has a maximum for the most likely value of the initial number of defects prior to the review, i.e. N. Upon maximising, the formula gives the maximum-likelihood estimate of N. n is defined as the number of unique defects found, excluding those found at the review meeting. The definition of n means that defects found by more than one reviewer are only counted once. n_j is the number of defects found by reviewer j and J is the number of reviewers.

The maximum-likelihood function which is maximised is [14]:

$$L(N) = \log\binom{N}{n} + \sum_{i=1}^{J} n_j \log n_j - NJ\log N + \sum_{i=1}^{J} (N-n_j)\log(N-n_j) \quad \text{Eq. 1}$$

This function can be maximised numerically, but since N is an integer the simplest solution is to plot or simply calculate the function for N Š n or (n+r) until the maximum is reached, where r is the number of defects found at the review meeting.

2.4 Detection Profile Estimation

The objective of the Detection profile estimation method is to estimate the number of defects using the information of how many reviewers that found a specific defect. The method is based on sorting and plotting the number of reviewers that found different defects, and then estimating the total number of defects approximating the data with a mathematical function.

Based on the studies we have conducted [13, 15], we have here found it suitable to use an exponentially decreasing function. It should, however, be noted that we would in particular recommend to sort and plot the data, and then based on the plot choose an appropriate function.

A fictitious data set resembling the form we have observed from real review data is given in Figure 3. The objective is to illustrate the form of the plot underlying the Detection profile estimation method. It should ne noted that we have assumed 8 reviewers (J=8) and 10 defects (K=10), which is rather unrealistic but convenient for illustration purpose. In Figure 3, we can see that the first defect is found by all eight reviewers, the second defect is found by six reviewers and so forth. An alternative interpretation is to view it as defect number one occurred eight times. The basic idea behind this method is that the data should be sorted according to the number of occurrences.

In plotting the data according to Figure 3, it can be noted that it should be feasible to approximate the plot with an exponentially decreasing function. This can be used to estimate the total number of defects, if assuming:

- Adding more reviewers means that more defects will be discovered and finally all defects will be found.
- The data resemble an exponential distribution when plotted. This fulfilment of this assumption is probably dependent on the applied review method.
- The total number of defects is estimated from the function, assuming there is an additional defect if the function has a value greater than 0.5 for integers above the number of defects already found. Thus, the total number of defects is estimated as the last integer value which results in that the function is greater than 0.5.
- Furthermore, it is assumed that transforming the exponential function and the data into a linear model does not considerably affect the estimate. Basically, it is assumed that this transformation does not affect the estimate more than the actual uncertainty in the data. A similar model has been proposed in software reliability modelling, and correction factors are derived in [4]. For practical purposes, it is mostly assumed that the correction factors are not needed.

This method has no explicit assumption about independent reviewers, although it is assumed implicitly through the assumption of an exponentially decreasing function, where it is expected that at least some defects are found by a single reviewer.

In mathematical terms, we denote the number of reviewers that found defect k by m_k. It should be noted that we treat m_k as being continuous although k only can assume discrete values. Furthermore, we let A be the number of reviewers to find all defects and b is a factor describing how the

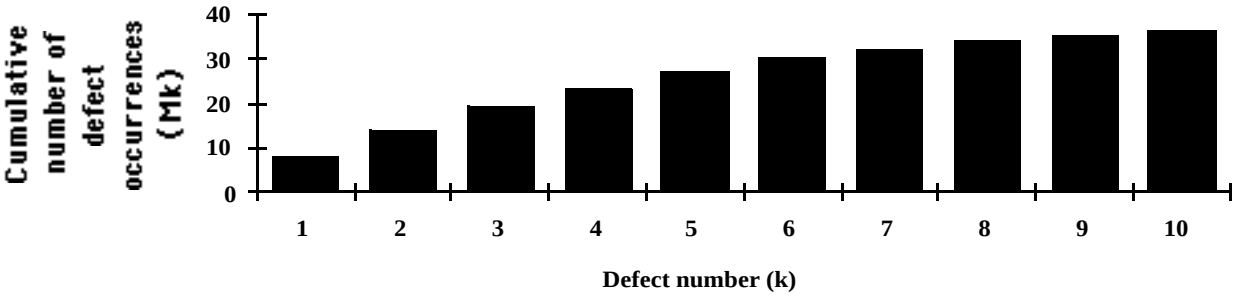

FIGURE 4: The cumulative number of defect occurrences (or number of defects found by the reviewers).

exponential function decreases. The following function is hence assumed:

$$m_k = A \times \exp(-b \times k) \qquad \text{Eq. 2}$$

This function can be made linear by taking the logarithm on both sides. Thus, we obtain:

$$\ln(m_k) = \ln(A) - bk \qquad \text{Eq. 3}$$

Letting $m(k) = ln(m_k)$ and $a = ln(A)$, and then using linear regression, after having taken the logarithm of the different values of m_k, to estimate a and b, we obtain an opportunity to determine the function in Eq. 2.

2.5 Cumulative Number of Defect Estimation

An alternative method is to plot the cumulative number of defects found. Before doing this, it is important to notice that we have two perspectives of the number of defects. First, we may consider the total number of defects in terms of the sum of all remarks by all reviewers. Second, we can just determine the total number of unique defects, hence removing duplicate remarks, i.e. remarks raised by more than one reviewer.

By sorting the defects in order based on the number of reviewers that found a specific defect (which also can be interpreted as the number of defect occurrences of defect k), we obtain a function which is increasing and asymptotically approaching the number of defects detected. Note that we are not plotting the unique number of defects found, but the total number of defects found. This type of curve is often seen in software engineering, when, for example, discussing the contribution of software modules to the overall fault content [12]. A data set is illustrated in Figure 4 to provide an understanding of the plot forming the basis for the Cumulative estimation method. In Figure 4, we see that the first defect occurs eight times, and when adding the second defects (occurring six times), the cumulative number becomes 14 and so on. The main idea underlying this method is that we are able to create a function which asymptotically approaches the total number of defects occurring. It should be noted that it is not the number of unique defects, but the number detected by the reviewers. We have chosen to plot the defects on the x-axis; it is possible to plot the reviewers on this axis and obtain a similar function. The only problem occurring with the latter approach is that we obtain very few bars. It is likely that we have more defects than reviewers, hence getting a smoother function and more data points when fitting the exponential function using the number of defects on the x-axis.

Once again, it is possible to approximate the plot with an exponential function. A function can be formulated based on the following assumptions:

- Adding more reviewers means that more defects will be discovered and finally all defects would be found.
- The number of unique defects, which is the parameter of primary interest, can be derived from the cumulative number of defects found. It is assumed that the number of unique defects not found, can be derived directly from the estimate of the cumulative number of defects, minus the number of defects already found. In other words, we assume that all of the remaining defects are considered to be unique.

  This assumption means that the estimate we obtain is most likely an upper bound of the number of remaining defects, or at least a rather conservative estimate.

This method has no explicit assumption about independent reviewers, although it is assumed implicitly through the assumption of an exponential function, where it is expected that the contribution of individual reviewers vary. The latter may not be visible in the data if the reviewers cooperate.

In this particular case, we let M_k be the number of defects detected after k defects have been included in the cumulative number, and denoting the total number of defects found by the reviewers by K (note that it is not unique defects, but the total number of defects found). It should be noted that we treat M_k as a continuous function although k can only assume discrete values. Furthermore, let C be the total number of defects to be detected and d the exponential factor describing the curve. Thus, we obtain:

$$M_k = C \times (1 - \exp(-dk)) \qquad \text{Eq. 4}$$

This function is similar to the one proposed as the mean value function in the Goel-Okumoto software reliability model [9]. The meaning of M_k is very similar to the one in that model, and thus the parameters in Eq. 4 can be determined in the same way. In particular, it has been

TABLE 1. Summary of the methods.

Method	Max.-Likelihood	Detection profile	Cumulative
Approximate function	Maximum-like-lihood function	Exponentially decreasing function	Exponentially increasing function
Criterion: estimate the number of remaining defects	The estimate is equal to the value of *N* that maximizes the M-L function minus the number of unique defects found.	The estimate is equal to the last time the function is greater than 0.5 minus the number of unique defects found.	The estimate is equal to *C* - *K*, where *K* is the total number of defects found by the reviewers.

shown that the parameters of the Goel-Okumoto model can be estimated for grouped data, i.e. the number of faults in certain intervals is logged and then the parameters in the model can be determined. This can be done after having data from a number of intervals. The major problem with the data in software reliability modelling is that it is not normally in order, i.e. due to random variations the number of faults in the intervals are not decreasing as expected. Thus, making it hard to estimate the parameters until after a large number of intervals.

The situation with the review data is far better. The data are sorted so that the assumptions of the model are fulfilled, which means that there should be no problem in estimating the parameters in Eq. 4 using the estimation method from the software reliability model. This method is based on a maximum-likelihood estimation where the *d* parameter has to be determined numerically and then the *C* parameter can easily be estimated. The two likelihood functions, which are used to determine *C* and *d* can be found in, for example, [16], and they are given in Eq. 5 and Eq. 6. Eq. 5 is first applied to find *d* numerically, *C* is then easily found through Eq. 6.

$$\frac{\partial}{\partial d}\ln(L(m_1, m_2, \ldots, m_n; d)) =$$

$$= \sum_{k=1}^{n} \frac{m_k(k\exp(-dk) - (k-1)\exp(-d(k-1)))}{\exp(-d(k-1)) - \exp(-dk)} - \frac{Kn\exp(-dn)}{1-\exp(-dn)} = 0 \qquad \text{Eq. 5}$$

$$C = \frac{K}{1-\exp(-dn)} \qquad \text{Eq. 6}$$

where *K* is the total number of defects and *n* is the unique number of defects found, m_k is the number of reviewers that found defect *k*. The formula in Eq. 5 is a simplification of the more general formula which is presented in the software engineering literature. The simplification is due to the fact that the defects are numbered from one and onwards. The more general formula is able to cope with varying time intervals, see for example [16].

2.6 Summary of Methods

In summary, we have emphasized three different methods for estimating the total number of defects in a review object. One of the methods has been presented and investigated earlier [5, 13, 14, 15]. The two other estimation methods are new, in particular the idea of sorting the actual data and fit mathematical functions is novel. The information obtained from the methods can be used to control the software development process, more precisely we are able to use the estimates as one criterion whether to continue to development or to go back, either for further reviews or for rework before continuing on.

The function for each of the methods and the criterion to derive the estimate of the number of remaining defects are summarized in Table 1.

3 EVALUATION OF THE METHODS

3.1 Introduction

An important problem of the methods is remaining, and that is how to apply the proposed methods. In particular, we must determine when to use the different methods or if all three methods should be used. In the latter case, we must also determine how to combine them. A number of questions arises:

1. Is there a pattern in terms of the order of the estimates? Can we, for example, always expect the estimate from the cumulative method to give the highest estimate?
2. Is the best estimate, i.e. the one closest to the true value, dependent on the quality and type of the review? In other words, if we are able to judge the competence of the reviewers and, for example, log the time they spent reading, should this affect which method we use for estimation? Is the best method dependent on the reading technique, for example checklist vs. perspective-based reading?
3. Should we trust one estimate, or should we take the mean value or even a weighted average? To weigh estimates can be one way of coping with the uncertainty in which estimate is actually the best, and it can be a feasible solution until we have learnt which method that works best in our particular environment.

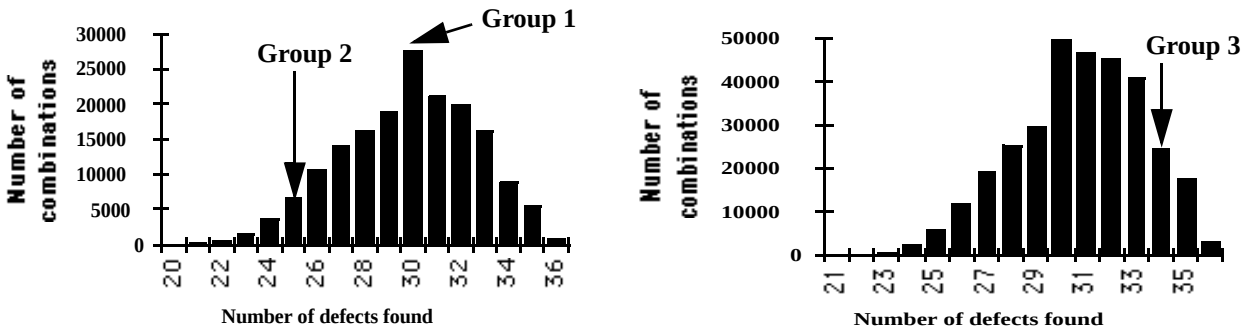

FIGURE 5: Group performance when selecting seven respectively eight reviewers out of the 22 available reviewers.

In order to address these questions, two separate evaluations are performed. It is important to notice that in order to enable an evaluation, we have to know the number of defects in the document prior to the review, otherwise we are unable to compare the estimate with the actual value. Thus, the evaluation is focused upon two experimental studies conducted earlier where the total number of defects is known, either through defect seeding or prior extensive reviews and testing. In the latter case, the uninspected and untested code was saved to make it useful for evaluation purposes.

The first evaluation is based on data collected when reviewing a textual document. The data set is presented in [15]. The second evaluation is from C-code reviews, and the data are presented in [13].

3.2 A First Evaluation: Textual Document

This example is based on a review of a textual document, where the defects were seeded. In total 38 defects were present in the document and 22 reviewers participated in the experiment. The reviewers were in the initial analysis [15] divided into three groups by random. The analysis in [15] is concerned with maximum-likelihood estimates and a method for dividing the review data into several classes and then perform estimations for the different classes independently. The main idea is to use information from previous reviews to help us improve the estimations. The main advantage would be that we base the estimation on prior experiences, and the main drawback is that we become dependent on previous reviews. It would of course be most beneficial if the best estimation is solely based on data from the current review.

The data from the review of the textual document can be used here for evaluating the new methods in comparison with the maximum-likelihood method. It should be noted that based on that we have data from 22 independent reviewers a large number of groups can be created. We have chosen here to focus on the three groups used in [15], but to complement this information we have investigated how good these three groups are in comparisons with all groups of size seven and eight that can be created, see Figure 5.

From Figure 5, it can be seen that the weakest group with seven reviewers that can be created from the 22 reviewers find together 20 defects out of the 38 defects that were present in the document. The distribution in Figure 5 resembles the normal distribution in shape as can be expected. It should be noted that the total number of combinations are 170544, i.e. the number of ways that we can select seven reviewers out of the available 22. The data from the figure can now be used to actually determine where in the distribution the random groups are placed.

The outcome of applying the three estimation methods presented above, and the actual number of defects found by the reviewers are presented in Table 2, and the placements of the group performance are shown in Figure 5. The placements illustrate how good the actual groups are in comparison with a random group.

TABLE 2. Application of the three estimation methods.

Total number of defects	Number of defects found by the reviewers	Correct value	Maximum-likelihood method	Detection profile method	Cumulative method	Mean value of estimation methods
Group 1 (7 reviewers)	30	38	30	38	46	38
Group 2 (7 reviewers)	25	38	25	32	36	31
Group 3 (8 reviewers)	34	38	34	44	51	43

TABLE 3. Application of the three estimation methods.

Total number of defects	Number of defects found by the reviewers	Correct value	Maximum-likelihood method	Detection profile method	Cumulative method	Mean value of estimation methods
Program 3A (5 reviewers)	18	22	28	25	25	26
Program 4A (5 reviewers)	15	16	16	19	22	19
Program 5A (5 reviewers)	15	16	17	18	20	18
Program 6A (5 reviewers)	33	35	39	41	46	42
Program 7A (4 reviewers)	20	20	25	24	26	25

From Table 2, the best estimation method for each group can be determined:

Group 1: Detection profile method (correct estimate)

Group 2: Cumulative method (underestimate with two defects)

Group 3: Maximum-likelihood method (underestimate with four defects)

Furthermore, it can be noted that the mean value of the different estimates does not perform better than the best method. On the other hand, if we do not know, based on previous experience, which method is the best then the mean value may be a sensible approach, instead of taking one specific method.

This first evaluation seems to indicate two things. First, that there may be a correlation between the performance of the review team and the best estimation method. This is based on comparing the placement of the three groups in Figure 5 and noting that different estimation methods are best for the different groups. Second, that the mean value of the estimation methods may be better than choosing one of the methods without knowing that the actual choice of method is well-founded.

Based on this analysis, it is now possible to address the three questions posed above.

The estimates seem to come out in the same order, i.e. maximum-likelihood methods, Detection Profile method and Cumulative method. Is this a common pattern?

This evaluation does not contradict the assumption that there is a relation between the performance of the group and the best estimation method. It is, however, too early to draw any conclusions. The issue has to be further investigated.

The mean value seems to be a sensible approach, if we are not able to show any direct correlation between the performance of the group and the best estimation method, or if we are unable to in advance predict the performance of the assigned review team.

To further address these questions, and to study the methods in code inspections a second evaluation is performed based on the data presented in [13].

3.3 A Second Evaluation: C-Code Review Data

The experiment reported in [13] is based on reviews of programs written within the Personal Software Process [10]. The programs are written in C, and the defects are real defects made by the software engineer. The total number of defects are assumed to be known after extensive reviews and testing. The experiment was made using five programs (PSP programs 3A-7A) and each reviewer read three programs. This means that four programs were reviewed by

TABLE 4. Absolute error in percentage based on the different approaches.

	M-L	Profile	Cumulative	Mean
Group 1	21.1%	0%	21.1%	0%
Group 2	34.2%	15.8%	5.3%	18.4%
Group 3	10.5%	15.8%	24.2%	13.2%
3A	27.3%	13.6%	13.6%	18.2%
4A	0%	18.8%	37.5%	18.8%
5A	6.3%	12.5%	25%	12.5%
6A	11.4%	17.1%	31.4%	20%
7A	25%	20%	30%	25%

TABLE 5. Analysis of the mean error and its standard deviation.

Error	ML	Profile	Cumulative	Mean
Mean	17.0%	14.2%	23.5%	15.8%
Standard deviation	11.7%	6.2%	10.3%	7.5%

five reviewers and the final program was reviewed by four reviewers. This is hence reviews with less participants than in the first evaluation.

The results of the second evaluation is presented in Table 3.

It should be noted that based on the performance of the individual reviewers and the number of possible combinations, we are once again able to judge how good the review teams actually selected are [13]. Based on the analysis in Table 3 and the possible ways of combining reviewers into different groups, we are now able to address the three questions posed above.

The estimates do not come out in the same order. It should, however, be noted that the Cumulative method consistently produces the highest estimate. This is in accordance with expectation when comparing with the formulation of the method.

This evaluation contradicts the assumption that there is a relation between the performance of the group and the best estimation method.

The mean value seems to be a sensible approach, unless we settle for either the maximum-likelihood method or the Detection profile method and use the Cumulative method as a conservative estimate. This issue is further addressed below when doing a statistical evaluation of the differences in the estimates.

We have seen that it is not possible to find one superior model, and that the best method varies. The challenge to establish a relationship between factors such as competence among the reviewers, ability to cover different perspectives and time spent in review, and the best estimation method remains.

4 STATISTICAL EVALUATION

The mean absolute error in percentage is used as a measure of the goodness of the different methods. We are interested in evaluating both the individual methods, and the mean value approach discussed above. The mean value approach is a result of a qualitative observation based on the outcome of the two evaluation (textual and C-code). The objective here is then to test statistically if one of the four approaches (three individual and mean value) is statistically better than the others. Hypothesis: the methods produce the same mean error. The hypothesis is tested with an ANOVA test based on the figures in Table 4. The three first groups refer to the textual document and the programs denoted 3A-7A are the C-programs.

An ANOVA test is not able to show any statistical significance between the methods. It should, however, be noted that there is a statistical significance when comparing only the best method (Detection profile method), where best is determined by the mean error, and the worst method (Cumulative method). To gain further understanding of the differences, the mean and standard deviation of the error are determined, see Table 5.

It is worth noting that the mean error for the Detection profile method is lowest, and the standard deviation is also considerably lower than the other individual methods. Thus, based on the descriptive statistics presented, it is clear that the Detection profile method deserves further studies, and it may be worthwhile applying as a means for process control in conjunction with reviews. Furthermore, the Cumulative method may very well be applied to obtain a conservative estimate. Thus, we conclude that the two new methods, based on sorting and plotting the data, provide new opportunities in estimating the number of remaining defects after reviews, and they are suitable complements to other approaches such as the maximum-likelihood method. More studies are, however, needed in order to make a more definitive recommendation, although the results are promising.

5 SUMMARY

Two new methods for estimating the number of remaining defects after a review have been introduced. Both methods are based on sorting the actual data and then plot the data according to a criterion. The novelty of the approach lies in the sorting of the data and then fitting a mathematical function, rather than basing the estimates on statistical analysis with underlying assumptions which mostly are not fulfilled. We have used two specific functions in this paper, but by sorting and plotting the data intelligently, we could fit any function that seems suitable.

The new methods provide new insight into the area of process control from review data. The Detection profile method gives the lowest mean error and provides a stable estimate (low standard deviation), and the Cumulative method may be used as a worst case estimate. We were, however, unable to prove the difference between the methods with statistical significance.

It can be concluded that further studies are needed, although the two new estimation approaches are promising. Thus, we would like to encourage others to use the new methods in order to improve our understanding of them in order to use them effectively. In particular, more studies must be directed towards evaluating when certain estimation methods should be applied based on, for example, number of reviewers, competency, experience and review effort. Moreover, we must gain an improved understanding regarding the pattern in the estimation, for example, when does a specific method overestimate?

ACKNOWLEDGMENTS

We would like to thank the reviewers of the paper for valuable comments. Moreover, we would also like to thank Dr. Lionel Briand, Fraunhofer Institute of Experimental Software Engineering, Kaiserslautern, Germany for insightful and useful comments.

This work was partly funded by The Swedish National Board for Industrial and Technical Development (NUTEK), grant 1K1P-97-09673.

REFERENCES

1. A.F. Ackerman, L.S. Buchward and F.H. Lewski. Software Inspections: An Effective Verification Process. *IEEE Software*, May 1989, pp. 31-36.
2. V. Basili, S. Green, O. Laitenberger, F. Shull, F. Lanubile, S. Sørumgaard and M. Zelkowitz. The Empirical Investigation of Perspective-Based Reading. *Empirical Software Engineering: An International Journal*, Vol. 1, No. 2, pp. 133-164.
3. L. Briand, K. El Emam, B. Freimut, B. and O. Laitenberger. Quantitative Evaluation of Capture-Recapture Models to Control Software Inspections. In *Proc. 8th International Symposium on Software Reliability Engineering* (Albuquerque, New Mexico, USA, 1997), pp. 234-244.
4. P.A. Currit, M. Dyer and H.D. Mills. Certifying the Reliability of Software. *IEEE Transactions on Software Engineering*, 11, 12 (1986), pp. 1411-1423.
5. S.G. Eick, C.R. Loader, M.D. Long, L.G. Votta and S.A. Vander Wiel. Estimating Software Fault Content Before Coding. In *Proc. 14th International Conference on Software Engineering* (Melbourne, Australia, 1992), pp. 59-65.
6. M.E. Fagan. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal* 15, 3 (1976), pp. 182-211.
7. M.E. Fagan. Advances in Software Inspections. *IEEE Transactions on Software Engineering* 12, 7 (1986), pp. 744-751.
8. T. Gilb and D. Graham. Software Inspections. *Addison-Wesley*, Reading, Massachusetts, USA, 1993.
9. A.L. Goel and K. Okumoto. Time-Dependent Error-Detection Rate Model for Software and Other Performance Measures. *IEEE Transactions on Reliability.* R-28, 3 (1979), pp. 206-211.
10. W.S. Humphrey. A Discipline for Software Engineering. *Addison-Wesley*, Reading, Massachusetts, USA, 1995.
11. M.R. Lyu (editor). Handbook of Software Reliability Engineering. *McGraw-Hill*, 1996.
12. N. Ohlsson, M. Helander and C. Wohlin. Quality Improvement by Identification of Fault-Prone Modules using Software Design Metrics. In *Proc. of Sixth International Conference of Software Quality* (Ottawa, Canada, 1996), pp. 1-13.
13. P. Runeson and C. Wohlin. An Experimental Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections. Submitted to *Empirical Software Engineering: An International Journal*, 1997.
14. S.A: Vander Wiel and L.G. Votta. Assessing Software Designs Using Capture-Recapture Methods. *IEEE Transactions on Software Engineering* 19, 11 (1993), pp. 1045-1054.
15. C. Wohlin, P. Runeson and J. Brantestam. An Experimental Evaluation of Capture-Recapture in Software Inspections. *Software Testing, Verification and Reliability* 5, 4 (1995), pp. 213-232.
16. M. Xie. Software Reliability Modelling. *World Scientific Publishing Co. Pte. Ltd.*, 1991.