

A SYSTEM FOR CONTROLLING SOFTWARE INSPECTIONS

Vandana Gupta, Alok R. Patnaik, Nishith Goel
Cistel Technology Inc.,
200-210 Colonnade Road,
Ottawa K2E 7L5.
email: vandana@cistel.com

Khaled El Emam
National Research Council, Canada, Institute for
Information Technology
M-50, Montreal Road, Ottawa K1A 0R6.
Khaled.El-Emam@nrc-cnrc.gc.ca

Abstract

Software inspections are a powerful tool for detecting faults in software during the early phases of the life cycle. Deciding when to stop inspections is an important determinant of inspection effectiveness. Capture-recapture (CR) models can be used to estimate defect content, and hence help to make a reinspection decision. We present Monte Carlo simulations of six CR models. The objective is to find the best CR model. This builds on previous work by simulating the context of high-reliability systems. The results indicate that model MtCh, which underestimates median relative error, gives zero failures and has the best decision accuracy.

Keywords: Inspection, Capture-recapture models, Defect Content Estimation.

1. Introduction

Software inspection is a method for statically verifying documents. Michael Fagan first described software inspections in 1976 (Fagan, 1976 [1]). Identifying a defect early in the life-cycle of a software object helps to detect design problems and thereby prevents its spread to other pieces of software. Such early detection and mitigation of defects is highly desirable both from reliability and economic point of view. A typical software inspection may involve a few inspectors, between 3 to 5, sometimes even more depending on the nature of the software. The results of these inspections are critical in determining the need for re-inspection which can be evaluated in an objective manner using statistical models and thus ensuring reliability. For example, Doolan (1992)[2] reports industrial experience indicating a 30 times return on investment for every hour devoted to inspection of software requirement specifications. Russel (1991)[3] reports a similar return of 33 hours of maintenance saved for every hour of inspection invested.

This benefit depends primarily on the quality of the inspection. For increasing the effectiveness of inspections, contemporary research has focused on improved reading techniques and on re-inspections (El Emam and Laitenberger 2001 [4]). Reinspections can be considered a part of the general problem of when to stop inspections. The decision of re-inspection further depends on a good estimation of the number of defects in the software artefact and the abilities of the individual inspectors. The statistical models to describe these results are derived from wildlife research to determine animal population. One of the aims here is to arrive at a suitable model that best matches the software inspection process.

In the next section we give a brief overview of the capture-recapture (CR) methods, followed by the simulation. The results are presented in section 4 and implications are discussed in section 5.

2. Overview of Capture-Recapture Models

In wildlife research CR models are used to estimate the animal population. During CR studies animals are captured, marked, and released on several trapping occasions. When a marked animal is captured on a subsequent trapping occasion it is said to be re-captured. CR models use these data to estimate the population size. This principle can be used in software inspections (Eick et al. 1992 [5]). A defect discovered by one inspector and rediscovered by another is said to be re-captured. The method uses the overlap between the sets of faults found by the different reviewers to estimate the fault content. The capture-recapture based estimation of population size begins with sampling of the population. The results of sampling are used as parameters in an estimator function, which gives the size of the population, if certain conditions are fulfilled. The following are the most commonly used CR models in software engineering:

- Model M0: All different defects have the same detection probability, and all inspectors have the same detection capability.
- Model Mh: Different defects can vary in their detection probability, but all inspectors have the same detection capability.

- Model Mt: All different defects have the same detection probability, but the inspectors have different detection capabilities.
- Model Mth: This allows for different detection probabilities for the different defects and inspectors.

There are two estimators for model Mt (Maximum Likelihood Estimator (MtMLE), Chao's Estimator (MtCh)[6]). Similarly, for model Mh there are two estimators (Jackknife Estimator (MhJE), Chao's Estimator (MhCh)[7, 8]). In our study we considered all of the above six models.

3. Research Method

In this section we specify the study points for our simulation, and describe how the different models were evaluated.

In a given piece of software there are defects with varying degree of difficulty. While it is relatively simpler to find and correct minor defects, the major defects, though their number is small, remain largely undetected and cause potentially the most serious damage. Here we are more concerned with the smaller number of major defects since the population of these defects is more important to estimate.

3.1 Study Points

As mentioned earlier, we performed simulations with the population size of 10, 20 and 30 hard 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.

Inspectors	Ability to detect defects
2	(0.1, 0.9), (0.25, 0.75), (0.4,0.6), (0.3, 0.3), (0.8, 0.8), (0.5, 0.5)
3	(0.1, 0.5, 0.9), (0.25, 0.5, 0.75), (0.4, 0.5, 0.6), (0.3, 0.5, 0.3), (0.8,0.5,0.8), (0.5,0.5,0.5)
4	(0.1,0.4,0.6,0.9), (0.1,0.1,0.9,0.9), (0.5,0.5,0.9,0.9), (0.9,0.9,0.9,0.9), (0.1,0.1,0.1,0.1), (0.5,0.5,0.5,0.5), (0.5,0.5,0.5,0.9), (0.1,0.5,0.5,0.5), (0.1,0.1,0.1,0.9), (0.1,0.9,0.9,0.9)

Table 1 Numbers within the brackets indicate the ability of an inspector to detect a defect for each inspection. For example, (0.9,0.9,0.9,0.9) is a team of four experts, while (0.1,0.1,0.1,0.1) is a team of four novices.

For each study point, defined as a combination of inspector ability, defect population size and defect difficulty, 1000 inspections were simulated for each of the six models.

3.2 Evaluation Criterion

We computed median relative error (med(RE)) which is the median of the 1000 simulations, of difference between estimated and actual population as a fraction of actual population. Thus med(RE) gives an indication of a model's bias (i.e. this would allow us to understand the behaviour of the CR models and help interpret the results of the decision accuracy evaluations). Decision accuracy is a measure of how accurately the model can predict the correct decision, to re-inspect or not to. Basically, the CR models are used to make a binary re-inspection decision. For controlling inspections, this decision would be based on whether the effectiveness of the inspection is above a specified threshold. We are considering two values of thresholds for decision accuracy, 0.57 and 0.7. Furthermore, we compute the number of times a model fails to provide an estimate.

4. Results

In these simulations our goal has been to find the best model that helps us accurately predict the decision on re-inspection. We have used realistic scenarios of small number of rather difficult defects which can have serious impact.

- Most models do not perform well in these conditions i.e. models tend to fail to estimate the population, often 90% of the simulations. The only model which estimated every time is MtCh. Increasing the number of inspectors from 2 to 4 does not improve the performance of the other models. We therefore consider MtCh to be the best of the models studied.
- The model MtCh tends to underestimate the relative median error in the sense that it has a large negative bias. It tends to give a decision accuracy better than the other models. This is consistent with the results of previous studies [4].
- The most important factor affecting the failure of the models is the overlap. Overlap is the identification of identical defects found by multiple inspectors. Experts in the team help to increase this factor.
- We find from our simulations that as we increase the number of inspectors from 2 to 4 the number

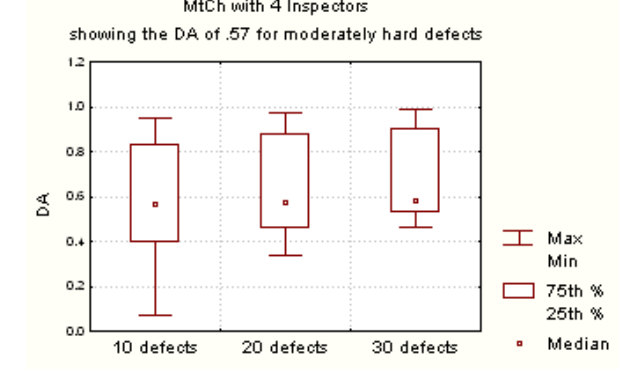
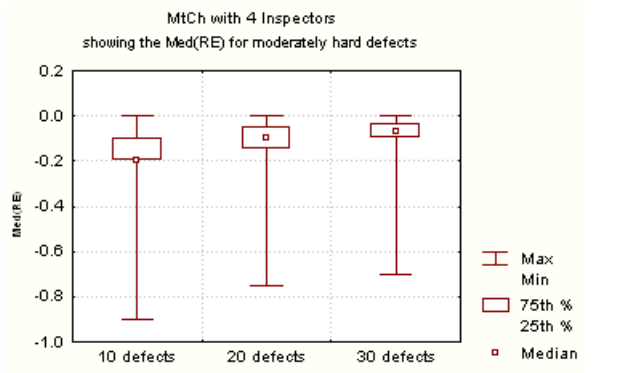
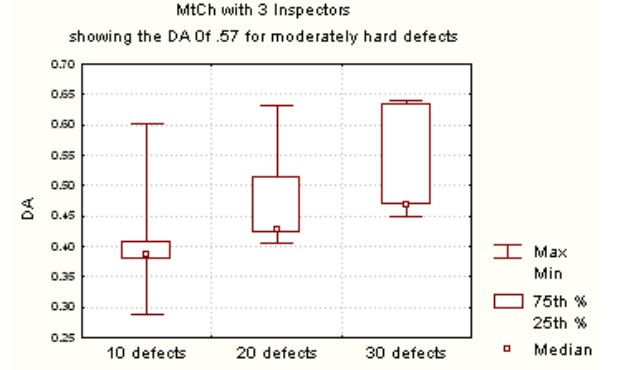
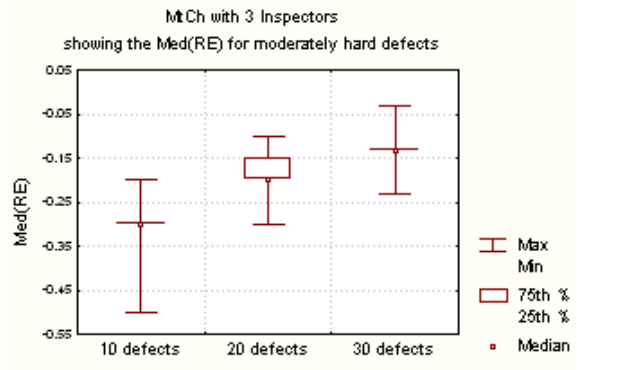
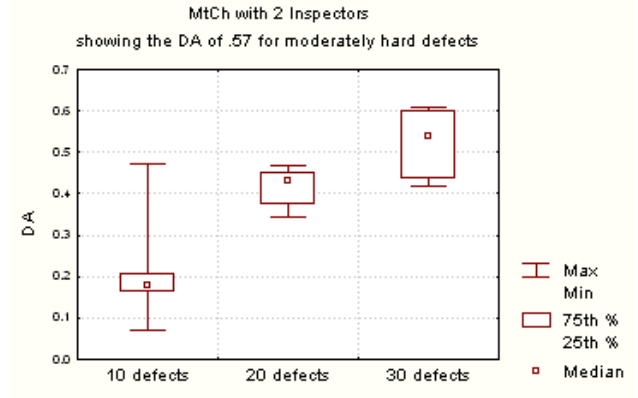
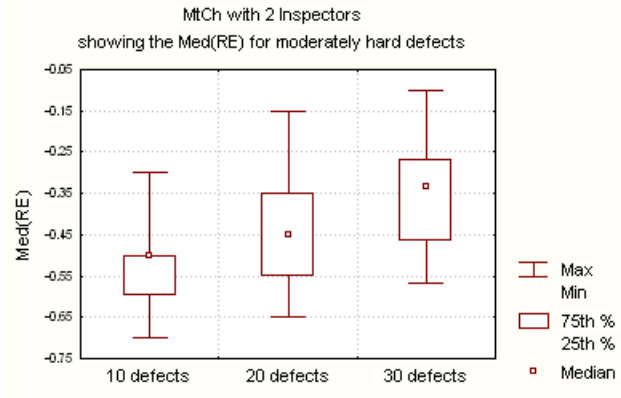


Fig 1: Box plots showing Med(RE) for 2, 3 and 4 inspectors for defect population of 10, 20 and 30 moderately hard defects (probability of detection of such a defect of 0.4).

Fig 2: Box plots showing Decision accuracy (DA) for 2,3 and 4 inspectors for a threshold of 0.57 for moderately hard defects.

of failures decrease for any given model, and the median relative error moves closer to 0.0. Also, when the number of defects increases from 10 to 30, we notice improvements in the above mentioned parameters.

We present the median relative error and decision accuracy for the model MtCh for 2, 3 and 4 inspectors in Fig 1 and 2 respectively. As noted earlier both decision accuracy and med(RE) improve as the number of inspectors increase, and also when the defect population increases. As expected, the variation in the results increases with the decrease in defect population.

5. Discussion

Given the small number of hard defects, the CR models fail to estimate the population of such defects. We believe that it is mainly due to the effects of overlap of defects being detected by various inspectors. However, we notice less failures when the number of inspector is increased, as one may expect. In the case of mixed type of defects, the failure rate is small since even though the hard defects remain undetected, it is the number of easy defects that allow the models to estimate the population. As mentioned earlier, we are more interested in the estimation of population of hard defects and hence the decision to re-inspect the software. Among the models tried, model MtCh consistently gives better results than the other models, confirming the results found earlier by El Emam and Laitenberger (2001)[4].

So far, model MtCh has been observed to estimate better than the rest of the models. This is supported by the figures showing the median relative error of model with 4 inspectors and 10 and 30 hard defects. We have two different types of defects, hard and moderately hard. The hard defects tend to have larger bias (farther away from zero) than moderately hard defects. This can be understood from the fact that the overlap parameters for the most difficult defects is low and hence leading to larger bias.

As we pointed out, the most important factor in our study has been the overlap parameter. The larger the overlap, ie chance of the defect to be detected by more than one inspector is more, the better is the estimate of defect population. Therefore, with a smaller number of inspectors and considering only the hard defects, a number of models fail to estimate the defect population. With larger number of failures the other parameters of the simulations such as median relative error, inter-quartile range, decision accuracy are rather poorly determined and hence unreliable. In order to improve these parameters, obviously we have to increase the overlap of defects found by the inspectors. This can be done in two ways: by increasing the number of inspectors along with the increase in their defect detection ability and by increasing the number of defects. For high reliability system, the number of hard defects will be minimal.

While considering the overlap of defects, it is necessary to consider the abilities of the inspectors. As one expects, having experts in the team helps in estimating the number of defects more accurately and hence decide the need for re-inspection of the software.

6. Conclusion

Capture-recapture models have been proposed as a means for controlling the effectiveness of software inspections. In this paper we showed the effect of CR models on a less

number of difficult defects. The model that we found most effectively detecting the faults is MtCh. Compared to other models, this one did not fail to provide an estimate under any of the conditions we studied, and therefore is a reasonable choice. As the decision accuracy for MtCh is better than the rest of the models, but is not very satisfactory. Previous results indicate that combining the estimates with the other variables can improve decision accuracy [9].

In this sense we are encouraged to look into capture-recapture models dealing with endangered species in wildlife where the models must deal with a low number of observations.

References

- [1] M.E Fagan, "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems Journal*, 15(3): 182-211,1976.
- [2] E.P Doolan, "Experience with Fagan's Inspection Method ," *Software-Practice and Experience*, vol. 22(2), pp. 173-182. , Feb 1992.
- [3] G.W Russell, "Experience with Inspections in Ultralarge-Scale Developments," *IEEE Software*, vol 8, pp. 25-31, 1991.
- [4] K.El Emam and Oliver Laitenberger, "Evaluating Capture-Recapture Models with Two Inspectors," *IEEE Transactions on Software Engineering*, vol 27, Sep 2001.
- [5] Eick, S.G, Loader, C.R, Long, M.D, Votta, L.G and Vander Weil, S.A, "Estimating Software Fault Content Before Coding", *Proc. of the 14th International Conference on Software Engineering*, pp. 59-65, 1992.
- [6] A Chao, "Estimating the Population Size for Capture-Recapture Data with Unequal Catchability," *Biometrics*, 43, pp783-791 (1987).
- [7] A Chao, "Estimating Animal Abundance with Capture Frequency Data," in *Journal of Wildlife Management*, 52, pp 295-300, 1988.
- [8] A Chao, "Estimating Population Size for Sparse Data in Capture-Recapture Experiments," *Biometrics*, 45, pp427-438, 1989.
- [9] J Barnard, K. El Emam and D Zubrow, "Using Capture-Recapture Models for the Reinspection Decision," *Software Quality Professional*, vol 5, March 2003.