Information and Software Technology 56 (2014) 495–505

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Performance appraisal of software testers

Tanjila Kanij [a,*], John Grundy [a], Robert Merkel [b]

[a] Swinburne University of Technology, Melbourne, Australia
[b] Monash University, Melbourne, Australia

ARTICLE INFO

Article history:
Available online 23 November 2013

Keywords:
Performance appraisal
Software testers
Project management

ABSTRACT

Context: To determine the effectiveness of software testers a suitable performance appraisal approach is necessary, both for research and practice purposes. However, review of relevant literature reveals little information of how software testers are appraised in practice.
Objective: (i) To enhance our knowledge of industry practice of performance appraisal of software testers and (ii) to collect feedback from project managers on a proposed performance appraisal form for software testers.
Method: A web-based survey with questionnaire was used to collect responses. Participants were recruited using cluster and snowball sampling. 18 software development project managers participated.
Results: We found two broad trends in performance appraisal of software testers – same employee appraisal process for all employees and a specialized performance appraisal method for software testers. Detailed opinions were collected and analyzed on how performance of software testers should be appraised. Our proposed appraisal approach was generally well-received.
Conclusion: Factors such as number of bugs found after delivery and efficiency of executing test cases were considered important in appraising software testers' performance. Our proposed approach was refined based on the feedback received.

1. Introduction

The reliability of delivered software, to a large extent, depends on the performance of software testers. An accurate performance appraisal of software testers is thus very important for their recruitment, monitoring and development, and for testing team performance management. Furthermore, from a research perspective, to conduct studies of factors that potentially affect software testers' performance, a validated, reliable instrument to assess software testers' performance is an essential prerequisite. For example, in a research study [1] investigating the influence of personality on the effectiveness of software testing the authors needed to distinguish different levels of performance. To accomplish this, a method to assess high performing software testers was necessary.

However, from an extensive search of relevant literature we did not find any widely accepted and well established performance appraisal method for software testers. Therefore, as an operational need for our research, we have proposed a new Performance Appraisal Form (PAF) for software testers. However, any such instrument should be validated for use.

In this study, we sought a broader insight into industrial practice in the area of tester performance appraisal by surveying nearly 20 software development project managers to describe the practices in their own organizations, and their own views on tester performance appraisal. We then attempted to validate the approach taken in our PAF by collecting feedback from software development project managers on the proposed PAF, obtaining detailed feedback from 10 of them. With this two-pronged approach, we not only sought direct comment on our proposed PAF, we hoped to find out whether industrial practice could further inform our PAF design, and also whether the PAF proposed could be of industrial as well as research interest.

The rest of the article is arranged as follows – Section 2 summarizes our review of relevant literature, Section 3 details our research questions, Section 4 describes the proposed Performance Appraisal Form (PAF), Section 5 describes the method of this research study, Section 6 presents our results, Section 7 lists the threats to validity of the research, Section 8 presents our discussion on the findings and finally Section 9 concludes the article.

2. Related work

2.1. Performance evaluation of software testers

As reported in our earlier research [2], there is no widely accepted instrument or approach to evaluating the performance of software testers in the academic literature. However, we found

[*] Corresponding author. Tel.: +61 392148840.
E-mail address: tkanij@swin.edu.au (T. Kanij).

0950-5849/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.11.002

some suggestions about criteria that may be important for evaluating software testers' performance. Fenton and Pfleeger [3] suggest measuring efficiency of software testing using the number of bugs found per KLOC. Grady and Caswell [4] suggest looking for average reported bugs per working day. However, Kaner [5] has discouraged considering only bug counts to measure software testers' efficiency as bug counts are influenced by reliability of code being tested, difficulty of testing the code, and the testing techniques being used (for example, exploratory and regression testing will produce different bugs).

Kaner [6] proposed a multidimensional assessment method for software testers, emphasizing qualitative assessment of testers' plan of testing, execution of tests and bug reports. He suggests the reviewer conducts short, regular discussions with testers regarding their test progress to obtain information. Kaner's proposed approach is a plausible way to evaluate testers; however, it is not supported by any research results. In addition, the evaluation approach is time consuming and is dependent on the perception of the reviewer. Appropriate manager training and experience is needed to successfully carry out this type of evaluation.

In our earlier survey [2] we found that bug report quality was considered important in assessing software tester's performance.

2.2. Performance evaluation of other software practitioners

Killingsworth et al. [7] described a model to motivate and evaluate information systems staff using five factors: product quality, customer outreach, staff development, administrative efficiency and fiscal responsibility. A senior project manager and team leader assess each employee on each of the five factors with varying weights for the reviewer. For example product quality accounts for 40% of the review for a team leader and 20% of the review for a senior manager.

Mayer and Stalnaker [8] describe a number of methods that are useful for selection and evaluation of computer personnel. While most of the methods presented in their paper are useful for selection of programmers, very few of those can be used for evaluation as well. These include: Dickmann's [9] Programmer Appraisal Instrument (PAI) with four performance areas – professional preparation and activity, programmer competence, dealing with people and adapting to the job; Bairdain's approach [10] considering the following factors – programming knowledge/capability, working style, temperament traits and personal professional items and Berger and Wilson's [11] Basic Programmer Knowledge Test (BPKT) evaluating programmer's knowledge on six areas – logic estimation and analysis, flow diagramming, programming constraints, coding operations, program testing and checking, and documentation.

Powell [12] presented 13 categories to be used to rate programmers and analysts. Some of the categories were tact and diplomacy, project selection, project planning, self expression – written and oral, ability to complete the job, and supervision. He defined each of the categories and proposed a distribution of performance according to his method for a group of 20 programmers and analysts.

The proposals for programmer assessment indicates similar instruments can be developed for software testers as well.

3. Research questions

Current methods utilized to appraise software testers in the software industry have not been reported or evaluated in any detail in the open literature [2]. We therefore do not know what are the relative advantages and disadvantages of current performance appraisal practices for software testers and how we can improve on these.

Based on our review of the literature and analysis of different requirements listed in job advertisements for testers, we designed a new Performance Appraisal Form (PAF) for software testers. We then wanted to collect feedback on the appropriateness of our proposed PAF and to make suggested improvements. A brief description of the proposed PAF is given in the following subsection. The PAF itself is available at:http://www.testingsurveys.org/PAF_static/initialPaf.html

In this study, we attempt to answer four research questions via a two-part practitioner survey:

- How is performance of software testers currently appraised in industry?
- What are the advantages and disadvantages of current performance appraisal methods used for software testers?
- How can the currently used performance appraisal methods for software testers be improved?
- What do software project managers think of our proposed PAF for software testers?

4. Proposed Performance Appraisal Form (PAF)

The objective of our proposed PAF is to provide a standard assessment instrument to assess overall performance of software testers from different performance dimensions. Some performance appraisal instruments use multiple forms to assess different aspects of employee performance. However, for simplicity we chose to design an integrated form. The performance dimensions of our proposed PAF were based on different approaches [13] to performance appraisal: *Performer focused appraisal:* This approach attempts to discern whether some qualities are exhibited by the performer or not.*Work behavior based appraisal*: This approach judges the performance on the work behavior of the performer. *Result focused appraisal:* This approach includes assessment of performance based on predefined goals and objectives.

For software testers, how effectively testing has been carried out and how efficiently the testing contributed to the reliability of the software, are important. We also believe there are some general skills that are important to be high performing software testers. The appraisal form, therefore include different rating dimension on work behavior, work outcome and personal attributes, with seven dimensions in total.

In order for the better the understand-ability of the readers, before describing the dimensions of our proposed PAF, we discuss some of the scale types used to evaluate on those dimensions. We have used three types of rating scale: behavior frequency scales, compare against standard scales and evaluation concept scales [13]. A behavior frequency scale considers the occurrence of defined behavior; labels such as "always", "seldom", and "never" and used. Standard scale type compares the performance against a standard with labels such as "exceeds standard", "below standard" and so on. The evaluation concept scale judges the quality of performance and the associated labels are typically "outstanding", "marginal", "unsatisfactory" and so on.

4.1. Dimensions related to work outcome

Two work outcomes considered to be included in the dimensions related to work outcome of a software tester are bug report and number of bugs reported (bug count).

4.1.1. Bug report

Kaner [6] emphasized on the qualitative assessment of bug report based on – ease of understanding, sufficient information to replicate the bug, short and precise description, absence of