<table>
<thead>
<tr>
<th>Title</th>
<th>Extended Metrics to Evaluate Cost Effectiveness of Software Inspections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Sabaliauskaite, Giedre; Kusumoto, Shinji; Inoue, Katsuro</td>
</tr>
<tr>
<td>Citation</td>
<td>Transactions on Information and Systems. E87-D(2) P.475-P.480</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2004-02-01</td>
</tr>
<tr>
<td>Text Version</td>
<td>publisher</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/11094/50957">http://hdl.handle.net/11094/50957</a></td>
</tr>
<tr>
<td>DOI</td>
<td></td>
</tr>
<tr>
<td>rights</td>
<td>copyright©2004 IEICE</td>
</tr>
</tbody>
</table>
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 defects 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 of $M_k$ 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 of $M_k$ 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.

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; $\Delta C_t$ – testing cost saved by inspection; Virtual testing cost – testing cost if no inspections are executed. By spending cost $C_r$ during inspection, the cost $\Delta C_t$ 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): $C_{DEF}$ – cost spent to detect actual defects during preparation; $C_{FP}$ – cost spent to detect false positives during preparation; $C_{DEF}$ – cost needed for testing to detect remaining defects; $C_{FP}$ – cost needed for testing to detect defects introduced by false positives during preparation.

In this model, by spending cost $C_{DEF}$ during inspection, the testing cost $\Delta C_t$ is being saved. However, by spending the cost $C_{FP}$ during inspection, the cost $C_{FP}$ is being added to the testing cost. Therefore, the costs $C_{FP}$ and $C_{FP}$ 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; $C_{mADDDEF}$ – cost spent to detect additional defects during inspection meeting stage; $C_{mADDFP}$ – cost spent to detect additional false positives during inspection meeting stage; $C_{mLOSTDEF}$ – cost spent to eliminate
actual defects detected during preparation stage; $C_{\text{mELIM FP}}$ – cost spent to eliminate false positives detected during preparation stage; $C_{\text{ADD FP}}$ – cost needed for testing to detect defects introduced by additional false positives detected during inspection meeting stage; $C_{\text{Lost DEF}}$ – testing cost needed to detect defects lost during inspection meeting stage; $\Delta C_{\text{ADD DEF}}$ – testing cost saved by additional defects detected during inspection meeting stage; $\Delta C_{\text{ELIM FP}}$ – testing cost saved by false positives eliminated during inspection meeting stage; $\Delta C_{\text{DEF}}$ – testing cost saved by defects detected during preparation and confirmed during inspection meeting stages.

In this model, cost $C_{\text{mELIM FP}}$ spent to eliminate false positives, reduces the testing cost by $\Delta C_{\text{ELIM FP}}$, and cost $C_{\text{mLOST DEF}}$, spent to eliminate actual defects, increases testing cost by $C_{\text{Lost DEF}}$. Inspection meeting cost spent to detect additional defects, $C_{\text{mADD DEF}}$, reduces the testing cost by $\Delta C_{\text{ADD DEF}}$; and inspection meeting cost spent to detect additional false positives, $C_{\text{mADD FP}}$, increases the testing cost by $C_{\text{ADD FP}}$. In other words, the costs $C_{\text{mLOST DEF}}, C_{\text{mADD FP}}, C_{\text{LOST DEF}}$ and $C_{\text{ADD FP}}$ are the costs lost by inspection meeting (Fig. 5).

Inspection meeting costs $C_{\text{mDEF}}$ and $C_{\text{mFP}}$, 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 $DEF_{\text{total}}$, and the number of defects found during inspection as $DEF_r$. Then we get the following equation:

$$M_f = \frac{DEF_r}{DEF_{\text{total}}}.$$  \hspace{1cm} (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:

$$M_c = \frac{\Delta C_r}{C_r}. \hspace{1cm} (2)$$

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:

$$M_k = \frac{\Delta C_t - C_t}{C_t + \Delta C_t} \hspace{1cm} (3)$$

$M_k$ 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 $\Delta C_t$ compared to the virtual testing cost ($C_t + \Delta C_t$) 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 $\Delta C_t$ is a testing cost saved by preparation, and costs $C_{\text{FP}}$ and $C_{\text{FP}}$ 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 $M_{\text{JDFV}}$ to evaluate the
**Preparation Losses**, which is the ratio of “cost lost by inspections” by “cost saved by inspections”:

\[ M_{\text{JDV}} = \frac{C_{\text{FP}} + C_{\text{DEF}}}{\Delta C_{i}} \]  

(4)

Not only the costs \( C_{\text{FP}} \) and \( C_{\text{DEF}} \), but also \( C_{\text{FP}} \) 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 \( M_{s\text{JDV}} \), 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:

\[ M_{s\text{JDV}} = \frac{\Delta C_{i} - C_{i} - C_{\text{FP}}}{C_{\text{DEF}} + \Delta C_{i} \Delta C_{i}} \]  

\[ = \frac{\Delta C_{i} - C_{\text{DEF}} - C_{\text{FP}}}{C_{\text{DEF}} + \Delta C_{i}} \]  

(5)

In metric \( M_{k} \) (see Sect. 4.3), the virtual testing cost (the testing cost if no inspection is executed) is defined as \( (C_{i} + \Delta C_{i}) \). However, inspection increases the testing cost by \( C_{\text{FP}} \) if false positives have been detected (see Fig. 4). Therefore, if no inspection is executed, the cost \( C_{\text{FP}} \) 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 \( (C_{\text{DEF}} + \Delta C_{i}) \). In addition, since the testing cost \( C_{\text{FP}} \) 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 \( M_{s\text{JDV}} = M_{k} \), otherwise \( M_{s\text{JDV}} < M_{k} \).

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: \( C_{\text{DEF}} \) and \( C_{\text{FP}} \). \( C_{\text{DEF}} \) is a testing cost spent to detect defects lost during inspection meeting, and \( C_{\text{FP}} \) 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 \( \Delta C_{\text{ADD}_F} \), and by eliminating false positives detected during preparation stage \( \Delta C_{\text{ELIM}_F} \).

**Inspection meeting costs \( C_{\text{DEF}} \) and \( C_{\text{FP}} \).** 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 \( C_{\text{ADD}_F} \) and \( C_{\text{ELIM}_F} \), spent to detect additional defects and to eliminate false positives, reduce testing costs by \( \Delta C_{\text{ADD}_F} \) and \( \Delta C_{\text{ELIM}_F} \) respectively. Costs \( C_{\text{ADD}_F} \) and \( C_{\text{ELIM}_F} \) spent to detect additional false positives and to eliminate actual defects, increase testing costs by \( C_{\text{DEF}} \) and \( C_{\text{ADD}_F} \) respectively. The purpose of the inspection meeting should be to minimise the costs \( C_{\text{ADD}_F} \) and \( C_{\text{ELIM}_F} \), and to maximise the costs \( C_{\text{DEF}} \) and \( C_{\text{ADD}_F} \).

Similarly as the metric \( M_{\text{JDV}} \) (Eq. (4)) to evaluate the losses of preparation stage, we introduce a new metric \( M_{\text{MEET}} \) 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):

\[ M_{\text{MEET}} = \frac{C_{\text{DEF}} + C_{\text{ELIM}_F}}{\Delta C_{\text{ADD}_F} + \Delta C_{\text{ELIM}_F}} \]  

(6)

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

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

\[ M_{s\text{MEET}} = \frac{\Delta C_{\text{DEF}} + \Delta C_{\text{ELIM}_F}}{C_{\text{DEF}} + C_{\text{ADD}_F} + \Delta C_{\text{DEF}}} \]  

(7)

**Metric \( M_{s\text{MEET}} \)** (Eq. (7)) is a modification of metric \( M_{k} \) (Eq. (3)). In Eq. (7), the virtual testing cost is defined as \( (C_{\text{DEF}} + \Delta C_{\text{ADD}_F} + \Delta C_{\text{ELIM}_F} + \Delta C_{\text{DEF}}) \), and the testing costs \( C_{\text{DEF}} \), \( C_{\text{ADD}_F} \) and \( C_{\text{ELIM}_F} \), caused by inspection, are added to the inspection cost.

6. Comparison of Proposed Metrics to \( M_{e} \) and \( M_{k} \) Metrics

The differences among \( M_{e} \), \( M_{k} \) 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 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
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 $M_{LJDV}$ to evaluate inspection losses, the value of $M_{LJDV}$ is 2.04 in Case I, 0.31 in Case II, and 0.2 in Case III. Thus, $M_{LJDV}$ identifies that inspection losses are the greatest in Case I.

The values of the extended cost effectiveness metric $M_{LJDV}$, 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 $M_{LJDV}$ 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 $M_l$ 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 $M_{LMEET}$ in Case IV is 2.75, and in Case V is 0.6. Therefore, $M_{LMEET}$ identifies that inspection losses are greater in Case IV as compared to Case V. The value of metric $M_{LMEET}$ is $−0.188$ in Case IV, and 0.0104 in Case V. Consequently, metric $M_{LMEET}$ is more precise than $M_k$ and $M_l$ 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 $M_{LJDV}$ (Eq. (4)) and $M_{LMEET}$ (Eq. (6)) to evaluate the Preparation Losses and the Inspection Meeting Losses respectively. Also, we have proposed two metrics $M_{LJDV}$ (Eq. (5)) and $M_{LMEET}$ (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


