

Software Defect Content Estimation: A Bayesian Approach

Achin Jain,
SITE, University of Ottawa
800 King Edward Avenue
Ottawa, ON K1N 6N5, Canada

Alok R. Patnaik
Tecsis Corporation,
200-210 Colonnade Road,
Ottawa K2E 7L5, Canada

Pulak Dhar, Vineet Srivastava
Cistech Limited,
30 Concourse Gate, Unit 35
Ottawa, ON K2E 7V7, Canada

ABSTRACT

Software inspection is a method to detect errors in software artefacts early in the development cycle. At the end of the inspection process the inspectors need to make a decision whether the inspected artefact is of sufficient quality or not. Several methods have been proposed to assist in making this decision like capture recapture methods and Bayesian approach. In this study these methods have been analyzed and compared and a new Bayesian approach for software inspection is proposed.

All of the estimation models rely on an underlying assumption that the inspectors are independent. However, this assumption of independence is not necessarily true in practical sense, as most of the inspection teams interact with each other and share their findings. We, therefore, studied a new Bayesian model where the inspectors share their findings, for defect estimate and compared it with Bayesian models in the literature, where inspectors examine the artefact independently. The simulations were carried out under realistic software conditions with a small number of difficult defects and a few inspectors. The models were evaluated on the basis of decision accuracy and median relative error and our results suggest that the dependent inspector assumption improves the decision accuracy (DA) over the previous Bayesian model and CR models.

I. INTRODUCTION

Software inspection (Ebenau and Strauss 1994; Gilb and Graham 1993) is a method to detect faults in software artefacts early in the development cycle (Pettersson *et al.* 2004). It was introduced by Fagan (1976) and after that it has become a vital part of the software development process. During software inspection a software artefact is examined for errors by a team of inspectors and all the errors detected are removed. A typical inspection team consists of 3-4 inspectors with varied abilities. It has been shown that a defect that leaks to the next step in a software development will cost at least 10 times more to detect and correct (O'Neill 1997). Therefore, it is crucial to detect and remove errors as early as possible in the software development life cycle.

It has been reported that only 52 percent of the re-inspection decisions were correct (Barnard *et al.* 2003). Therefore, a correct re-inspection decision is of utmost importance. The decision of whether to re-inspect or not depends on the estimation of the total number of defects in the artefact. Several methods have been proposed to assist in making a correct re-inspection decision such as capture recapture (CR) models. All of these methods fall into the category of Defect Content Estimation Techniques (DCETs) that try to estimate the total number of defects in a document based on the number of defects detected during inspections and then the re-inspection decision can be made.

II. MOTIVATION

There are different kinds of defects present in a piece of software of varying degree of difficulty. The defects can be broadly classified into two categories, minor, defects that are easy to find and major, defects that are hard to find. For example, a defect that would cause the system to fail to satisfy a requirement would be classified as major; all others would be classified as minor (e.g., typographical errors, minor standards violation) (NASA, 1993). Though the number of major defects is less, they are the ones that cause the most of the damage. Therefore, we are primarily interested in estimating the number of major defects, which have a strong impact on product quality and functioning. Hence, the accuracy of the defect content estimation techniques (DCET) is an important issue for their practical application in the industry.

The defect estimation problem is similar to the problem of estimating animal abundance in biology and wildlife research using a technique called capture-recapture. Eick *et al.* (1992) first applied the concept of capture-recapture to software inspections. Objective empirical evaluation of CR models started with the study of Wohlin *et al.* (1995). However, this study was conducted with non-software engineering documents. Subsequent work used software engineering artefacts (Miller 1998). Other researchers considered the incorporation of Bayesian methods to estimate defect content (Basu and Ebrahimi 1998 & Gupta, 2003), performed further evaluations of CR models (Thelin and Runeson 1999a) and

evaluated their applicability to perspective-based reading (Thelin and Runeson 1999b).

It has been shown that the CR models fail under realistic software conditions (Briand *et al.* 1998, Briand *et al.* 2000, El Emam *et al.* 2001, Gupta 2003) and several modifications have been suggested to improve the results. Gupta *et al.* (2003) proposed the use of subjective estimates to estimate the population size. The basic concept behind the subjective estimates is to ask inspectors after an inspection to estimate the percentage of defects in a document they believe they have actually found (El Emam *et al.* 2001). Combining this information with a Bayesian DCET, one can estimate the total number of defects in a document. In many cases of population studies, prior information is available about the population size. Gupta (2003) has incorporated the prior information using a β -distribution and derived Bayesian estimators for the population size.

The models studied so far are based on the assumption of independence among inspectors. This assumption is not necessarily true in practice where inspectors may exchange notes. Therefore, in our research we try to study the effect of dependence among inspectors on the defect estimate.

In this paper we focus on the concepts rather than giving detail mathematics, mainly due to lack of space. Next section describes the concept of capture-recapture technique. Sections IV and V describe the Bayesian models for independent and dependent inspection processes. The research methods are discussed in Section VI, followed by results in Section VII. The summary of the work and the future prospects are outlined in Section VIII.

III. THE CONCEPT OF CAPTURE-RECAPTURE

In capture recapture method to estimate animal abundance, animals are captured, marked, and then released on a number of trapping occasions. A marked animal that is caught at a subsequent trapping occasion is said to be recaptured. The number of marked animals that are recaptured allows one to estimate the total population size based on the overlap. As an example, suppose one wants to estimate the size N of a population. Let n_1 animals are captured on first day. These animals are marked and released into the population. After allowing some time for the marked and unmarked animals to mix, a second trapping occasion is performed on another day. On this day, suppose n_2 animals are captured. Let this sample of n_2 animals consists of m_2 animals bearing a mark (animals captured on both days) and $n_2 - m_2$ animals without a mark (newly captured animals). Assuming that the ratio of marked to total animals in the second sample is equal to the ratio of marked to total animals in the entire population, the so-called Lincoln-Peterson Estimator for the number of animals in the

population can be derived as (Seber 1982 and White *et al.* 1982),

$$\hat{N} = \frac{n_1 n_2}{m_2}$$

Applying the same principle to software inspections, each inspector is considered as a trapping occasion and the defects as the animals. Each inspector reads the software artefact and tries to find the defects, i.e. draw independent samples from the population of defects. Based on the overlap of defects amongst inspectors, one can estimate the total number of defects in a software artefact and hence the remaining defects can be calculated. Taking into account this number of remaining defects, one can decide on a more objective basis whether the software artefact has to be re-inspected.

IV. THE BAYESIAN MODEL: INDEPENDENT INSPECTION

Ananda (1997) described a Bayesian model to estimate the population of mountain sheep and this model was used by Gupta (2003) for software inspections. The difference between a normal capture recapture and a Bayesian model is that the latter allows incorporating the prior knowledge directly into statistical analysis. The prior function incorporates the experience of the inspectors in terms of a β -distribution. The likelihood function is determined from the inspection data i.e. detection of defects by various inspectors. The product of the likelihood function and the prior distribution is minimized to estimate the total defect content. The selection of the prior distribution is crucial in the Bayesian analysis.

V. BAYESIAN MODEL: DEPENDENT INSPECTION

The assumption of independent inspector is not necessarily true all the time. We consider the dependence among inspectors examine the improvements to the defect estimate. We follow the model described by Basu and Ebrahimi (1998), Basu and Ebrahimi (2001), and Basu (2003), which incorporates the dependence among inspectors. This model also falls into the broad category of capture recapture methods where defects is considered as animals and the inspectors are considered as trapping occasions and the task of estimating the number of defects is similar to estimating the animal population.

VI. RESEARCH METHOD

There are different factors that can have a strong impact on the performance of an estimator namely, the number of inspectors, the number of defects and their degree of difficulty and the degree of dependence among the inspectors.

Number of Inspectors and Their Abilities

It is expected that more inspectors imply more errors being detected and hence a better estimate of the total number of

defects. However, employing a large number of inspectors is not feasible for practical and economical reasons. In studies dealing with the biological application of capture-recapture models, use of five trapping occasions (equivalent to five inspectors in software engineering) is suggested, though a number of 7 or 10 was found more appropriate (Otis *et al.* 1978, Briand *et al.* 2000).

Apart from the number of inspectors the other factor that affects the performance of an estimator is the defect detection abilities of the inspectors. Therefore, we vary the inspector abilities from 0.1 (novice) to 0.9 (expert).

Number of Defects and Their Degree of Difficulty

In practice, the number of defects that seriously affect a software, i.e. major defect is low and most of the CR models do not work well when the number of defects is low (Briand *et al.* 1998, Briand *et al.* 2000, El Emam *et al.*, 2001, Gupta 2003). Our focus is to test models that work well with low number of defects and we have considered a small defect population ranging from 10 to 30. Since we are interested in major defects, a degree of difficulty of 0.1 (very hard to find) and 0.4 (moderately difficult to find) is used in the simulations.

Dependence Among Inspectors

In our research we want to study the impact of dependence among inspectors on the accuracy of the estimate. Software inspection literatures have not addressed the issue of dependence so far and they assume that the inspectors work independently. However, this assumption is not necessarily true all the time, since inspectors can share their findings during informal or formal talks and hence can affect the accuracy of the result. In our study we use two levels of dependence, $\rho=0.2$ (weak correlation) and 0.4 (moderate correlation) for the simulations.

Evaluation Criteria

We use the evaluation criteria used by El Emam *et al.* (2001) and Gupta (2003) to evaluate the performance of the models.

Decision Accuracy

CR models are used to make a binary re-inspection decision, i.e. re-inspect or not re-inspect. For controlling inspections, this decision would be based on whether the effectiveness of the inspection is above a specified threshold. The effectiveness threshold is set to ensure a high quality of inspection. Since actual effectiveness is not known, CR estimate are used to calculate the estimated effectiveness.

Let Q_p denote the threshold effectiveness set by the organization, then the decision can be stated in terms of the following inequality (Gupta, 2003):

$$Q_p \leq \frac{D}{\hat{N}}$$

where, D is the total number of defects found and \hat{N} is the estimated number of defects. The ratio, $\frac{D}{\hat{N}}$ is the estimated inspection effectiveness. The artefact is passed on to the next phase if this inequality is satisfied. If it is not satisfied, then the artefact should be re-inspected.

In a study (Briand *et al.*, 1998) it was found that the average effectiveness of code inspections in practice was 0.57, and the most likely value was 0.7 and hence we used these two values. The lower threshold intended to ensure “above average” defect detection effectiveness, and the higher threshold is intended to ensure “best in class” effectiveness.

Simulations

In our model it was very difficult to get the posterior estimate using the traditional integration method. Hence, we used Gibbs Sampling (George *et al.* 1992) to get the estimate for N .

Gibbs Sampling

The Gibbs sampler is a technique for generating random variables from a (marginal) distribution indirectly, without having to calculate the density (George *et al.* 1992). Most of the applications of Gibbs sampler have been in Bayesian models and it reduces the amount of work required to do complicated calculations.

Selection of Bayesian Parameters

The most important factor in a Bayesian method is the choice of the prior distribution. In our research the prior distribution used is the Dirichlet distribution ($\kappa\alpha$) described by κ and α . As stated earlier we use Gibbs sampling to obtain the estimate for N and the conditional distributions.

This distribution is described by a and b parameters of a β distribution which are determined by the prior mean $E(p)$ and the standard deviation σ_p . Since we do not have any knowledge of the prior, we try to determine these parameters from the inspection data. We define the mean as n_0/N , where n_0 is the maximum number defects detected by an inspector, and N is the total number of defects in the artefact. We allow our estimated number to deviate as much as up to $\pm 30\%$ from the actual defect population. Thus, the prior mean can now be written as,

$$E(p) = \frac{n_0}{N - Er \times N}$$

where, offset from the actual mean, $Er = 0, \pm 0.1, \pm 0.2, \pm 0.3$.

We chose values of 0.025, 0.05, 0.075, 0.1 and 0.2 for standard deviation. For each value of the standard deviation, we had seven values of $E(p)$ as given above. Now, a and b can be calculated from $E(p)$ and σ_p as follows,

$$a = E(p) \left(\frac{E(p)}{\text{var}(p)} [1 - E(p)] - 1 \right)$$

$$b = a \frac{1 - E(p)}{E(p)}$$

κ and α are derived from a and b using the following formula,

$$a = \kappa \alpha$$

$$b = \kappa (1 - \alpha)$$

Study Points

We consider the following sets of variables for our simulations: the number of difficult defects, the probability of a defect being found, number of inspectors and their defect detection capability. We performed simulations with the population size of 10, 20 and 30 difficult defects with detection probabilities of 0.1 (very difficult to detect) and 0.4 (moderately difficult). We used 2, 3 and 4 inspectors for the simulations and we also considered for the simulations the general defect detection effectiveness of the inspectors themselves (i.e., their ability to detect defects). The last variable that we used was the correlation factor.

Study points for each type of simulation were constructed by combining the values of the above mentioned variables. For each study point 1000 inspections were simulated.

VII. RESULTS

The purpose of our simulations has been to study the effect of dependence among inspectors on the decision accuracy and compare it with the previous Bayesian model (Ananda 1997, Gupta 2003). We have used realistic scenarios of small numbers of rather difficult defects, which can have serious impact on the quality of the product. We present the results in terms of decision accuracy for both the thresholds of 0.57 and 0.7 for 2, 3 and 4 inspectors and the defect size of 10, 20 and 30. The above results are also presented with respect to two degrees of difficulties, 0.1 (extremely difficult defects) and 0.4 (moderately difficult defects); and two degrees of dependence, 0.2 (weak correlation) and 0.4 (moderate correlation) as was done by Basu and Ebrahimi (1998), Basu and Ebrahimi (2001), and Basu (2003).

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.91	0.97	1.00	1.00	1.00	1.00
0.2	0.77	0.90	0.99	1.00	1.00	1.00
0.1	0.89	0.95	0.99	1.00	1.00	1.00
0.0	0.82	0.95	0.98	1.00	1.00	1.00
-0.1	0.79	0.95	0.96	1.00	1.00	1.00
-0.2	0.87	0.95	1.00	1.00	1.00	1.00
-0.3	0.82	0.88	0.97	1.00	1.00	1.00

Table 1: DA for 10 Defects of 0.1 degree of difficulty and standard deviation of 0.025 and $\rho=0.2$

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.87	1.00	1.00	1.00	1.00	1.00
0.2	0.91	0.99	1.00	1.00	1.00	1.00
0.1	0.93	0.99	1.00	1.00	1.00	1.00
0.0	0.91	0.99	1.00	1.00	1.00	1.00
-0.1	0.90	1.00	1.00	1.00	1.00	1.00
-0.2	0.91	1.00	1.00	1.00	1.00	1.00
-0.3	0.84	0.99	0.99	1.00	1.00	1.00

Table 2: DA for 20 Defects of 0.1 degree of difficulty and standard deviation of 0.025 and $\rho=0.2$

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.93	0.98	1.00	1.00	1.00	1.00
0.2	0.90	0.99	1.00	1.00	1.00	1.00
0.1	0.87	0.98	0.99	1.00	1.00	1.00
0.0	0.90	1.00	1.00	1.00	1.00	1.00
-0.1	0.92	0.98	0.99	1.00	1.00	1.00
-0.2	0.79	0.97	1.00	1.00	1.00	1.00
-0.3	0.92	0.97	0.99	1.00	1.00	1.00

Table 3: DA for 30 Defects of 0.1 degree of difficulty and standard deviation of 0.025 and $\rho=0.2$

Main Results

The main variables in our simulations were the number of inspectors and their abilities, number of defects and their degree of difficulty and the correlation factor. The detailed results are given in Appendix A. We present a summary of the results in this section.

We observed that the DA decreased as the standard deviation of the prior increased. We also noted that within a given standard deviation, the DA did not seem to change significantly with respect to the error (Er). It was also

observed that the failure rate in each simulation increased when the abilities of the inspectors decreased and also when the defects became more difficult to find. Additionally, the DA increased with an increase in the dependence among inspectors. These general trends were observed throughout the entire simulations.

1. We first considered changing the number of inspectors while keeping all the other parameters fixed. The results for 10, 20 and 30 defects and the degree of difficulty of 0.1 and standard deviation of 0.025 are shown in Tables 1, 2 and 3. The DA for both the thresholds (0.57 and 0.7) increased as the number of inspectors increased. Here we chose the inspector abilities as 0.5 for all the inspectors, which means a moderate ability. The DA of 10 defects increased from 0.82 to 1.0 for a threshold of 0.7 as the number of inspectors increased from 2 to 4. There was a similar increase in DA for a threshold of 0.57. This trend was observed for 20 and 30 defects as well (Fig. 1).
2. We obtained the second set of results by varying the ability of the inspectors. The two groups of inspectors have been chosen to illustrate the extreme variations, i.e. a team of 4 experts and a team of 4 novices. The DA for the team of experts is significantly more than the DA of a team of novices, for both the thresholds used in our simulations.
3. Next, we varied the number of defects while keeping the number and the ability of the inspectors fixed. The DA increased with increasing number of defects for a given number of inspectors. For example, the DA increases from 0.54 to 0.84 for a threshold of 0.7 for 10 defects. Similar increase in DA was observed for 0.57 thresholds as well. The same trend was observed for all the inspection teams.
4. Lastly, we varied the degree of correlation while keeping all the other parameters fixed. The DA increased with the increase in the degree of correlation. Similar trend is observed for 3 and 4 inspectors also (Fig. 2).
5. The most important result is that the DA of Dependent Bayesian CR model is remarkably higher than the Bayesian CR model (Gupta, 2003) under realistic conditions (Fig. 3).

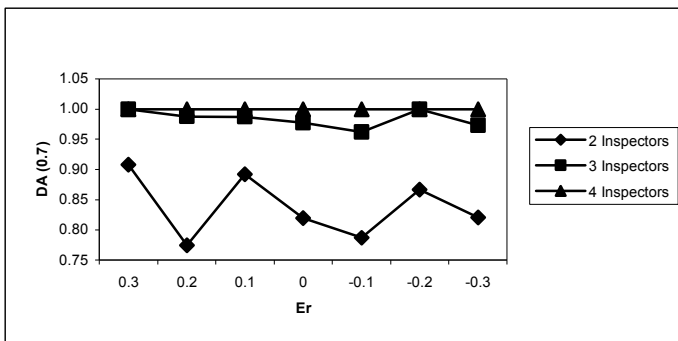


Fig. 1: DA for 10 defects of 0.1 degree of difficulty and standard deviation of 0.025 and $\rho=0.2$

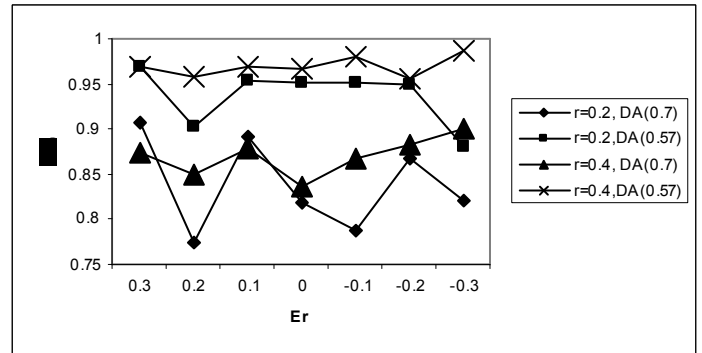


Fig. 2: Comparison for $\rho=0.2$ and 0.4, 10 defects and 0.1 degree of difficulty, 2 inspector and moderate inspector abilities (0.5) and standard deviation = 0.025.

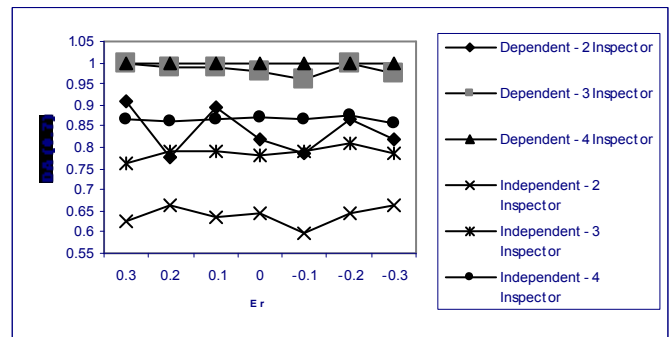


Fig. 3: Comparison of DA of 0.7 for the two Bayesian models for 10 defects and 0.1 degree of difficulty and $\rho=0.2$. Inspection team consists of moderate abilities, i.e. 0.5

We can understand all the results of the Bayesian model in terms of the inspection data. The data matrix is sparse for low number of defects and gets filled with large number of defects as well as larger number of inspectors, thereby increasing the DA. For example, there is a 25% gain in the DA as the defect population increases from 10 to 20 defects, and from 20 to 30 defects for 2 inspectors and with 0.1 degree of difficulty. Also, DA is increased as the inspector abilities increase from a team of novices to a team of experts. Basically, for novices the matrix will be sparse and their subjective estimates will be wrong in the worst-case scenario.

It is clear from our results that the DA increases with ρ , which describes the degree of dependency among the inspectors. More dependency such as $\rho = 0.4$ gives better DA than the lower value of $\rho = 0.2$. As the dependence is increased, the inspectors will find more errors and hence the data matrix gets filled up and therefore the estimate about the number of errors is better.

VIII. CONCLUSIONS

The main focus of our research was to compare the performance of a Bayesian model with dependence among inspectors under a realistic scenario and compare it with the

performance of the previous Bayesian model (Gupta 2003). We extended the work of Gupta (2003) and used the same set of parameters and variables to evaluate our model. Extensive Monte Carlo simulations were carried out for 2, 3 and 4 inspectors and defect population of 10, 20 and 30 with 0.1 and 0.4 degrees of difficulty and two degrees of dependence 0.2 and 0.4. We find that the decision accuracy improves with increase in the number of inspectors, and number of defects, when the defects become easier to find and also with an increase in the degree of dependence. We can understand the results and the behaviour of the Bayesian model in the context of the sparseness of the data matrix and the prior distribution. In the case of a sparse data matrix, the estimated population depends entirely on the prior mean and the standard deviation; hence in such a case it is very important to have an accurate value of the prior distribution. However, when the data matrix is not sparse, the estimation of the population of defects depends more on the likelihood function rather than on the prior distribution. Therefore, with a comparatively full data matrix, the likelihood function gets better defined. As the prior moves away from the correct value, decision accuracy drops significantly. Since the reliability of the decision accuracy depends on the correctness of the prior distribution, particularly in the case of a sparse data matrix, it is rather essential to have a correct prior distribution. Also, with the introduction of dependence among inspectors, the number of defects detected increases, since an error detected by an inspector is more likely to be detected by other inspectors.

Comparing the independent inspector Bayesian model with the dependent inspector Bayesian model, it is observed that under most circumstances the decision accuracy values of the dependent inspector Bayesian model are higher than those for independent inspector Bayesian model. The decision accuracy, even for 4-inspector independent inspector Bayesian model in some cases is much less than 2, 3-inspector dependent inspector Bayesian model.

REFERENCES

1. Ananda, M. M. A., "Bayesian methods for mark-resighting surveys", *Comm. in Statistics - Theory and Methods*, 26(3), 685 – 697, 1997.
2. Basu, S. and Ebrahimi, N., "Estimating the number of undetected errors: Bayesian model selection", *Proc. of the International Symposium on Software Reliability Engineering*, pp 22-31, 1998.
3. Basu S. and Ebrahimi N., "Bayesian capture–recapture methods for error detection and estimation of population size: heterogeneity and dependence", *Biometrika*, 88(1): 269–279, 2001.
4. Basu S., "Bayesian inference for the number of undetected errors", 2003, Private Communication.
5. Briand, L., Emam, K. El. and Freimut, B., "A comparison and integration of capture recapture models and the detection profile method", *Proc. of the 9th International Symposium on Software Reliability Engineering*, 32-41, 1998.
6. Briand, L., Emam, K. El. and Freimut, B., "A comprehensive evaluation of capture-recapture models for estimating software defect content", *IEEE Transactions on Software Engineering*, 26(6), 518-540, 2000.
7. Eick, S.G, Loader, C.R., Long, M.D, Votta, L.G. and Vander Weil, S., "Estimating software fault content before coding", *Proc. of the 14th International Conference on Software Engineering*, 59-65, 1992.
8. Emam, K. El. and Laitenberger, O., "Evaluating capture-recapture models with two inspectors," *IEEE Transactions on Software Engineering*, 27, Sep 2001.
9. Fagan, M.E., "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, 15(3), 182-211, 1976.
10. George E.I. and Robert, C.P., "Capture-recapture estimation via Gibbs sampling", *Biometrika*, 79, 677-683, 1992.
11. Gilb, T. and Graham, D., "Software Inspection", *Addison-Wesley Publishing Company*, 1993.
12. Gupta, V., Patnaik, A. R., Emam, K. El. And Goel, N., "System for controlling software inspection", *CCECEC 2003 – Canadian Conference on Electrical and Computer Engineering, May 2003*.
13. Gupta V., "A study of estimation techniques of software defect content", *Masters Theses, School of Computer Science, Carleton University, Ottawa*, 2003.
14. NASA-GB-A302, "Software formal inspections guidebook", *National Aeronautics and Space Administration*, Washington, DC, 1993
15. Otis, D., Burnham, K., White, G. and Anderson, D., "Statistical inference from capture data on closed animal populations", in *Wildlife Monographs*, 62, 1-135, 1978.
16. O'Neill, D., "Issues in software inspection", *Software, IEEE Volume 14, Issue 1*, Jan.-Feb., 18 – 19, 1997.
17. Petersson, H., Thelin, T., Runeson, P., Wohlin, C., "Capture–recapture in software inspections after 10 years research—theory, evaluation and application", *The Journal of Systems and Software*, 72, 249–264, 2004.
18. Seber, G.A.F., "The Estimation of animal abundance and related parameters", *Charles Griffin & Company Ltd.*, London, 2nd edition, 1982.
19. Thelin, T. and Runeson, P., "Robust estimation of fault content with capture recapture and detection profile estimators", in *Proc. of the Conference of Empirical Assessment in Software Engineering*, 1999(a).
20. Thelin, T. and Runeson, P., "Capture-Recapture estimations for perspective-based reading- a simulated experiment ", *Proc. of the International Conference on Product Focused Software Process Improvement*, 182- 200, 1999(b).
21. White, G.C., Anderson, D.R., Burnham, K.P. and Otis, D.L., "Capture-Recapture and removal methods for sampling closed population", *Technical report*, Los Alamos National Laboratory, 1982.
22. Wohlin, C., Runeson, P. and Brantestam, J., "An experimental evaluation of capture recapture in software inspections", *In Software Testing, Verification and Reliability*, 5, 213-232, 1995.