C. Wohlin, P. Runeson and J. Brantestam, "An Experimental Evaluation of Capture-Recapture in Software Inspections", Journal of Software Testing, Verification and Reliability, Vol. 5, No. 4, pp. 213-232, 1995.

An Experimental Evaluation of Capture-Recapture in Software Inspections

Claes Wohlin[*], Per Runeson[**] and Johan Brantestam[**]

[*] **Dept. of Communication Systems**
Lund Institute of Technology
Lund University
Box 118
S - 221 00 LUND
Sweden

[**] **Q-Labs**
IDEON Research Park
S - 223 70 LUND
Sweden

Abstract

The use of capture-recapture to estimate the residual faults in a software artifact has evolved as a promising method. However, the assumptions needed to make the estimates are not completely fulfilled in software development, leading to an underestimation of the residual fault content. Therefore, a method employing a filtering technique with an experience factor to improve the estimates of the residual faults is proposed in this paper.

An experimental study of the capture-recapture method with this correction method has been conducted. It is concluded that the correction method improves the capture-recapture estimate of the number of residual defects in the inspected document.

Keywords:

Experimental software engineering, software inspections, capture-recapture, software metrics, fault content estimation

1.0 Introduction

Software inspections are described as one of several cost-effective techniques to improve software quality. Inspections have been discussed extensively since their introduction by Fagan (Fagan, 1976). A comprehensive description of software inspections can be found in (Gilb *et al.*, 1993).

A method of estimating the residual faults after an inspection using capture-recapture has been proposed by (Eick *et al.*, 1992). This method has been further studied by (Vander Wiel *et al.*, 1993), who concluded that the method often underestimated the actual number of faults. The capture-recapture method can be applied to any type of document being inspected, and is thus applicable not only during code inspections, but, for example, also during inspection of requirements specifications. The capture-

recapture method can be combined with any other suitable measures for software inspection. A number of measures are proposed by (Gilb *et. al*., 1993), for example, a measure for estimated effectiveness of the inspection. This type of measure is, however, not discussed here.

The objective of this study was to find solutions to some problems identified in applying the capture-recapture method in software inspections. The proposals were tested using an experiment in which the number of faults in the document is known.

2.0 Brief introduction to capture-recapture

2.1 Inspections and capture-recapture

The capture-recapture method was originally used to estimate animal populations through multiple independent counters. Statistical inference is used to draw conclusions about the total population size. In software engineering the method is applied to counting the number of faults based on data from several independent sources. It is assumed that one reviewer capture a certain fault, and as other reviewers identify the same fault they are said to recapture the fault. This means that the reviewers can be treated as independent counters in a similar way to when estimating wildlife populations. The method has been proposed by (Eick *et al.*, 1992) for fault content estimations in inspections in the software development process. The method is hence quite new but it is extremely simple to apply and the result is intuitively appealing.

Inspections employing the capture-recapture method consist of four different steps:

1. Inspection preparation

   Inspection preparation means reading the documents to be inspected carefully and noting all faults found. Each reviewer is responsible for being well-prepared before going to the inspection meeting. “Well-prepared” means that a minimum of preparation time should be required to be able to include a reviewer in the estimation process.

2. Inspection meeting and data collection

   The inspection is led by a moderator (or chairman) who is responsible for collecting the fault information from the individual reviewers. The moderator should also note the preparation time for each reviewer and if it is judged that one or several reviewers have not prepared themselves as required, they should be excluded from the estimation process as their poor preparation will influence the estimate in an unpredictable way. It is recommended that the moderator determines a minimum preparation time in advance for the data to be useful.

3. Data analysis

   The fault data collected during the inspection form the basis for analysis. The analysis is performed by applying an estimation method to the data collected. Different methods can be used and they are based on different assumptions. Two methods are discussed in Section 2.2.

4. Result evaluation and decision making

   As a result of the data analysis, the number of faults prior to the inspection, and hence the residual faults after the inspection, can be estimated. The method provides some important estimates which can be used to estimate trends and formulate thresholds. It is hence possible to define thresholds for approval of a particular document, but specific values must be determined explicitly for each organisation applying the capture-recapture method.

   The estimates may also form the basis for process improvement, since if the number and type of faults can be determined, it will then be possible to improve both the document creation process (which may be the software development process) and the inspection process itself by improving the inspection guidelines.

The capture-recapture method is closely related to fault seeding, where a known number of faults are entered into the documents to be used in the estimation process. The major advantage with the capture-recapture method is that faults do not have to be introduced into the documents; the faults already present are used in the estimation procedure. The capture-recapture method does not assume that any faults are known prior to inspection preparation. The estimate is based only on faults found by individual reviewers preparing for the inspection.

2.2 Estimation methods in capture-recapture

2.2.1 Maximum-Likelihood estimation method

The capture-recapture method as presented by (Eick *et al.*, 1992) is based on the Maximum-Likelihood estimation method. The estimation method is based on the following assumptions.

- All faults are found by a specific reviewer with equal probability. Differences between reviewers are allowed, but each reviewer is assumed to have the same probability of finding all the faults.
- 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 faults prior to the inspection, i.e. *N*. Upon maximisation, the formula gives the Maximum-Likelihood estimate of *N*. *n* is defined as the number of unique faults found, excluding those found at the review meeting. The definition of *n* means that faults found by more than one reviewer are only counted once. n_j is the number of faults found by reviewer *j* and *m* is the number of reviewers. The Maximum-Likelihood function which is maximised is (Vander-Wiel *et. al.*, 1993):

(Eq. 1)

$$L(N) = \log\binom{N}{n} + \sum_{j=1}^{m} n_j \log n_j - Nm\log N + \sum_{j=1}^{m} (N - n_j)\log(N - n_j)$$

This function can be maximised numerically, but 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 faults found at the review meeting.

2.2.2 Jackknife estimation method

The Jackknife estimation method is based on the following assumptions.

- All reviewers have the same probability of detecting a specific fault, but the different faults may have different detection probabilities. This assumption is the opposite to that made above for the Maximum-Likelihood method.
- The reviewers work independently.

The Jackknife estimation method has been evaluated by (Vander Wiel *et al.*, 1993), but it was not shown to be superior to the Maximum-Likelihood method. Therefore, the Maximum-Likelihood method was used throughout this investigation.

The first assumption above, made for the Jackknife estimation method is, however, used as a basis below when discussing how the faults found during inspection can be divided into different classes through a filtering technique. The assumption is not used as stated, but the main idea behind the assumption is used, see below.

3.0 Capture-recapture problems

There are some problems associated with the capture-recapture method using Maximum-Likelihood estimates.

- A consequence of the assumptions used in the model is that if many reviewers are involved in the inspection, then it is very unlikely that any faults are estimated to remain at all. However, practical experience contradicts this. Some faults are very difficult to find and some may not be found by any reviewer, hence this causes a problem when applying the method.
- Faults found at the inspection meeting are not included in the estimation procedure. The faults found at the meeting are subtracted from the estimate, hence lowering the estimate and if many faults are found at the meeting then the estimate of the number of faults prior to the inspection will be equal to the number of faults found. A large number of faults found at the meeting indicates that a large number of faults remain, but the method implies the opposite, as the estimation method is based on faults found prior to the meeting.

- The estimation method does not work if there is no overlap between the reviewers. The method gives an infinite estimate of the faults prior to the inspection. No overlap should indicate that a large number of faults remain, but the method is unable to cope with this situation accurately.

The two first items are closely related to the assumption of the equal probabilities of a specific reviewer finding the different faults present in the inspected document. The first item is related to this assumption since it is well known that some faults are very difficult to find and may also be found through a combination of reviewers, hence contradicting the assumption, see Section 4.0. The second item reduces the estimate of the residual fault content incorrectly, because it is assumed that all faults found at the inspection meeting are part of the estimate and in many situations this means that the number of residual faults after the meeting is estimated to be equal to zero. This is incorrect since some faults may be impossible to find for a single reviewer, but it is possible to locate the fault at the meeting due to the combinations of competence and experience of the individual reviewers. Therefore, it is essential to propose an estimation method, which tries to compensate for the unrealistic assumption about equal probabilities for all faults in the document for a single reviewer. The solution is, however, not the Jackknife estimation method as shown by (Vander Wiel *et. al.*, 1993).

4.0 Reviewer profile

4.1 Reviewer profile model

As mentioned before, the Maximum-Likelihood method is based on the assumption that for a specific reviewer, the probability of finding each different fault is the same. This is obviously not true, since it is known that different faults are more or less easy to find. In order to explain the behaviour of a reviewer, a simple reviewer model is introduced. It is based on dividing the faults into k classes, denoted F_i where $1 \leq i \leq k$, each representing a fault type with a certain probability, p_i, of being found by the specific reviewer. This means that the probability of a reviewer finding a fault from fault class F_i is p_i. By presenting all the fault classes with their probabilities in a diagram, a reviewer profile is obtained, see Figure 1.

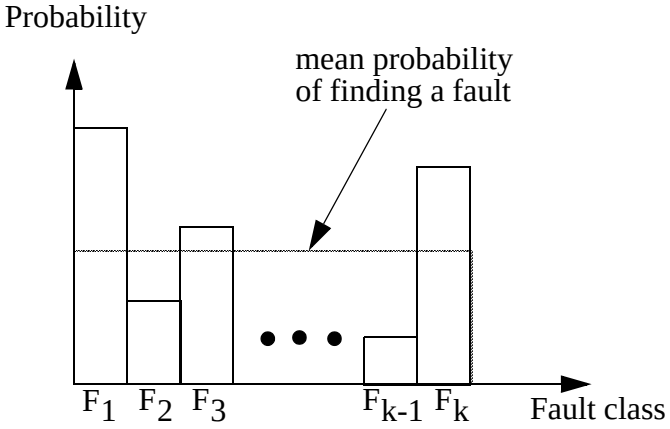


FIGURE 1. Reviewer profile for a reviewer and mean probability of finding a fault.

4.2 Example of applying the model

Different reviewers have different skills in finding different faults. However, it is very likely that different reviewers have fairly similar reviewer profiles, i.e. the reviewers will find certain faults difficult to locate, while others are relatively easy to find. This is due to the fact that the reviewers have similar backgrounds, i.e. they speak the same language, they have roughly the same education, and so on. This makes it likely that there will be a number of faults belonging to certain fault classes which almost all the reviewers will find, and faults belonging to other fault classes which only a few reviewers will find. It can hence be argued that different reviewers have a higher probability of finding some faults than others. This, however, does not mean that they have the same probability of finding these faults as, for example, is assumed in the Jackknife estimation method.

In this discussion, a very simple example is used with only two fault classes and two reviewers. The reviewer profiles, which are quite similar, are presented in Figure 2. The model is applicable to any reviewer profile and any distribution between fault classes, see below.

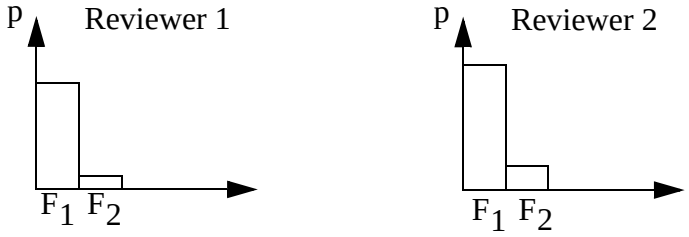


FIGURE 2. Example reviewer profiles.

Assume that the two reviewers inspect a document with the same number of faults in each fault class. It is then very likely that more faults belonging to fault class 1 (F_1) will be found. Since both the reviewers are more skilled in finding these faults, they are also more likely to find the same faults. On the other hand, the reviewers find less faults belonging to F_2, and thus the probability is very small that they will find the same faults belonging to this class, see Figure 3.

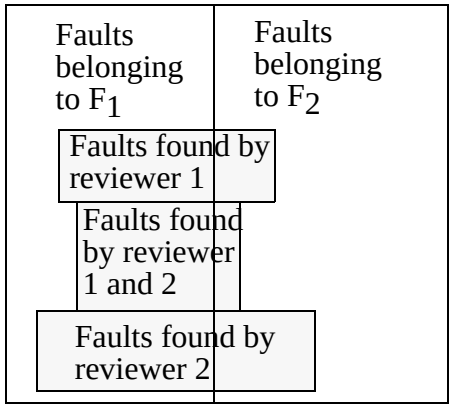


FIGURE 3. Identified faults.

If the Maximum-Likelihood method was to be applied to the data from this inspection it would probably indicate that most faults had been found. However, it is known that a large number of faults remain in fault class F_2. The reason for this is that the majority of the identified faults are from fault class F_1 and among these faults there is a large overlap since these faults are very likely to be found by both the reviewers.

Based on this simple example, which was presented to highlight the problems associated with equal probabilities of finding all faults for each reviewer, it is concluded that the assumption is wrong and the method of estimation must be adapted to take account of this fact.

5.0 Solution procedure

The three problems described in Section 3.0 are critical when the estimation technique is put into practice. The capture-recapture method including Maximum-Likelihood is still tractable as it allows for the estimation of the residual fault content, but it requires some improvements. Therefore, a procedure which takes into account the effects of the above problems is proposed. The method is, to some extent pragmatic, but it is better to be pragmatic than stubborn and continue to use a mathematically correct method which notoriously produces underestimates.

The procedure can be summarized as follows.

1. The faults must be divided into groups such that it is more likely that the assumption regarding the same probability of detection holds. This can be achieved by introducing a filter which divides the fault data into different classes. The filter must be based on the number of reviewers who find a specific fault, since if many reviewers find a specific fault then it can be assumed that it is more likely that a reviewer will find this fault, than a fault which has only been found by one reviewer. The first step is therefore to identify a suitable filter.

   The construction of a filter is based on the same assumption as the Jackknife estimation method, see Section 2.2.2, i.e. the faults are not detected with the same probability, as assumed when using the Maximum-Likelihood estimation method for all faults in one class. However, the filtering technique does not require that the detection probabilities to be the same for all reviewers, which is assumed in the Jackknife estimation method.

   An extreme filter is to divide the fault data into classes based on the exact number of reviewers who have found a specific fault, i.e. class 1: faults found by all reviewers, class 2: faults found by all reviewers but one and so forth. This is the most fine-grained filter that could be constructed. This type of filter would not work in practice as a certain overlap between the reviewers is needed to perform the estimation of the residual fault content.

   Another example of a filter is to divide the fault data based on percentages: for example, class 1: faults found by at least 50% of the reviewers and class 2: faults found by less than 50% of the reviewers. Two classes are used when discussing filtering techniques below. An analysis of the estimates based on changes in the per-

centage value is presented in Section 7.0, based on data from the experimental study. The choice of a filter must be based on experience so that the best possible filter for a specific organisation is identified.

2. The Maximum-Likelihood function must be used for each fault class and the estimates are summed to obtain the total number of faults in the inspection document, and hence the estimate of the residual faults. If the number of faults found at the meeting is larger than the estimate of the residual faults at the start of the meeting, then the estimate must be considered as unreliable.
3. If the reviewers have no fault in common in any of the fault classes, then the estimation method fails, and a method of overcoming this problem must be formulated. The estimation must be based on experience, i.e. the faults found by single reviewers must be multiplied by an experience constant. Initially, the constant is assigned a value of 2, which is equivalent to assuming that as many faults as have been found by single reviewers are remaining. This rule must be applied when the estimation method fails. In the long run, experience must be the basis for determining the multiplicative factor. (See also Section 7.0.)
4. The estimate of the number of faults prior to the inspection meeting is made by summing the estimates, hence the residual faults are easily derived. This can be complemented with a confidence analysis, as discussed by (Vander Wiel *et al.*, 1993).

To summarize, the division into classes by applying a filtering technique makes use of one of the advantages of the Jackknife estimation method, but it does not suffer from its drawbacks. The Maximum-Likelihood estimation method is then applied to each fault class, which makes the assumptions of the Maximum-Likelihood method more realistic. The proposed procedure was then evaluated through an experiment.

6.0 The experiment

6.1 Introduction

An experiment was designed and conducted to evaluate the capture-recapture method when applied to fault data obtained from inspections. The data were divided into fault classes and the residual fault content after the inspection was estimated using the Maximum-Likelihood method for each fault class. The objective was to assess the classification technique, which is based on filtering the collected fault data.

The hypothesis is that it is possible to identify a classification scheme or filtering technique which improves the fault content estimation from fault data collected from inspections.

6.2 Experimental design and instrumentation

6.2.1 Inspection team

The experiment was conducted with an inspection team of 22 reviewers randomly divided into three groups. This resulted in two inspection groups each with seven reviewers and one group with eight reviewers. The 22 reviewers were a mixture of fourth-year undergraduate students and personnel employed as software engineering consultants at Q-Labs.

All reviewers were familiar with inspection techniques in general and no special instruction or training was given prior to the experiment. Prior to inspection, the team was informed about the document to be inspected and also about the objective of the study, including information on the capture-recapture method.

6.2.2 Inspected document

As strict control of the experiment was desired, it was deemed infeasible to use a real software document. The experiment required that the number of faults in the inspected document be known in advance. Therefore, it was decided to use a textual document since many documents in software production are written in natural language. The capture-recapture method can be applied to any type of document, hence the choice of document should not influence the outcome of the experiment to any great extent.

The inspected document was a technical paper which had first been run through a spell checker and had then been edited by an English-speaking professional prior to publication, see (Wohlin et al., 1994). The document consisted of 18 double-spaced pages, of which 2 pages were figures. In total, the document contained 5645 words.

As the objective was to evaluate the estimation procedure, it was necessary to introduce a known number of faults into the document. Therefore, it was decided to seed a known number of faults into the document and then use the edited document as the correct answer after the inspection. The faults were randomly distributed in the document.

6.2.3 Faults in the document

The objective was to seed faults which were similar to the faults that had been removed based on the spell-checker and the editing of the document. The types of faults to be introduced were discussed carefully and it was agreed that faults which could clearly be identified as faults should be seeded; implying that only faults which it was deemed that the reviewers could find and identify as faults should be seeded. This was viewed as a prerequisite for the experiment and it was not believed to influence the results. Thus, it was decided not to seed dubious formulations, which could always be debated as being correct, or not.

In total, 37 faults were seeded into the document and one unknown fault was also found. No other faults had been found since, by the authors or communicated by the

readers of the paper, see (Wohlin et. al., 1994). The number of faults of each type, including the unseeded fault, were as follows:

- 6 spelling errors considered simple to find (e.g. detialed instead of detailed)
- 9 spelling errors considered difficult to find (e.g. hypotesis instead of hypothesis)
- 4 logical errors (e.g. two definitions changed place)
- 3 incorrect references (e.g. figure 7 instead of figure 6)
- 5 wrong or missing words (e.g. pacific instead of specific)
- 11 grammatical errors (e.g. is instead of are)

All faults could objectively be considered as faults, as an edited version of the document existed and the faults had been seeded very carefully. No judgement regarding good or bad wording was to take place concerning the faults.

6.2.4 Threats to the validity of the experiment

The following potential threats were identified.

- The inspected document is not a software document.

  The capture-recapture method can be applied to any type of document, and many documents produced during software development are textual documents. Therefore, the threat is regarded as non-critical.

- The faults are seeded.

  It was unfortunately necessary to seed faults to enable assessment of the estimation procedure. Fault seeding is always a potential problem, but the objective was to seed faults which were representative of the types of faults that were removed in the first place. Thus, the influence of the seeding is minimised.

- The experimental design is not representative of real inspections.

  A laboratory experiment is often a necessary step in assessing new technical proposals prior to transferring the technique to practical use. The objective was, however, to imitate the real inspection with capture-recapture as closely as possible.

- The inspection team is not representative of software developers.

  This threat is not regarded as being critical as many of the fourth-year students will go into software development quite soon. The students were also mixed with more experienced personnel by including a number of consultants from Q-Labs in the experiment.

6.3 Conducting the experiment

The inspection preparation was conducted by distributing the document to the reviewers, who were supervised to ensure that no cooperation took place. The reviewers marked items which they considered as being faults directly in the document. The inspection preparation lasted for 1-1.5 hour and the reviewers were asked to leave the room when they were finished and had handed in the inspected document. After the

inspection preparation, the documents were scrutinized and the fault data were collected. This was done to imitate the inspection meeting and data collection. The fault data were collected in tables, which are shown in Appendix A.

The inspection team reassembled the next day and the results of the inspection preparation and the imitation of the inspection meeting and data collection were discussed. The reviewers were given back their documents containing faults found which were regarded as faults, markings indicating items which were judged not to be faults and seeded faults which were not found by any reviewer. This meeting was essential for the experiment to ensure that the markings had been correctly interpreted and also to reach a consensus within the inspection team regarding what constituted a fault.

6.4 Data analysis

6.4.1 Inspection data

Some significant characteristics of the data are presented below as a summary of the data presented in Appendix A:

- Between 4 and 21 faults were reported by the individual reviewers.
- In total, 36 unique faults were found by the inspection team, i.e. 2 faults remained undetected in the document. The first group found 30 unique faults, the second group found only 25 unique faults while the third and final group found 34 unique faults.
- No fault was found by all 22 reviewers. One fault was found by as many as 20 of the reviewers, while 5 faults were found by single reviewers.

6.4.2 Estimations without a filter

The data were analysed as if being the result of one inspection group as well as three inspection groups. The results are presented in Table 1. The last column presents the relative error in percent, for example (36-38) / 38, where 36 is the estimate and 38 the correct answer.

TABLE 1. Estimates of the number of faults without a filter.

	Unique faults found	Inspection meeting	N	Error
22 reviewers	36	0	36	-5.3%
Group 1	30	0	30	-21.0%
Group 2	25	0	25	-34.2%
Group 3	34	0	34	-10.5%

The results show that the estimates of the number of faults prior to the inspection become equal to the number of unique faults found. The conclusion from the estimation is that the method gives an underestimation, which is believed to be due to the assumption of equal probabilities of finding all the faults for a single reviewer. Some faults were found by an extremely large proportion of the reviewers, while others were

not found by any reviewer, hence indicating that it is very unlikely that the assumption is valid, see Section 3.0.

It is possible to derive the probability that a fault will not be found by any of the reviewers.

- When using the Maximum-Likelihood method, the probability of a single reviewer of finding a fault is estimated, as the number of faults found by that reviewer divided by the estimate of the total fault content, i.e. $p_j = n_j / N$ for reviewer j.
- The reviewers are independent, hence non-detection probabilities for the different reviewers can be multiplied by each other.

In the experiment the probability can be derived from the fault data in Appendix A. The probability of a fault not being found is as low as $7.1 * 10^{-5}$. This probability is obtained from:

p = (1-15/30) * (1-12/30) * … * (1-10/30) * (1-14/25) * … * (1-18/25) * (1-15/34) * … *(1-10/34),

where, for example 15/30 is the probability of reviewer 1 to finding a fault and then one minus this probability is the probability that a fault is not found by the first reviewer. The probability that none of the 22 reviewers finds a specific fault is, of course, the product of the probabilities that none of the individual reviewers finds the fault. This clearly explains why the Maximum-Likelihood estimate becomes equal to the number of faults found. All the same, when studying the 22 reviewers as one group, it is known that two faults were not found by any of the 22 reviewers. This is the case as this was a controlled experiment where the faults were seeded, which is not the case when applying the method in practice.

6.4.3 Estimations with a filter

A filter of 40% was chosen, dividing the fault data into two classes as follows:

- 22 reviewers => 22 * 0.40 = 8.8, therefore class 2 becomes faults found by eight or fewer reviewers while class 1 are those faults found by more than eight reviewers. Truncation is used throughout the calculations when applying a filter.
- Group 1 => 7 * 0.40 = 2.8, which gives class 2 as the faults found by one or two reviewers and class 1 are those faults found by three or more reviewers.
- Group 2 is treated the same as group 1 since it has the same the number of reviewers, i.e. 7.
- Group 3 => 8 * 0.4 = 3.2, hence class 2 becomes faults found by three or fewer reviewers while class 1 are those faults found by more than three reviewers.

The division can be regarded as dividing the faults into an easy (class 1) and a hard class (class 2). The results of the estimation procedure are presented in Table 2.

TABLE 2. Estimates of the number of faults with a 40% filter.

	Unique class 1	Unique class 2	Meeting	N_{C1}	N_{C2}	N	Error
22 reviewers	14	22	0	14	22	36	- 5.3%
Group 1	16	14	0	16	21	37	- 2.6%
Group 2	13	12	0	13	17	30	- 21%
Group 3	14	20	0	14	22	36	- 5.3%

As expected, the estimate of faults in class 1 is equal to the number of faults found assigned to class 1, while some faults were not found in class 2, thus improving the estimate of the initial number of faults. This is at least the case when the number of reviewers is realistic. It is not very likely that anyone has the resources to allow 22 reviewers to study one document (except in experimental studies such as this).

The faults are placed in the classes according to the number of reviewers who found the different faults. An interesting aspect is to study which of the faults are placed in the same class for all three groups and, in particular, to see which faults are always placed in class 1. If many faults are placed in class 1 for all three groups, it indicates that some of the faults are more easily found than others, hence the assumption of equal probability is shown to be invalid. Two faults are placed in class 2 for all three groups, namely faults 13 and 27, while seven faults are placed in class 1 for all three groups. These seven faults are 1, 2, 5, 10, 15, 18 and 26, which, upon being studied and categorized subjectively, are easier to find than many of the others. There are of course some deviations from what one would expect, but the overlap is considerable.

An extreme filter is to place all faults found by more than one reviewer in class 1 while faults found by only one reviewer are placed in class 2. An analysis of the data based on this type of extreme filter gives the results presented in Table 3, where special consideration must be taken, since for faults in class 2 the reviewers have no fault in common. This is in accordance with the rule described in Section 5.0, i.e. by applying a multiplicative factor of 2.

TABLE 3. Estimates of the number of faults with factor 2 filter.

	Unique class 1	Unique class 2	Meeting	N_{C1}	N_{C2}	N	Error
22 reviewers	31	5	0	31	10	41	7.9%
Group 1	21	9	0	21	18	39	2.6%
Group 2	18	7	0	18	14	32	-15.8%
Group 3	27	7	0	27	14	41	7.9%

From Table 3, it can be seen that the total number of faults is slightly overestimated in three cases out of four. The multiplicative factor must, however, be determined based on the data available after the experimental study. This is further discussed in Section 7.0. It must be noted that the minimum relative error is 2.6% as the number of faults is discrete, see Table 3.

6.5 Conclusions of the experiment

It is impossible to state which type of filter is best in general, but it can be concluded that it is possible to find a filter which gives good estimates. Thus, the experiment supports the hypothesis that it is possible to improve the estimates by introducing a filtering technique when applying capture-recapture to inspections. Therefore, it is important to try to identify the best possible filter for a particular environment and type of document. Two different types of filters have been suggested, and these must be investigated further to determine an optimal filter.

The filter for each organisation must be based on experience and continuous collection of data, hence it is not possible to reuse the quantitative results of this experiment, but the filtering technique as a general method can be applied to other types of documents and environments. A more thorough study of the influence of the choice of filter based on the collected data is presented in Section 7.0.

7.0 Filter analysis

7.1 Introduction

Based on the conclusion that it is possible to find filters which improve the predictive ability of the capture-recapture method, it was decided to identify the best filters for this particular data set. The objective was to identify filters which could be used in the future in similar experiments to evaluate the technique even further. In the experiment, two different filters were used: a 40% filter and a factor 2 filter. The latter means using the faults found by a single reviewer as a filter. These values can now be improved based on the experience from the experiment.

The three groups were considered individually and some different types of filters were chosen to illustrate how the estimate changes with the filter. The filters were chosen so that the fault data were divided into two classes for all the filters. The filters were based on varying the percentage thresholds defining the different fault classes, where a low percentage value indicates that class 2 is small, i.e. faults which could be considered difficult to find.

Two possible ways of creating a realistic filter have been identified.

1. The percentage value defining the filter is chosen such that the estimate becomes equal to the correct value found in the experimental study. This can actually be done in two ways.
   - either a suitable percentage is determined for each group and the mean value of the percentage values is chosen as an appropriate filter, or
   - a line is derived by taking the mean value of the three groups for each percentage value. The filter is then chosen based on the line derived.

The first method means that a filter is chosen for each group, then a mean value is determined based on the three values derived, hence obtaining a mean percentage value which should be the best filter in some sense. In the second case, a mean value is derived for each percentage figure and a line is constructed from which the filter is determined. These approaches are illustrated in the example in Section 7.2.

2. Another method is to use a filter where faults found by only one reviewer are placed in class 2. The estimation of the total number of faults in this class can be found through a scaling factor, i.e. the multiplicative factor discussed in Section 5.0. The factor from Section 5.0 is replaced by a mean value based on the experimental study.

The percentage value or scaling factor identified is, of course, highly dependent on the number of reviewers, environment, application and type of document. Therefore, it is necessary to conduct a similar investigation to the one presented in Section 7.2 for a specific organisation with its own characteristics. The objective of the subsequent section is primarily to illustrate how a filter can be determined.

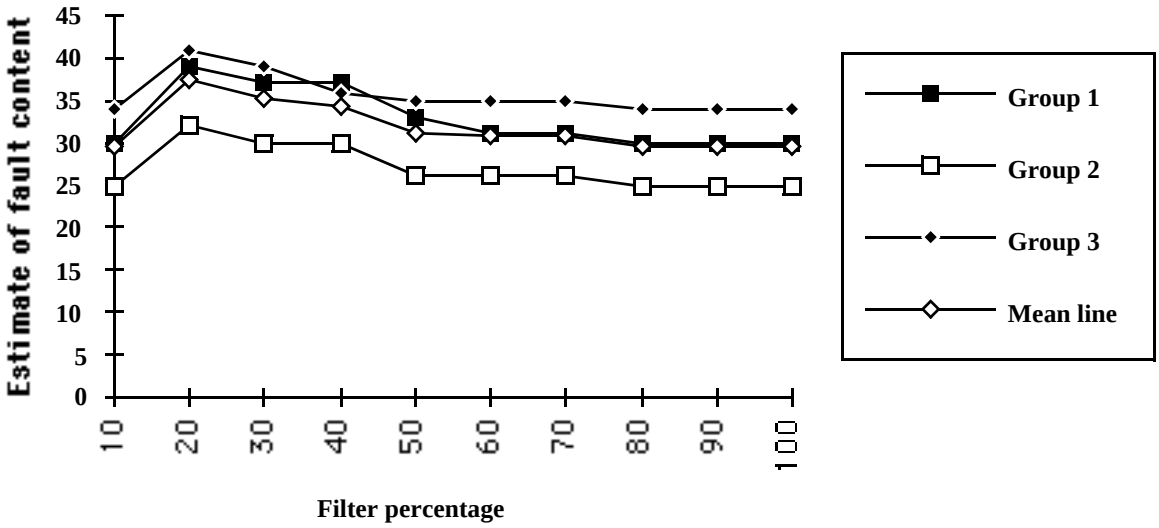
7.2 Example

The following filtering cases were investigated: 10%-100% in steps of 10%. The results for the 20% filter can be compared with those in Table 3, while 10% and 100% are equivalent to the outcome presented in Table 1, which is the analysis proposed by (Eick *et al.*, 1992) and (Vander Wiel *et al.*, 1993), where the percentage is defined such that all faults fall into either class 1 or class 2. The 40% filter is the same as that presented in Table 2, Section 6.4.3. The results are presented in Table 4 and further illustrated in Figure 4, where the total number of faults are plotted.

TABLE 4. Estimates based on different filters.

Filter	Gr. 1 Cl. 1	Gr. 1 Cl. 2	Gr. 1 Total	Gr. 2 Cl. 1	Gr. 2 Cl. 2	Gr. 2 Total	Gr. 3 Cl. 1	Gr. 3 Cl. 2	Gr. 3 Total
10%	30	0	30	25	0	25	34	0	34
20%	21	18	39	18	14	32	27	14	41
30%	16	21	37	13	17	30	20	19	39
40%	16	21	37	13	17	30	14	22	36
50%	12	21	33	6	20	26	7	28	35
60%	3	28	31	3	23	26	7	28	35
70%	3	28	31	3	23	26	4	31	35
80%	0	30	30	1	24	25	2	32	34
90%	0	30	30	1	24	25	1	33	34
100%	0	30	30	0	25	25	0	34	34

FIGURE 4. Analysis with different filters.



For the two ways of constructing the best possible filters, the following results are obtained. (cf. Section 7.1)

1. Filter based on Figure 4

   In this case the percentage value for the first method, (see Section 7.1), is estimated to be:

   • 29.4%, i.e. the best filter is determined for each separate group and the mean value is then calculated. In this case the value is the mean of 25%, 30% and 33.3%.

   Interpolation was used to determine the first and the third value, while it was necessary to take 30% for the second group as no value was above 38, and 20% which gives the best result is based on the multiplicative factor, see Figure 4. The multiplicative factor used so far is not based on experience, hence the 20% filter is not suitable to use here. The three values give a mean value of 29.4%. This value is, however, unrealistic as the number of faults is discrete, but it indicates how the filter may be chosen to be effective and it is possible to weight the faults. This implies that some of the faults found by 2-3 reviewers should be divided between the classes according to the interpolation.

   Using the second method, the value found for the filter was

   • 20%, i.e. the three lines in Figure 4 are merged into one mean line and the best filter is then determined from this line. In this case the mean line is closest to 38 at 20%, where the value is 37.3 after adding 39, 32 and 41 and dividing by 3.

   It should be noted that the percentage value of 20% is based on the scaling factor proposed in Section 5.0, and is thus not a valid percentage value. This filter can be compared with the one presented below.

2. Faults found by only one reviewer

   For group 1, 8 faults were not found and 9 faults were found by only one reviewer. Therefore, the estimate from class 2 should ideally be 17, including the 9 faults found. The estimate using the straightforward Maximum-Likelihood estimate is 21 for class 1, i.e. faults found by more than one reviewer. The multiplicative factor

hence becomes: 17/9 = 1.89, which would give the estimate 38, since 21 + 1.89 * 9 = 38. The same reasoning can be applied to groups 2 and 3. A mean value of the factor can then be derived from the outcome of the three groups. The result is (1.89 + 2.86 + 1.57) / 3 = 2.11. A factor of 2.11 should thus in the future be used in the estimation procedure, instead of the initially assumed factor 2.

7.3 Summary

The choice between the two approaches discussed in the previous section must be based on experience, and initially it is suggested that both approaches be used. It must, however, be observed that both approaches mean that one factor must be determined based on actual experience, i.e. experimental studies are needed in the specific environment and for the different types of document that are inspected, as well as for typical sizes of inspection teams.

The adoption of one of the approaches should, however, mean that fault content estimations, which can form the basis for approval or disapproval of a document, can be formulated in terms of thresholds.

8.0 Conclusions

The capture-recapture method appears to be promising in attempts to estimate the fault content of documents. As the capture-recapture method with Maximum-Likelihood estimates notoriously gives underestimated values evidenced by the experiment as well as by (Vander Wiel *et al.*, 1993), it is necessary to find methods of overcoming this problem in order to obtain reliable estimates.

The solutions presented are founded on identifying a filter based on experience. The filter must be used to divide the faults found into more than one class, and it is recommended that two classes be initially identified. The classes are defined such that some of the assumptions of the method are better fulfilled, hence the estimate is improved. Two main filters have been discussed:

- a percentage value dividing the faults into two groups based on the number of reviewers who found a specific fault,
- a filter which ensures that faults found by only one reviewer are collected into one class.

The latter approach requires that it must be possible to perform estimations even if the reviewers have no fault in common. An experience factor is proposed to overcome this problem. In the first approach, the percentage value must be based on experience.

The capture-recapture method is easily adopted, but it must be combined with a metric, i.e. an experience factor which determines the division into different fault classes. The metric must hence reflect the actual situation in a specific organisation as the assumptions of the models are probably not fully fulfilled in practice. Capture-recapture estimations combined with a filtering technique provide an important method of

performing fault content estimations throughout the software life cycle. The method will therefore provide useful support for managers and decision makers regarding the acceptance of documents in software development.

Acknowledgements

Part of this study was carried out on behalf of Telia (Swedish Telecom), and in particular the authors would like to thank Jan-Eric Johansson at Telia for letting the results be published. The authors would also like to express their gratitude to the personnel at Q-Labs and the students at Lund University who participated in the experiment. The comments from the anonymous referees have been very valuable and they have improved the quality of the paper considerably. Finally, the authors would like to acknowledge Helen Sheppard, Word for Word for improving the English.

References

Eick, S. G., Loader, C.R., Long, M.D., Votta, L.G and Vander Wiel, S. A. (1992) 'Estimating Software Fault Content Before Coding', Proceedings 14th International Conference on Software Engineering (IEEE conference), Melbourne, Australia, pp. 59-65.

Fagan, M. E. (1976) 'Design and Code Inspections to Reduce Errors in Program Development', *IBM Systems Journal*, Vol. 15, No. 3, pp. 182-211.

Gilb, T. and Graham, D. (1993) 'Software Inspections', Addison-Wesley, Reading, Massachusetts, USA, pages 471.

Vander Wiel, S.A., and Votta, L.G. (1993) 'Assessing Software Designs Using Capture-Recapture Methods', *IEEE Transactions on Software Engineering*, Vol. 19, No. 11, pp. 1045-1054.

Wohlin, C., and Runeson, P. (1994) 'Certification of Software Components', *IEEE Transactions on Software Engineering*, Vol. 20, No. 6, pp. 494-499.

Appendix A: Fault data

The reviewers have been divided into three groups by random. The results are presented in table 5-7 for the three groups separately.

TABLE 5. Faults found by reviewers in group 1.

Fault	R1	R2	R3	R4	R5	R6	R7	Sum
1	1	0	0	1	1	0	1	4
2	1	0	0	1	1	0	1	4
3	0	1	1	1	0	1	0	4
4	0	1	1	1	0	1	0	4
5	1	1	0	1	1	0	0	4
6	1	1	0	1	1	0	0	4
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	1	0	0	1	0	2
10	1	0	0	1	1	0	0	3
11	0	0	0	0	1	1	0	2
12	0	0	0	0	1	0	0	1
13	1	0	0	0	0	0	0	1
14	1	1	0	1	0	0	0	3
15	1	1	0	1	1	0	1	5
16	1	0	0	0	1	0	0	2
17	0	0	0	0	0	0	0	0
18	1	1	0	1	1	0	1	5
19	0	0	0	0	1	0	0	1
20	1	1	0	1	1	0	0	4
21	1	0	0	0	1	0	1	3
22	1	1	0	0	1	0	1	4
23	0	0	0	0	0	0	1	1
24	0	0	0	0	0	0	0	0
25	1	0	0	0	0	0	1	2
26	0	1	1	0	1	0	1	4
27	1	0	0	0	1	0	0	2
28	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0
30	0	0	1	0	0	0	0	1
31	0	0	0	0	0	0	0	0
32	0	0	0	0	1	0	0	1
33	0	1	0	1	1	1	1	5
34	0	0	0	0	0	0	0	0
35	0	0	0	0	1	0	0	1
36	0	1	0	1	1	0	0	3
37	0	0	1	0	0	0	0	1
38	0	0	0	0	1	0	0	1
Sum	15	12	6	13	21	5	10	

TABLE 6. Faults found by reviewers in group 2.

Fault	R8	R9	R10	R11	R12	R13	R14	Sum
1	1	0	0	0	1	1	0	3
2	1	0	0	1	1	0	1	4
3	1	0	0	0	0	0	1	2
4	1	0	0	0	0	0	1	2
5	1	0	1	0	0	0	1	3
6	1	0	0	1	0	0	0	2
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	1
9	1	0	0	1	0	0	1	3
10	1	0	1	0	0	1	1	4
11	1	1	0	1	0	0	1	4
12	0	0	1	1	0	1	0	3
13	0	0	0	0	0	1	0	1
14	0	0	0	0	0	0	1	1
15	1	0	0	1	1	1	1	5
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	1	1	1	1	1	1	1	7
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	1	1
21	0	1	0	0	0	0	0	1
22	0	0	1	0	0	0	0	1
23	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
25	1	0	0	0	0	1	1	3
26	1	1	1	1	0	0	1	5
27	1	1	0	0	0	0	0	2
28	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	1	1
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	0	0	1	0	0	0	1	2
34	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0
36	0	0	1	0	0	1	1	3
37	0	0	0	0	0	0	0	0
38	0	1	0	1	0	0	1	3
Sum	14	6	8	9	4	8	18	

TABLE 7. Faults found by reviewers in group 3.

Fault	R15	R16	R17	R18	R19	R20	R21	R22	Sum
1	1	1	0	1	0	1	0	0	4
2	1	0	0	0	1	1	1	0	4
3	1	0	0	0	1	1	0	0	3
4	1	0	0	0	1	1	0	0	3
5	1	1	0	1	1	1	1	1	7
6	1	0	0	0	1	0	0	0	2
7	0	1	0	0	0	0	0	0	1
8	0	1	0	1	0	0	0	0	2
9	0	0	0	1	0	0	1	1	3
10	0	1	1	1	0	1	0	0	4
11	1	0	0	1	0	0	1	0	3
12	0	1	1	1	0	1	1	1	6
13	0	1	0	1	0	0	1	0	3
14	0	0	0	1	0	1	1	1	4
15	1	1	0	1	0	1	1	1	6
16	1	0	0	0	0	0	0	0	1
17	0	1	1	1	0	0	1	1	5
18	1	1	1	1	1	1	1	1	8
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	1	0	1
21	1	1	1	0	0	0	1	0	4
22	1	1	1	0	0	0	1	0	4
23	1	0	0	1	1	0	0	0	3
24	0	0	0	0	0	0	0	0	0
25	1	1	0	1	1	1	0	0	5
26	0	1	1	1	0	1	0	1	5
27	0	1	1	0	0	0	0	0	2
28	0	0	0	1	0	0	0	0	1
29	0	1	0	0	0	0	0	0	1
30	0	0	0	1	0	0	1	0	2
31	0	0	0	0	0	0	0	0	0
32	0	0	0	1	0	0	0	1	2
33	1	0	0	1	1	0	1	0	4
34	0	1	0	0	0	0	0	0	1
35	0	1	0	0	1	0	0	0	2
36	0	0	0	1	0	0	0	1	2
37	0	0	0	0	0	0	0	0	0
38	0	0	0	1	0	0	0	0	1
Sum	15	18	8	21	10	12	15	10	