The University of Osaka
Institutional Knowledge Archive

Title	Extended Metrics to Evaluate Cost Effectiveness of Software Inspections
Author(s)	Sabaliauskaite, Giedre; Kusumoto, Shinji; Inoue, Katsuro
Citation	Transactions on Information and Systems. 2004, E87-D(2), p. 475-480
Version Type	VoR
URL	https://hdl.handle.net/11094/50957
rights	copyright©2004 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

LETTER

Extended Metrics to Evaluate Cost Effectiveness of Software Inspections

Giedre SABALIAUSKAITE[†], *Student Member*, Shinji KUSUMOTO[††], *and* Katsuro INOUE[††], *Members*

SUMMARY Software inspection is one of the most effective methods to detect defects. However, inspections are not always worthwhile. This letter proposes an inspection cost model to describe inspections-related costs and extended metrics to evaluate the cost effectiveness of software inspections.

key words: *software metrics, cost effectiveness, software inspection*

1. Introduction

For more than twenty-five years software inspections have been considered an effective and efficient method for software quality improvement [3], [4], [9]. The goal of inspections is to detect defect before the testing stage begins. According to the data published in the literature, software design inspections save on average 44% of the defect detection costs, and code inspections save on average 39% of the defect detection costs [1]. However, are the software inspections always worthwhile?

In order to evaluate the effectiveness of the inspections with respect to software development cost, several metrics have been previously proposed [2], [3], [6]. Collofello et al. have taken into account of the costs consumed and saved by inspections and proposed a metric, called Cost Effectiveness [2]. Kusumoto et al. proposed a metric M_k for evaluating cost effectiveness of inspections, which is based on the degree to which costs to detect all faults from the software in a project are reduced by inspections [6]. Yet, none of those metrics includes the information about false positives (the issues that require no repair) introduced during inspection, although the rework of false positives is costly and can introduce new defects [8], [12].

This letter presents a) an inspection cost model that describes all costs related to inspections, b) extension of the metric M_k to evaluate the cost effectiveness as well as the losses of inspections. Proposed cost model and metrics can be useful for project managers to decide whether inspections are worthwhile.

The structure of the letter is as follows. Section 2 describes the inspection process. Section 3 describes the inspection cost model. In Sect. 4, we review the existing metrics to evaluate software inspections. Section 5 presents the extended metrics for software inspection evaluation. In Sect. 6, we compare extended metrics to Collofello's metric and M_k. Section 7 concludes the letter with an overall summary and directions for future research.

Manuscript received June 19, 2003.
Manuscript revised August 28, 2003.
[†]The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan; and Information Systems Department, Kaunas University of Technology, Kaunas 3026, Lithuania.
[††]The authors are with the Graduate School of Information Science and Technology, Osaka University, Toyonaka-shi, 560–8531 Japan.

2. Inspection Process

Software inspection as structured process was first described by Fagan [3]. It consists of the following stages: *Planning* (moderator ensures that the entry criteria are met); *Overview* (author of the product presents the review team an overview of the product); *Preparation* (individual reviewers analyze the work product with the goal of understanding it thoroughly); *Inspection meeting* (reviewers meet face-to-face in an inspection team to scrutinize the work product for defects); *Rework* (author performs rework to correct the identified defects); *Follow-up* (moderator verifies that all the defect have been corrected). Recently, the goals of preparation and inspection meeting stages have been modified. The common modification is to include defect detection as an explicit goal of preparation [7], [10], [11]. Furthermore, several researchers [5], [10], [13] discuss if the inspection meetings are really necessary, because they increase the cost of software inspection and usually do not detect more defects than it was already detected during preparation stage.

We will introduce two diagrams (Fig. 1 and Fig. 2) to explain the life-cycles of defects and false positives respectively. These diagrams show in what stages defects and false positives are being introduced, detected and removed.

The life-cycle of a defect is depicted in Fig. 1. It shows four cases of defect's life-cycle: d1, d2, d3 and d4. In all the cases, defects are introduced before inspection process begins, for instance during designing or coding. According to Fagan's model [3], defects are detected and confirmed during inspection meeting stage, and removed by author during rework (case d1). However, defects are usually detected by individual reviewers during preparation, confirmed by inspection team during inspection meeting, and removed by author during rework (case d2). In some cases, defects are detected during preparation, however not confirmed as defects during inspection meeting (case d3). Some defects are not detected during inspection at all (case d4). In cases d3 and d4, defects are detected and removed only during testing.

The life-cycle of a false positive is shown in Fig. 2. It

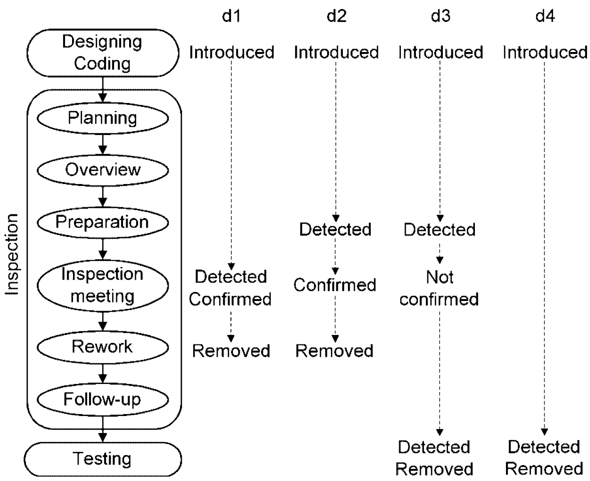

Fig. 1 Life-cycle of a defect.

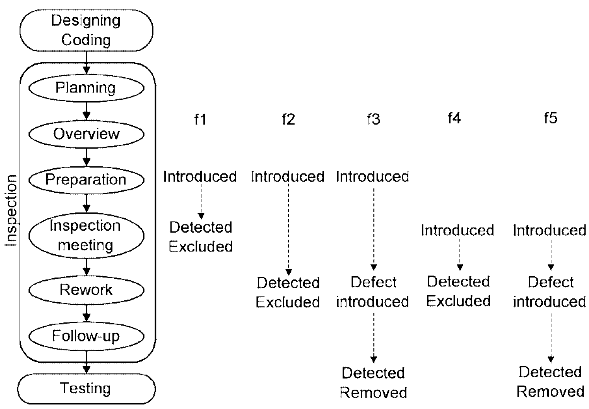

Fig. 2 Life-cycle of a false positive.

identifies five cases of false positive's life-cycle: f1, f2, f3, f4 and f5. False positives can be introduced during preparation or inspection meeting stages. In cases when false positives are introduced during preparation stage, they can be detected and excluded from defect list by inspection team during inspection meeting (case f1) or by author during rework (case f2). False positives introduced during inspection meeting can be detected and excluded from defect list by author during rework (case f4). However, if false positive is not excluded from the defect list, the rework will be done and consequently a defect may be introduced, which will be detected and removed only during testing (cases f3 and f5).

The goal of the inspection is to ensure that the minimum number of defects and false positives reaches testing.

3. Inspection Cost Model

3.1 Traditional Cost Model

The traditional inspection cost model consists of the following components [6] (Fig. 3): C_r – cost spent for inspection; C_t – cost needed for testing; ΔC_t – testing cost saved by inspection; *Virtual testing cost* – testing cost if no inspections

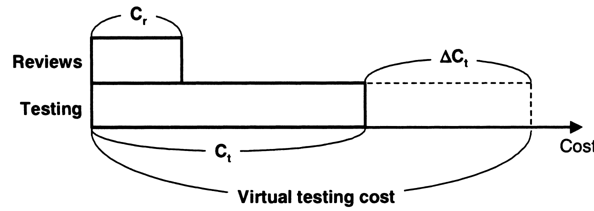

Fig. 3 Traditional cost model.

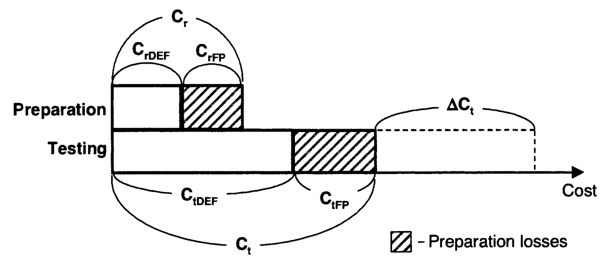

Fig. 4 Extended cost model for preparation stage.

are executed. By spending cost C_r during inspection, the cost ΔC_t is being saved during testing.

3.2 Extended Cost Model for Preparation Stage of Inspection

In order to evaluate the influence of the false positives introduced during preparation stage of inspection over the testing cost, we extend the traditional cost model. The following additional costs are defined (Fig. 4): C_{rDEF} – cost spent to detect actual defects during preparation; C_{rFP} – cost spent to detect false positives during preparation; C_{tDEF} – cost needed for testing to detect remaining defects; C_{tFP} – cost needed for testing to detect defects introduced by false positives during preparation.

In this model, by spending cost C_{rDEF} during inspection, the testing cost ΔC_t is being saved. However, by spending the cost C_{rFP} during inspection, the cost C_{tFP} is being added to the testing cost. Therefore, the costs C_{rFP} and C_{tFP} are the preparation stage losses (Fig. 4).

3.3 Extended Cost Model for Preparation and Inspection Meeting Stages of Inspection

Inspection meetings are usually carried out after preparation stage is completed. However, several authors question the usefulness of the meetings [5], [10], [13]. To enable the evaluation of the benefits and the losses of inspection meetings, we propose an extended cost model for preparation and inspection meeting stages (Fig. 5).

The following are the costs (Fig. 5): C_{mDEF} – cost spent to confirm actual defects detected during preparation stage; C_{mFP} – cost spent to confirm false positives detected during preparation stage; C_{mADD_DEF} – cost spent to detect additional defects during inspection meeting stage; C_{mADD_FP} – cost spent to detect additional false positives during inspection meeting stage; C_{mLOST_DEF} – cost spent to eliminate

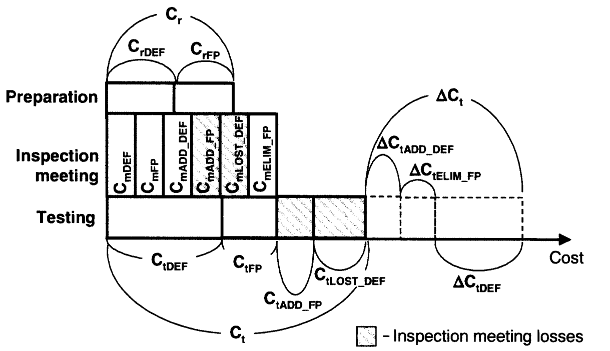

Fig. 5 Extended cost model for preparation and inspection meeting stages.

actual defects detected during preparation stage; C_{mELIM_FP} – cost spent to eliminate false positives detected during preparation stage; C_{tADD_FP} – cost needed for testing to detect defects introduced by additional false positives detected during inspection meeting stage; C_{tLOST_DEF} – testing cost needed to detect defects lost during inspection meeting stage; ΔC_{tADD_DEF} – testing cost saved by additional defects detected during inspection meeting stage; ΔC_{tELIM_FP} – testing cost saved by false positives eliminated during inspection meeting stage; ΔC_{tDEF} – testing cost saved by defects detected during preparation and confirmed during inspection meeting stages.

In this model, cost C_{mELIM_FP} spent to eliminate false positives, reduces the testing cost by ΔC_{tELIM_FP}; and cost C_{mLOST_DEF}, spent to eliminate actual defects, increases testing cost by C_{tLOST_DEF}. Inspection meeting cost spent to detect additional defects, C_{mADD_DEF}, reduces the testing cost by ΔC_{tADD_DEF}; and inspection meeting cost spent to detect additional false positives, C_{mADD_FP}, increases the testing cost by C_{tADD_FP}. In other words, the costs C_{mLOST_DEF}, C_{mADD_FP}, C_{tLOST_DEF} and C_{tADD_FP} are the costs lost by inspection meeting (Fig. 5).

Inspection meeting costs C_{mDEF} and C_{mFP}, spent to confirm defects and false positives detected during preparation stage, do not increase or reduce testing costs; however, they increase the overall inspection cost.

4. Metrics to Evaluate Software Inspections

4.1 Fagan's Metric

Fagan [3] introduced the *Error Detection Efficiency* metric M_f for measuring inspection efficiency. M_f is defined as the number of defects found during inspection over the total number of defects in the product existing before inspection. We define the total number of defects in the product existing before inspection as DEF_{total}, and the number of defects found during inspection as DEF_r. Then we get the following equation:

$$M_f = \frac{DEF_r}{DEF_{total}}. \tag{1}$$

As we can see from Eq. (1), metric M_f does not account for the cost expended for inspection.

4.2 Collofello's Metric

Collofello and Woodfield [2] proposed *Cost Effectiveness* metric M_c, which is defined as a ratio of the "cost saved by the process" to the "cost consumed by the process". Using the notation described in Sect. 3.1, we get the following equation:

$$M_c = \frac{\Delta C_t}{C_r}. \tag{2}$$

Although metric M_c takes into account the costs consumed and saved by inspections, it does not take into account the total cost to detect all defects in the software product by inspection and testing.

4.3 Kusumoto's Metric

Kusumoto et al. [6] proposed a metric M_k to evaluate the cost effectiveness of software inspections in terms of reduction of cost to detect and remove all defects from software. Using the notation described in Sect. 3.1, we get the following equation:

$$M_k = \frac{\Delta C_t - C_r}{C_t + \Delta C_t} \tag{3}$$

M_k is a ratio of the reduction of the total costs to detect and remove all defects from documents using inspections in a project to the virtual testing cost. The testing cost is reduced by ΔC_t compared to the virtual testing cost (C_t + ΔC_t) if no inspection is executed (see Fig. 3).

5. Extended Metrics

5.1 Need for New Metrics

Among metrics M_f, M_c and M_k, metric M_k is the most practical one to evaluate the cost effectiveness of inspections. However, it includes only the total cost spent on inspection, not taking into consideration the composition of the inspection cost, which was described in Sect. 3.

We decided to extend metric M_k to conform to the extended cost model. Section 5.2 presents metrics for the extended cost model for preparation stage, and Sect. 5.3 presents metrics for the extended cost model of preparation and inspection meeting stages.

5.2 Extension of Metric M_k for Preparation Stage

In accordance with extended cost model for preparation stage (see Sect. 3.2), cost ΔC_t is a testing cost saved by preparation, and costs C_{rFP} and C_{tFP} are the costs lost by inspections, since additional effort is being spent during inspection for the detection of false positives, and those false positives cause additional cost during testing (see Fig. 4). We want to introduce a new metric M_{I_IDV} to evaluate the

Preparation Losses, which is the ratio of "cost lost by inspections" by "cost saved by inspections":

$$M_{l\_IDV} = \frac{C_{rFP} + C_{tFP}}{\Delta C_t} \tag{4}$$

Not only the costs C_{rFP} and C_{rDEF}, but also C_{tFP} is the cost caused by inspection, since additional effort is being spent during testing to remove the defects introduced by false positives during preparation stage. Therefore, we want to introduce a new metric M_{g_IDV}, which is an extension of metric M_k (see Eq. (3)), to evaluate the *Extended Cost Effectiveness of Preparation Stage of Inspection*. It can be expressed using the following formula:

$$\begin{aligned} M_{g\_IDV} &= \frac{\Delta C_t - C_r - C_{tFP}}{C_{tDEF} + \Delta C_t} \\ &= \frac{\Delta C_t - C_{rDEF} - C_{rFP} - C_{tFP}}{C_{tDEF} + \Delta C_t} \end{aligned} \tag{5}$$

In metric M_k (see Sect. 4.3), the virtual testing cost (the testing cost if no inspection is executed) is defined as $(C_t + \Delta C_t)$. However, inspection increases the testing cost by C_{tFP} if false positives have been detected (see Fig. 4). Therefore, if no inspection is executed, the cost C_{tFP} should not be included into the virtual testing cost. In Eq. (5), we exclude this cost from the virtual testing cost, and define virtual testing cost as $(C_{tDEF} + \Delta C_t)$. In addition, since the testing cost C_{tFP} is the cost caused by inspection as well, we add it to inspection costs in Eq. (5).

In case if no false positives have been introduced during preparation stage $M_{g\_IDV} = M_k$, otherwise $M_{g\_IDV} < M_k$.

5.3 Extension of Metric M_k for Preparation and Inspection Meeting Stages

According to the extended cost model for preparation and inspection meeting stages (see Sect. 3.3, Fig. 5), two additional testing costs are introduced by inspection meeting: C_{tLOST_DEF} and C_{tADD_FP}. C_{tLOST_DEF} is a testing cost spent to detect defects lost during inspection meeting, and C_{tADD_FP} is a cost spent to eliminate additional false positives, introduced during inspection meeting. Inspection meeting may save testing cost by finding additional defects detected during inspection meeting stage ΔC_{tADD_DEF}, and by eliminating false positives detected during preparation stage ΔC_{tELIM_FP}.

Inspection meeting costs C_{mDEF} and C_{mFP}, spent to confirm actual defects and false positive defects detected during preparation stage, do not have influence over testing cost, however they increase the overall inspection cost.

Costs C_{mADD_DEF} and C_{mELIM_FP}, spent to detect additional defects and to eliminate false positives, reduce testing costs by ΔC_{tADD_DEF} and ΔC_{tELIM_FP} respectively. Costs C_{mADD_FP} and C_{mLOST_DEF} spent to detect additional false positives and to eliminate actual defects, increase testing costs by C_{tLOST_DEF} and C_{tADD_FP} respectively. The purpose of the inspection meeting should be to minimise the costs C_{mADD_FP} and C_{mLOST_DEF}, and to maximise the costs C_{mADD_DEF} and C_{mELIM_FP}.

Similarly as the metric M_{l_IDV} (Eq. (4)) to evaluate the losses of preparation stage, we introduce a new metric M_{l_MEET} to evaluate the *Inspection Meeting Losses*, which is the ratio of "cost lost by inspection meeting" by "cost saved by inspection meeting" (see Fig. 5):

$$\begin{aligned} M_{l\_MEET} = &\frac{C_{mADD\_FP} + C_{mLOST\_DEF}}{\Delta C_{tADD\_DEF} + \Delta C_{tELIM\_FP}} \\ &+ \frac{C_{tADD\_FP} + C_{tLOST\_DEF}}{\Delta C_{tADD\_DEF} + \Delta C_{tELIM\_FP}} \end{aligned} \tag{6}$$

Although the costs C_{mFP} and C_{tFP} are the additional costs caused by inspection, we do not include them into the metric M_{l_MEET} (Eq. (6)), because they depend on both preparation and inspection meeting stages.

Similarly as the metric M_{g_IDV} (Eq. (5)), we want to propose a metric *Extended Cost Effectiveness of Preparation and Inspection Meeting Stages*, M_{g_MEET}, to evaluate cost effectiveness of software inspections, when both preparation and inspection meeting are performed:

$$\begin{aligned} &M_{g\_MEET} \\ &= \frac{\Delta C_t - C_r - C_m - C_{tFP} - C_{tADD\_FP} - C_{tLOST\_DEF}}{C_{tDEF} + \Delta C_t} \\ &= \frac{\Delta C_{tADD\_DEF} + \Delta C_{tELIM\_FP} + \Delta C_{tDEF}}{C_{tDEF} + C_{tADD\_DEF} + \Delta C_{tELIM\_FP} + \Delta C_{tDEF}} \\ &\quad - \frac{C_{rDEF} + C_{rFP} + C_{mDEF} + C_{mFP} + C_{mADD\_DEF}}{C_{tDEF} + C_{tADD\_DEF} + \Delta C_{tELIM\_FP} + \Delta C_{tDEF}} \\ &\quad - \frac{C_{mADD\_FP} + C_{mLOST\_DEF} + C_{mELIM\_FP}}{C_{tDEF} + C_{tADD\_DEF} + \Delta C_{tELIM\_FP} + \Delta C_{tDEF}} \\ &\quad - \frac{C_{tFP} + C_{tADD\_FP} + C_{tLOST\_DEF}}{C_{tDEF} + C_{tADD\_DEF} + \Delta C_{tELIM\_FP} + \Delta C_{tDEF}} \end{aligned} \tag{7}$$

Metric M_{g_MEET} (Eq. (7)) is a modification of metric M_k (Eq. (3)). In Eq. (7), the virtual testing cost is defined as $(C_{tDEF}+\Delta C_{tADD\_DEF}+\Delta C_{tELIM\_FP}+\Delta C_{tDEF})$, and the testing costs C_{tFP}, C_{tADD_FP} and C_{tLOST_DEF}, caused by inspection, are added to the inspection cost.

6. Comparison of Proposed Metrics to M_c and M_k Metrics

The differences among M_c, M_k and proposed metrics can be demonstrated with reference to the five imaginary projects (see Fig. 6). In all projects, if no inspections had been executed, the testing cost would be 1000 units.

In cases Case I – Case III of Fig. 6, the preparation stage of inspection has been performed. In cases Case IV and Case V of Fig. 6, preparation and inspection meeting stages have been performed. The notation of Fig. 6 is taken from the extended cost models (Fig. 4 and Fig. 5).

In Case I of Fig. 6 (a), inspection consumes 10 units of cost (6 units to detect actual defects, and 4 units to detect false positives), saves 100 units of testing cost, and the testing cost is 900 (700 to detect remaining defects in software

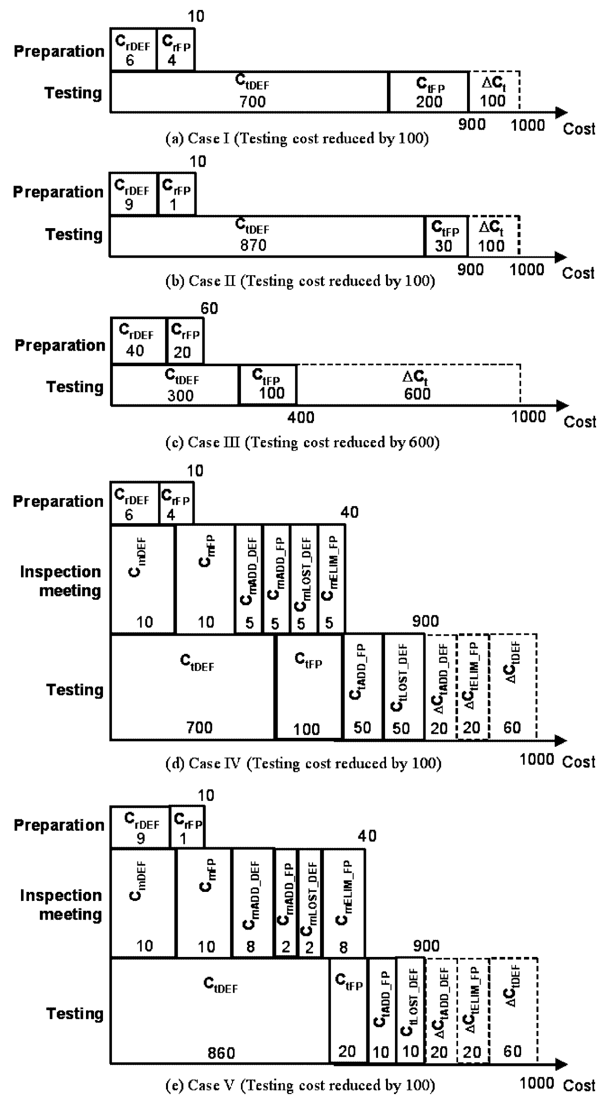

Fig. 6 Comparing five different cases of inspection.

product, and 200 to detect defects introduced by false positives). Therefore, the total cost is 910. Case II is similar to Case I, because it consumes the same costs on inspection and testing as in Case I, however the distribution of inspection and testing costs is different in Case II (during inspection, 9 units of cost are spent to detect actual defects and 1 unit to detect false positives; during testing, 870 units of cost are spent to detect remaining defects in software product and 30 to detect defects introduced by false positives). In Case III of Fig. 6 (c), inspection costs 60 units (40 units to detect actual defects, and 20 units to detect false positives), saves 600, testing cost is 400 (300 to detect remaining defects in software product, and 100 to detect defects introduced by false positives), and the total cost is 460.

If we apply Collofello's metric M_c to Cases I, II and III, the value of M_c is 10 in all cases. However, in the Case III, inspection saved much more of the total defect detection cost than in the Cases I and II, therefore Case III would be expected to be more cost effective.

The value of metric M_k for Case I and II is 0.09, and for Case III is 0.54. Thus, M_k identifies that the inspection in Case III is more effective than in Cases I and II. However, it does not show the difference between Cases I and II, although the inspection losses due to the false positives are greater in Case I. If we apply metric M_{l_IDV} to evaluate inspection losses, the value of M_{l_IDV} is 2.04 in Case I, 0.31 in Case II, and 0.2 in Case III. Thus, M_{l_IDV} identifies that inspection losses are the greatest in Case I.

The values of the extended cost effectiveness metric M_{g_IDV}, which takes into consideration inspection losses, are −0.14 in Case I, 0.06 in Case II, and 0.49 in Case III. As we can see from those results, metric M_{g_IDV} is more precise than M_k, because it shows that inspection in Case II is more effective than in Case I.

Case IV and Case V of Fig. 6 demonstrate the projects in which preparation and inspection meeting stages are performed. In both cases, preparation consumes 10 units of cost, inspection meeting consumes 40 units of cost, the testing cost is 900, and 100 units of cost are saved by inspection. However, these cases differ in the distribution of costs. During inspection meeting, in Case IV, 5 units of cost are spent to detect additional defects (Case V: 8 units), 5 units to detect additional false positives (Case V: 2 units), 5 units to eliminate actual defects (Case V: 2 units), and 5 units to eliminate false positives (Case V: 8 units). During testing, in Case IV, 700 units of cost are spent to detect defects (Case V: 860), 100 units to detect defects introduced by false positives (Case V: 20), 50 units to detect defects introduced by additional false positives detected during inspection meeting (Case V: 10), and 50 units to detect defects lost during inspection meeting (Case V: 10). The total cost in both cases is 950.

The value of metric M_c in both cases Case IV and Case V is 2, and the value of metric M_k in both cases is 0.05. The value of metric M_{l_MEET} in Case IV is 2.75, and in Case V is 0.6. Therefore, M_{l_MEET} identifies that inspection losses are greater in Case IV as compared to Case V. The value of metric M_{g_MEET} is −0.188 in Case IV, and 0.0104 in Case V. Consequently, metric M_{g_MEET} is more precise than M_c and M_k since it shows that inspection is more effective in Case V as compared to Case IV.

7. Summary and Conclusions

In this letter, we have proposed two extended cost models: a model to describe the costs spent during preparation stage, and a model to describe the costs spent during preparation and inspection meeting stages. Those models can be useful for other researchers to have a greater understanding of all the costs related to inspections.

In addition, we have proposed two new metrics M_{l_IDV} (Eq. (4)) and M_{l_MEET} (Eq. (6)) to evaluate the *Preparation Losses* and the *Inspection Meeting Losses* respectively. Also, we have proposed two metrics M_{g_IDV} (Eq. (5)) and M_{g_MEET} (Eq. (7)), which are the modifications of Kusumoto's metric (Eq. (3)), to evaluate the *Extended Cost*

Effectiveness of Preparation, and the *Extended Cost Effectiveness of Preparation and Inspection Meeting* respectively. All those metrics enable more precise evaluation of software inspections as compared to the conventional metrics.

Future research will be directed to the experimental evaluation of proposed metrics.

References

[1] L. Briand, K. El Emam, O. Laitenberger, and T. Fussbroich, "Using simulation to build inspection efficiency benchmarks for development projects," Proc. 20th International Conference on Software Engineering, pp.340–349, Kyoto, Japan, 1998.
[2] J.S. Collofello and S.N. Woodfield, "Evaluating the effectiveness of reliability-assurance techniques," J. Syst. Softw., vol.9, no.3, pp.191–195, 1989.
[3] M. Fagan, "Design and code inspections to reduce errors in program development," IBM Syst. J., vol.15, no.3, pp.182–211, 1976.
[4] T. Gilb and D. Graham, Software inspection, Addison-Wesley, 1993.
[5] P.M. Johnson and D. Tjahjono, "Does every inspection really need a meeting?," Empir. Softw. Eng., vol.3, no.1, pp.9–35, 1998.
[6] S. Kusumoto, K. Matsumoto, T. Kikuno, and K. Torii, "A new metrics for cost effectiveness of software reviews," IEICE Trans. Inf. & Syst., vol.E75-D, no.5, pp.674–680, May 1992.
[7] O. Laitenberger and J.M. DeBaud, "An encompassing life cycle centric survey of software inspection," J. Syst. Softw., vol.50, no.1, pp.5–31, 2000.
[8] L.P.W. Land, R. Jeffery, and C. Sauer, "Validating the defect detection performance advantage of group designs for software reviews: Report of replicated experiment," Proc. Australian Software Engineering Conference, Sydney, Australia, pp.17–26, 1997.
[9] O. Mizuno and T. Kikuno, "Empirical evaluation of review process improvement activities with respect to post-release failure," Proc. Empirical Studies on Software Development Engineering Workshop, Los Angeles, CA, USA, pp.50–53, May 1999.
[10] A.A. Porter and P.M. Johnson, "Assessing software review meetings: Results of a comparative analysis of two experimental studies," IEEE Trans. Softw. Eng., vol.23, no.3, pp.129–145, March 1997.
[11] G. Sabaliauskaite, F. Matsukawa, S. Kusumoto, and K. Inoue, "Further investigations of reading techniques for object-oriented design inspection," Inf. Softw. Technol., vol.45, no.9, pp.571–585, 2003.
[12] C. Sauer, R. Jeffery, L. Land, and P. Yetton, "The effectiveness of software development technical reviews: A behaviorally motivated program of research," IEEE Trans. Softw. Eng., vol.26, no.1, pp.1–14, Jan. 2000.
[13] L.G. Votta Jr., "Does every inspection need a meeting?," Proc. ACM SIGSOFT Symposium on Foundations of Software Engineering, ACM Software Engineering Notes, vol.18, no.5, pp.107–114, Dec. 1993.