C. Wohlin, H. Petersson, M. Höst and P. Runeson, "Defect Content Estimation for Two Reviewers", Proceedings of the 12th International Symposium on Software Reliability Engineering, pp. 340-345, Hongkong, China, December 2001.

Defect Content Estimation for Two Reviewers

Claes Wohlin
Dept. of Software Engineering and Computer Science, Blekinge Institute of Technology
Box 520, SE-372 25 Ronneby, Sweden
E-mail: Claes.Wohlin@bth.se

Håkan Petersson, Martin Höst and Per Runeson
Dept. of Communication Systems
Lund University
Box 118, SE-221 00 Lund, Sweden
E-mail: (hakanp, martinh, perr)@telecom.lth.se

Abstract

Estimation of the defect content is important to enable quality control throughout the software development process. Capture-recapture methods and curve fitting methods have been suggested as tools to estimate the defect content after a review. The methods are highly reliant on the quality of the data. If the number of reviewers is fairly small, it becomes difficult or even impossible to get reliable estimates. This paper presents a comprehensive study of estimates based on two reviewers, using real data from reviews. Three experience-based defect content estimation methods are evaluated vs. methods that use data only from the current review. Some models are possible to distinguish from each other in terms of statistical significance. In order to gain an even better understanding, the best models are compared subjectively. It is concluded that the experience-based methods provide some good opportunities to estimate the defect content after a review.

1. Introduction

Quality control throughout the software life cycle is essential. Reviews are one technique that provides opportunities for continuous evaluation of the quality [7],[1]. Thus, it is important to try to make the best possible use of the information from reviews, for example, to have methods to estimate the remaining defect content and not just focusing on the defects that were found. This type of estimate can be a valuable tool for decision-making, and they are the main concern of this paper. Several different estimation methods are evaluated in an experiment. Experimentation is a valuable tool in evaluating different methods, techniques and tools versus each other in software engineering. Experimentation in software engineering is treated in more detail in [16].

Figure 1 illustrates one possible way of incorporating the defect content estimations in the review process and the decision-making. The objective is not to show a complete review process, but rather to focus on how defect content estimation may become a natural part of the review process. First, the individual reviewers prepare and try to find defects. The next step depicted is the pooling of the individual findings. Data has to be collected on an individual level, but it is beneficial to pool the data before the meeting. Firstly, it can be decided whether a review meeting is needed or not and secondly, the pooled list of defects, together with a defect content estimate provide valuable input to the meeting. The pooling of the defects prior to the meeting is also believed to improve the individual preparation. Either when it is decided to not have a meeting or after the meeting the defect content estimation is one important input for the decision-makers to determine the next actions.

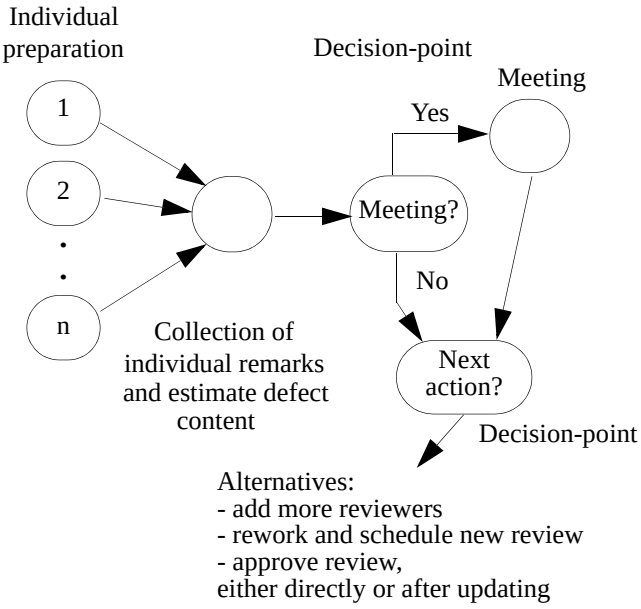

Figure 1. Potential use of defect content estimations in the review process.

Capture-recapture methods [5] and curve fitting methods [15] have been proposed for estimating the defect content in reviews. Experience-based methods have also been investigated in [14], [13] and [12]. This paper provides an empirical study of two reviewers with data from software reviews, i.e. real data although collected in a laboratory environment to enable control of the study.

A study of using different capture-recapture models for different number of reviewers is presented in [2]. It is noted in [2] that the estimation problem becomes particularly difficult for few reviewers, for example, some of the models are unable to produce estimates when there is no overlap between reviewers in terms of defects found. An additional problem is that the uncertainty in the estimate increases in the two reviewer case due to the small number of reviewers. Estimation based on two reviewers is also addressed in [6] using simulation.

Estimation of the remaining defect content after a review can be made in different ways. Three different types of methods are identified. The methods can briefly be described as:

- *Experience-based methods*, i.e. methods that in some way use historical data [8] to estimate the defects content. These types of methods have been addressed in, for example, [14] and [13].
- *Capture-recapture methods*, i.e. methods that use the degree of overlap between reviewers to estimate the defect content [5]. These methods have their origin in biology where they are used for population estimations [11].
- *Curve fitting methods*, i.e. methods that plot the review data from the reviewers in a predetermined way. Based on the plot a mathematical function is fitted and then used to estimate the defect content [15]. The curve fitting models are not evaluated in the paper, since they are not suited for the case with two reviewers.

The objective of this paper is threefold:

- To evaluate experience-based defect content estimation models and in particular study on which level inspection effectiveness experience should be reused.
- To investigate how good experience-based models are in comparison to capture-recapture models for estimation.
- To identify which models are best suited for estimation of the defect content when having only two reviewers.

This paper is organized as follows. In Section 2, experience-based methods are discussed and some issues for evaluation of the methods are introduced. Section 3 briefly presents capture-recapture methods and curve fitting methods. An experimental study of defect estimation methods is presented in Section 4. Finally, conclusions are presented in Section 5.

2. Experience-based methods

The main advantage of using an experience-based method is that it tries to capture previous experience using historical data. This implies that although it adapts to the current review, it is still based on the experience in the organization. For experience-based methods data from previous reviews is used to perform the estimation in the current review, while the other two types of methods only use data from the current review.

Other studies [14], [13] and [12] have evaluated experience-based methods, but they have not addressed the problem of only having two reviewers. The experience-based methods are believed to be particularly valuable for two reviewers, since reviews with only two reviewers are especially sensitive to the performance of the individuals at that specific occasion.

Moreover, the focus in this paper includes evaluating how experiences regarding review effectiveness should be reused and calculated. The effectiveness is here defined as the number of unique defects found out of the total number of defects in the artifact being inspected. The effectiveness is calculated for each individual reviewer. The actual use of the calculated effectiveness may however be used in different ways, which is discussed next

2.1 Use of review effectiveness

After having derived the review effectiveness, it is necessary to determine a suitable way of using the measures. Several different opportunities exist. Should it be personal? Should it be defined for a certain group of people? Should it be a mean value that can be used for all reviewers? Thus, there exist at least three different alternatives of how to capture and use experience in terms of review effectiveness.

- Individual experience base (denoted I)
  The first approach is that each person's data is reused on an individual basis. In other words, the reviewers have individual experience bases. In this way the estimation is done based on what the individuals participating in the review have achieved in the past in terms of defect detection at reviews.
- Group experience base (denoted G)
  Another alternative is to try to treat people in terms of groups, where the grouping is based on, for example, background and experience. The groups can be determined using Principal Component Analysis (PCA) [10]. In the empirical study in Section 4.3, reviewers are grouped based on a PCA of a set of reviewer profile measures that were collected in questionnaires.
- Average reviewer (denoted A)
  Finally, a third alternative is to assume that all reviewers behave in a similar way. In this case, a joint experience base is created for all reviewers, and an average review effectiveness is used.

These three alternatives are compared and evaluated empirically in Section 4.3.

3. Capture-recapture methods

The capture-recapture models have their origin in biology, where they are used to estimate animal populations. This also means that some of the notations are primarily adapted for this type of application. Capture-recapture models were first used for defect content estimation from reviews in [5]. These type of models rely on using a statistical estimator, where the estimator uses the degree of overlap in defect detection between reviewers to perform the estimation. Different capture-recapture models use different assumptions regarding reviewers and defects. Reviewers may have the same or different capability of finding defects, and the defects themselves may be equally difficult to find or some may be more difficult than others are. Thus, capture-recapture methods can be divided into four model types. These are:

1. Reviewers are assumed to have the same capability to find defects, and the different defects are found with the same probability. This type of model is denoted M0.
2. Reviewers are assumed to have the same capability to find defects, though different defects are found with different probabilities. This type of models is denoted Mh (variation by heterogeneity), since it takes the detection probabilities into account but not the reviewers' capabilities.
3. Reviewers are assumed to be different, i.e. they have different capabilities to detect defects, and all defects are found with the same probability. This type of models is denoted Mt (variation by time), since it takes the reviewers' abilities into account but not the detection probabilities.

4. Reviewers are assumed to be different, i.e. they have different profiles for detecting defects, and different defects are found with different probabilities. This type of model is denoted Mth.

These models are illustrated graphically in [2]. Statistical estimators can be applied to the different types of models. Statistical estimators for each type of model are presented in Table 1.

Table 1. Capture-recapture estimators.

Reviewer capabilities	Detection probabilities	
	Equal	**Different**
Equal	M0: Maximum-likelihood [11] (denoted M0-MLE)	Mh: Jackknife [11] (denoted Mh-Jack-knife)
Different	Mt: Maximum-likelihood [11] (denoted Mt-MLE) and Chao [3] (denoted Mt-Chao)	Mth: Chao [4] (denoted Mth-Chao)

These five models obtained when using these estimators are compared with the best of the experience-based models in Section 4.4.

4. Experimental study

4.1 Overview

The objective of the experiment is to compare a number of models to estimate the defect content. The data, to enable the comparison, is obtained from a series of reviews of C programs. Thus, the first part is a data collection step where reviews are conducted. The data is then input to the experience-based models, where the different parameters (primarily use of the calculated effectiveness) are varied according to the descriptions in Section 2. The models are evaluated using statistical methods and the best models are determined with statistical significance. The best experienced-based models are then compared with the capture-recapture models. Statistical methods are applied in order to determine which models are the best. The models that cannot be separated statistically, when comparing the mean values, are evaluated in a subjective evaluation where mean value, variation and maximum error of the models are studied. Based on the subjective evaluation, a set of models can be identified as being most appropriate when estimating the defect content for two reviewers.

4.2 Data collection

Review objects. The software programs reviewed are defined in [9] and developed according to the Personal Software Process (PSP) by a Ph.D. student with four years of industrial experience, although not in C programming. Programs 3A, 4A, 5A, 6A and 7A according to the PSP course scheme are selected for review. Programs 1A and 2A were not selected due to risks for initialization effects, i.e. basic programming mistakes are often made in the first couple of programs before the students get more used to programming. A total number of five review objects were selected in the experiment as a trade-off between reasonable workload for the participants and a sufficient number of reviews for the study. The original experiment is discussed and reported in [13]. Although the same data is used, different issues are evaluated. For example, the focus in [13] is not on two reviewers, and the issues raised for the experience-based methods are novel work reported here. The actual number of defects in the programs and the size of the programs can be found in Table 2.

Table 2. Defect and size data

Program	3A	4A	5A	6A	7A	Total
Defects	22	16	16	35	20	109
LOC[a]	190	113	85	304	208	900

a. These figures include new and reused code.

The defects are real defects introduced during software development. The code version available directly after coding was stored, i.e. it contains the defects later found in, for example, compilation and testing. Functional faults as well as cosmetic ones, like misspelled comments, are counted as defects. The majority of the defects are of syntax type, and the second largest class of defects is of functional type, such as logic, pointers and loops.

Participants. Eight reviewers with various industrial and academic backgrounds conducted the reviews. Four reviewers were Ph.D. students (B, D, E and G), three were software engineering seniors (C, F and H) and one was a recently graduated software engineer (A). The reviewers were categorized according to a reviewer profile scheme. The profiling includes six aspects: state-of-the-art knowledge in software engineering, as well as experience in project work, coding in C, reviews in general, reviews of C and problem domain (PSP). The participants rated their experience on a scale from 1 to 4, where 1 means no knowledge or experience while 4 means more than three years experience. The profiling data and the total time spent by each reviewer for the three programs can be found in [13]. Although, the data is on an ordinal scale and we only have 8 data points and six participants, we have chosen to conduct a PCA[1]. The intention is to enable division into similar groups based on documented experience rather than our knowledge of the individuals (Ph.D. students, software engineering seniors and recently graduated software engineer). The analysis resulted in three groups: Group 1 with reviewer C, F and H; Group 2 with reviewers B, D and G; and Group 3 with reviewers A and E.

It is noticeable that Group 1 corresponds to the senior software engineers, and Group 2 consists of three out of four Ph.D. students. The last group (Group 3) contains the recently graduated software engineer and the one of the Ph.D. students. Thus, the identified groups are close to the

1. All statistical analysis in the paper are conducted using StatView 5.0 on Macintosh.

grouping obtained by simply looking at the background in general.

The division into groups is used when creating experience bases based on groups, i.e. the groups denoted G in Section 2.1.

Operation. Each of the reviewers reviewed three programs out of the set of five. The programs were randomly allocated to the reviewers. Five reviewers each reviewed programs 3A-6A, and four reviewers reviewed program 7A. This means that in total 24 individual reviews were conducted.

The reviewers were given a package for each program containing:

- Code review process with:
  - corresponding code review checklist
  - time recording log
  - defect recording log
- C program code listing
- Requirements specification from the PSP book [9]
- Program design (textual)

Based on the five reviewed programs and the number of reviewers for each program, it is possible to create 46 pairs of reviewers, i.e. all possible pairs of reviewers are created. Obviously, this means that the pairs are not independent. It is, however, not viewed as a major threat to the study, since the primary interest is a relative comparison between different methods and possibilities.

Threats. The study includes threats on two aspects. First, it is threats regarding the data collection, i.e. the actual review and the objects reviewed. The second aspect is concerned with threats on how the model are compared. Moreover, threats may be divided into four types of validity: internal, external, construct and conclusion. External validity is concerned with generalization of the results. It has to be ensured that the observed relationship is causal; this is referred to as internal validity. The construct validity is related to that the experiment should reflect the concept to be studied. Finally, the conclusion validity is concerned with the relationship between the treatment and the outcome in the experiment. Threats to experiments are discussed in more detail in, for example, [16]. These four types of validity should be addressed for both aspects of threats.

1. Data collection procedure
   - External validity: The code has been written by a novice C programmer and hence the programs and defects may not be representative. The reviewers are mostly not industrial engineers, and they may not be representative of that population. These threats are not regarded as critical as they affect reviews in a similar way and hence the comparisons made between defect estimation models should not be affected to any major extent.
   - Internal validity: The programs were not reviewed in random order and hence there is a risk that the reviewers either improve or get tired as the conduct the reviews. This threat is partly addressed by that most of the reviewers did not perform all reviews on one day. The reviewers knew that they took part in an experiment, and this may have affected their behaviour. It is, however, not likely that this will affect the comparison of estimation models.
   - Construct and conclusion validity: These types of validity are primarily concerned with model comparison.
2. Model comparison
   - External and internal validity: Both the external and internal validity are primarily concerned with the data collection procedure, see above.
   - Construct validity: These threats are primarily concerned with the design of the study and social effects, such as hypothesis guessing. The former is not believed to be a problem. The reviews are conducted individually and the data is collected without the subjects knowing exactly how the different models should be compared. Moreover, the latter is not an issue since the model comparison is made based on the data collected and the individuals involved in the study are unable to influence this comparison.

     One specific threat observed is that the efficiency measure may be influenced by that reused code is reviewed for some of the programs. This will not influence the effectiveness measures since they are defined based on the defect content, but it may influence the efficiency measure since the review time may become shorter if the reviewers recognize that code has been reused.
   - Conclusion validity: The conclusion validity is primarily related to two concerns in this study, namely violations of assumptions of the statistical methods and the risk that since several hypotheses are evaluated and some of them may turn out significant by chance although there is no true relationship. The violations of assumptions are addressed by applying both non-parametric and parametric tests. The significant results are i line with the expectations, which indicates that the identified significant results are true findings.

In summary, the objective has been to address and consider the threats at the start of the study to ensure that the results are trustworthy. It is clear that threats exist, but they are not considered to be a major problem in the study.

4.3 Evaluation of the experience-based methods

The evaluation is based on the absolute relative error of the estimates. This is possible since we know the total number of defects in the programs after compilation and thorough testing. It should also be added that there is no preconceived opinion of which of the models that are best and hence all hypotheses are evaluated using two-sided tests and a significance level of 5%.

The null hypothesis is that there is no significant difference between the different alternatives of using the effectiveness. The three alternatives are individual, group

and average, see Section 2.1. The alternative hypothesis is that there is a difference between at least some of the pairs.

If the data is normally distributed, parametric statistical methods may be used. Most of the parametric methods are, however, fairly robust, and hence they may sometimes be used although the data is not normally distributed. To evaluate the robustness, a Kruskal-Wallis test, which is a non-parametric test, was applied and the result from it was compared with that of an ANOVA test. The results were the same and it was decided to use parametric tests. The main reason for aiming for parametric tests is that for a significant ANOVA test, it is possible to apply a Fisher's PLSD (Protected Least Significant Difference) test. This test evaluates all possible pairwise comparisons in a similar way as a t-test, although it is derived to be used after a significant ANOVA test has been obtained. If there is no statistical significance according to the ANOVA test, there is no statistical difference and hence, there is no basis to apply Fisher's PLSD.

It should also be noted that the result may be a result of the limited number of data points to derive the experience bases. For example, on the individual level, the experience base is based on two reviews since the estimation is done in the third. This is a direct result of that each reviewer conducted three reviews. For the group and in particular for the average case, we have more data points since they are derived from either a group of reviewers or all available reviewers. Thus, future work has to be directed towards further evaluations of these three opportunities.

In summary, we can conclude that it is not possible to say with any statistical significance whether an experience-based model should be based on individuals, groups or simply an average of all reviewers.

To evaluate the experience-based models further, the next step is to compare the three experience-based models with some of the available capture-recapture methods.

4.4 Experience-based versus capture-recapture

One problem with the capture-recapture methods is that they fail if there is no overlap between reviewers in terms of which defects they have found. This situation occurs sometimes, and it occurs more frequently in the case of few reviewers. Thus, this becomes a particular problem when investigating the case of two reviewers. For the 46 pairs compared, no overlap occurred in seven cases. These seven cases of non-overlap were removed to enable a fair comparison between the experience-based methods and the direct methods. It may be argued that we in fact should use models that produce estimates in all cases independently of the overlap, but in order not to favour the experience-based methods, it was decided to remove the cases with no overlap.

Next, the three experience-based models and the direct methods are evaluated for the data sets where there is an overlap. The results are presented in Figure 2. On the x-axis, A-C are the experience-based models, denoted EBM-I, EBM-G or EBM-A for Individual, Group or Average, see Section 2.1. D-H are the capture-recapture models, see Section 3. The only model, which is possible to significantly differentiate from the others, is the Mth-Chao estimator that performs worse than the others. This is not so surprising. The model is trying to take both differences in capability between reviewers into account as well as the differences between defects in terms of difficulty to detect them. This means that Mth-Chao needs more data than the other models to provide good and stable estimates. For only two reviewers, the data is highly dependent on single data points and this causes the Mth-Chao model to fluctuate a lot.

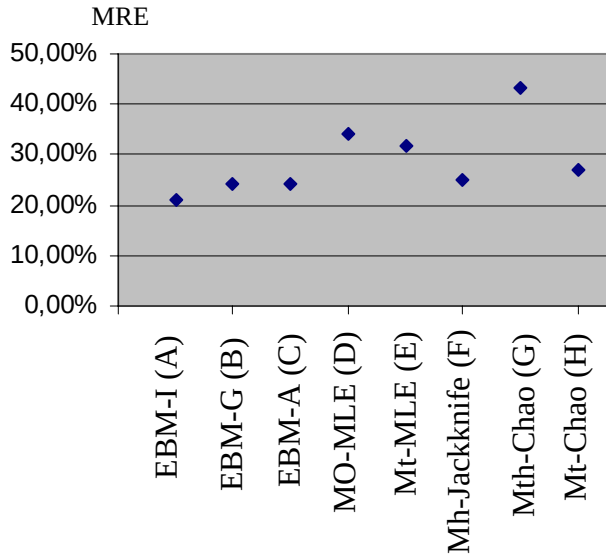

Figure 2. Evaluation of experience-based models and direct models.

From Figure 2, it can be seen that the mean absolute relative error in percentage varies between 22 and 35 percent for the remaining seven models (all except Mth-Chao), which we cannot differentiate with a statistical significance. Moreover, the standard deviation is in the range of 16-23 percent, and the maximum error by the models varies between 63 and 113 percent. In addition, it is worth noting that the best value of these three measures is not provided by the same model, hence it is difficult to objectively determine one superior model.

The evaluation results in that three experience-based models and four capture-recapture models are considered to be the best.

To evaluate if it was at all possible to subjectively differentiate between these seven models, a subjective evaluation was performed by the authors (as experts on defect content estimations using this type of models). This is motivated by that the statistical methods are unable to weigh different aspects of the models, for example, mean value vs. variation vs. maximum error in the estimation. In particular, it was also noted that different models were best for different measures as stated above. The authors have no vested interest in one model being superior than the others, so a fair evaluation was anticipated.

From the subjective evaluation, it was possible to identify three models that the evaluators agreed to be the three best models, although there was no agreement regarding the order of these three models. This is due to that

the evaluators have valued different aspects (mean, variation, worst case and so forth) of the models differently concluded that the models in the top trio are worth further studies. The models are: 1) experience-based model using an individual experience base (I) or 2) experience-based model using an average for all reviewers (A), and 3) the Mh-Jackknife model. These three models are the main candidates to use when making defect content estimations from reviews with only two reviewers. This result is partly contradicted by a simulation study [6], which concludes that the Mt-Chao model is suitable for two reviewers. This model was close to the first group in the study presented here, and it is probably wise to include the Mt-Chao model in the further studies.

5. Conclusions

In this paper, three experience-based models and five capture-recapture models have been compared. It was not possible to differentiate using statistical methods which of the experience-based approaches that is best.

In the comparison between different experience-based models and capture-recapture models, it was only possible to show with a statistical significance that one of the models were not as good as the others were. To differentiate between the remaining seven models, a subjective evaluation was conducted individually by the four authors. The evaluation resulted in that three models seemed superior to the others: two experience-based models and the Mh-Jackknife model. It is notable that the subjective evaluation is able to judge different comparison criteria (mean value, variation and maximum value), which the statistical tests are unable to.

Based on the evaluation, it is concluded that defect content estimations for two reviewers are difficult due to the dependence on individual data points. This does, however, not mean that it is impossible to make estimations; it does mean that we may have to accept that the accuracy of the estimates cannot be expected to be very good. On the other hand, the study shows that it is possible to find some models that behave better than others do. The three best models identified had a mean absolute relative error of around 20% and a standard deviation also around 20%. The three models identified out of the set of eight models should be studied further.

The study has shown that although difficult, it is possible to perform defect content estimations for two reviewers. In particular, the use of experience-based models seems to be a good approach when having few reviewers. Thus, it is concluded that experience-based models for defect content estimations are worth further studies, including both replicated experiments and real world applications.

Acknowledgment

The authors would like to thank Thomas Thelin and Dr. Anders Wesslén for valuable comments of an earlier version of the paper.

References

1. A. F. Ackerman, L. S. Buchwald and F. H. Lewski, “Software Inspections: An Effective Verification Process”, IEEE Software, Vol. 6, No. 3, pp. 31-36, 1989.
2. L. Briand, K. El Emam, B. Freimut and O. Laitenberger, “A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content”, IEEE Transactions on Software Engineering, 26(6), pp. 518-539, 1999.
3. A. Chao, “Estimating Population Size for Sparse Data in Capture-Recapture Experiments”, Biometrics, Vol. 45, pp. 427-438, 1989.
4. A. Chao, S. M. Lee and S. L. Jeng, “Estimating Population Size for Capture-Recapture Data when Capture Probabilities Vary by Time and Individual Animal”, Biometrics, Vol. 48, pp. 201-216, March 1992.
5. S. Eick, C. Loader, D. Long, L. Votta and S. Vander Wiel, “Estimating Software Fault Content Before Coding”, Proceedings 14th International Conference on Software Engineering, pp. 59-65, 1992.
6. K. El Emam and O. Laitenberger, “Evaluating Capture-Recapture Models with Two Inspectors”, to appear in IEEE Transactions on Software Engineering.
7. M. E. Fagan, “Design and Code Inspections to Reduce Errors in Program Development”, IBM Systems Journal, Vol. 15, No. 3, pp. 182-211, 1976.
8. N. Fenton, and S. L. Pfleeger, “Software Metrics: A Rigorous & Practical Approach”, 2nd edition. International Thomson Computer Press, Cambridge, 1996.
9. W. S. Humphrey, “A Discipline of Software Engineering”, Addison-Wesley, 1995.
10. S. K. Kachigan, “Statistical Analysis: An Introduction to Univariate & Multivariate Methods”. Radius Press, New York, 1986.
11. D. Otis, K. Burnham, G. White and D. Anderson, “Statistical Inference from Capture Data on Closed Animal Populations”, Wildlife Monographs, No. 62, October 1978.
12. H. Petersson and C. Wohlin, “An Empirical Study of Experience-based Software Defect Content Estimation Methods”, Proceedings 10th International Symposium on Software Reliability Engineering, pp. 126-135, Boca Raton, Florida, USA, November 1999.
13. P. Runeson and C. Wohlin, “An Experimental Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections”, Empirical Software Engineering: An International Journal, Vol. 3, No. 4, pp. 381-406, 1998.
14. 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.
15. C. Wohlin and P. Runeson, “Defect Content Estimations from Review Data”, Proceedings 20th International Conference on Software Engineering, pp. 400-409, 1998.
16. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, “Experimentation in Software Engineering – An Introduction”, Kluwer Academic Publishers, Boston, MA, USA, 2000.