Metrics for Evaluating the Performance of Software Tester and Testing Team

J.Vijayashree
VIT University,Vellore
vijayashree.j@vit.ac.in

Dr.B.Persis Urbana Ivy
VIT University,Vellore
persis.b@vit.ac.in

J.Jayashree
VIT University,Vellore
jayashree.j@vit.ac.in

Abstract - The major determining factor of software quality is the effectiveness of testing. The reliability of software project delivered depends on the performance of the individuals in charge for testing the software. Individual testers diverge in their effectiveness and efficiency. The factors contributing to this difference have not been so far well studied.

There are no standard methods for assessing the performance of software testers presently. Even there are no assessment criteria or metrics for evaluating the performance of software tester or software testing team. Literature survey reveals numerous different human-centric factors that have to be considered. However, the comparative prominence of these factors is unknown. In this paper few metrics were defined for measuring the performance of software tester and software testing team.

Keywords – tester, performance, team, factors

I. INTRODUCTION

The broad domain of software testing comprises different job responsibilities like creating test plans, running a variety of tests, documenting results.

Software testing denotes to the job of discovering faults, inspecting the fitness and appropriateness, and confirming the desired level of reliability of software systems before their deployment. Software testing is defined a process of executing test cases, which are cautiously designed using test case design techniques. Test-case design techniques target at confirming methodical coverage, uncovering of distinctive error types, and reduction of redundant testing.

During testing, the software tester searches for occurrences where the software does not perform according to its specification. Software bugs are identified during testing phase. For identifying bugs, software testers start with a test plan and then design appropriate tests according to the test plan. In order to fix the problem in software, these bugs need to be reported to the team of developers. Software developers perform necessary action based on the bugs identified. Re-testing is then performed by the testing team.

II. LITERATURE SURVEY

The objective of paper [1] is to find the different factors which helps in assessing the performance of the tester. The reliability of the delivered software completely depends on the individual responsible for it. Based on the personal opinion of the tester a survey is done in this paper. Here the number of fault identified by the tester is considered to be an important metric to assess the tester performance. The six factors considered for assessing are number of bugs found, quality of the bug report, severity of the bug, ability of bug advocacy and rigorousness of the test planning and execution.104 participants were there in this survey in that 72% are male testers. In that 28% of them are Indians and 24% of the testers are from US.60% of the testers are from large IT companies and 20% are from small IT company.31% of the respondent noted other factors like creativity, analysing capability, implementation of plan. Understanding are also important in there open ended question. The result of this survey shows that the factor number of bug found is least important and the factor quality of bug report to be the most important factor for assessing the tester's performance.

The main objective of paper [2] is to identify the important factor that determine the performance of the software testing team where the author have also the different types of team diversities have also been studied. The seven factors considered for this study are the performance of the individual tester, interpersonal skills, team playing capability, experience in testing, certification in testing, knowledge of specific problem domain and the compatibility with other team members. The survey also had 6 closed questions where the participants

were asked to rank the factors for 1 to 7. The result shows that 89% of the participants agreed that diversity helps in improving the performance. Participants thought that tester should be a good team player.4% of the respondent felt gender diversity and cultural diversities are to be considered. Team performance will be better when they have an experience working as a team.

Author [3] shares his experience while conducting a survey in testing industry. Four web based industrial survey is done with custom website. There was no face to face interaction in this completely survey. The four surveys are factors influencing software tester's performance, work log collection of software tester, testing manager opinion on a new performance appraisal form for software tester and personality of the software tester. Participants were recruited from LinkedIn and Yahoo groups through email. Participants were asked to sign in the CIS before participating in the survey. Number of participant's invitation sent to the number of response received is very less. So the response rate is very less. Author felt the nature of invitation also played an important role in the survey. In survey one and four the numbers of participants were high when compared to the second and third survey.

The goal of paper [4] is to determine the relationship between the performance in software testing and the specific personality traits. This study was done by software engineers and students. Individual testers vary in their effect but the author thought that this variation has not been studied. This study tried to investigate the individual tester's nature. The five models of personality factors being considered are extraversion, agreeableness, conscientiousness, neuroticism and openness to explore. These factors are tested with the help of metrics like bug location rate, weighted fault density and bug report quality. The result of this study shows that the tester who is very active are good in discovering and the testers who are careful are good in finding faults.

The study [5] is to understand how division of labours should be performed and test whether by adding additional testers will there be any increase in the number of defect.120 students participated in this experiment and performed manual testing. Time restriction and pressure are the two conditions imposed on them. Two hours fixed time slot for one group and another group can take as much as they needed for testing. Team of five time restricted testers used ten hours and detected 75% more defect than single non time restricted testers using ten hours. The result shows that multiple times pressured individual delivers high defect detection effect when compared to the non-time pressured individual. As time pressure increases efficiency also increase but effectiveness decrease. F score is the mean of effectiveness and validity. Effectiveness is the measure of share of unique defect found by a tester group. Validity is the share of valid unique findings among all findings. The number of unique defect increase as the number of testers increase. Time pressure has a positive effect on defect.

Paper [6] uses two research questions as what are the social and technical factors which influence the testing activity. How can the image of tester and testing can be improved in academic activities. The result shows that there is a low preference for students to take testing as future career.

III. PROPOSED METRICS

The main objective of this research is to evaluate the performance of the software tester and software testing team. Software tester's performance is one of the important factors which affect software project performance. For evaluating the performance of tester few metrics were defined. This metrics will also help testers to have an idea of software testing profession and will help them in deciding career objectives. The metrics defined are

1. Cost of finding a defect in testing (CFDT)
2. Test Case Adequacy
3. Test Case Effectiveness
4. Effort Variance
5. Schedule Variance
6. Schedule Slippage
7. Rework Effort Ratio
8. Review Effort Ratio
9. Requirements Stability Index
10. Requirements Creep
11. Weighted Defect Density

1. Cost of finding a defect in testing (CFDT)

Total time spent on testing including time to create, review, rework, execute the test cases and record the defects. This should not include time spent in fixing the defects.

Total effort spent on testing / defects found in testing

2. Test Case Adequacy: This defines the number of actual test cases created vs estimated test cases at the end of test case preparation phase. It is calculated as

No. of actual test cases / No. of test cases estimated

3. **Test Case Effectiveness:** This defines the effectiveness of test cases which is measured in number of defects found in testing without using the test cases. It is calculated as

   No. of defects detected using test cases*100/Total no: of defects detected

4. **Effort Variance** can be calculated as

   {(Actual Efforts-Estimated Efforts) / Estimated Efforts} *100

5. **Schedule Variance:** It can be calculated as

   {(Actual Duration - Estimated Duration)/Estimated Duration} *100

6. **Schedule Slippage:** Slippage is defined as the amount of time a task has been delayed from its original baseline schedule. The slippage is the difference between the scheduled start or finish date for a task and the baseline start or finish date. It is calculated as

   ((Actual End date - Estimated End date) / (Planned End Date – Planned Start Date) * 100

7. **Rework Effort Ratio:**

   {(Actual rework efforts spent in that phase / Total actual efforts spent in that phase)} * 100

8. **Review Effort Ratio:**

   (Actual review effort spent in that phase / Total actual efforts spent in that phase) * 100

9. **Requirements Stability Index:**

   {1 - (Total No. of changes /No of initial requirements)}

10. **Requirements Creep:**

    (Total No. of requirements added / No of initial requirements) * 100

11. **Weighted Defect Density:**

    WDD = (5*Count of fatal defects)+(3*Count of Major defects)+(1*Count of minor defects)

IV. CONCLUSION

It has been empirically proven that software projects performance is mainly affected due to poor software testing. Poor software testing is directly proportional to the software tester performance and software testing team performance. In order to find the effective methods for measuring the performance of software tester and the testing team, literature survey is made and few metrics have been defined in this paper. As future work, we intend to perform a nonparametric factor analysis to further investigate possible factors affecting confirmation.

REFERENCES

[1] T.Kanij, R.Merkel, and J.Grundy, “A preliminary study on factors affecting software testing team performance”, In International Symposium of ESEM, 2011, pp. 359-362.

[2] Pak-Lok Poon, T.H.Tse, Sau-Fan, Fei-Ching , “Contributions of tester experience and a checklist guideline of the identification of categories and choices for software testing”, In Software Quality Journal, 2011, pp. 141-163.

[3] T.Kanij, R.Merkel, and J.Grundy , “Performance Assessment Metrics for Software Testers”, In International workshop on CHASE, 2012, pp. 63-65.

[4] T.Kanij, R.Merkel, and J.Grundy , “Lesson learned from conducting industry survey in software testing”, In International Workshop on CESI, 2013, pp. 63-66.

[5] T.Kanij, R.Merkel, and J.Grundy , “An Empirical study of the Effect of Personality on Software Testing”, In International workshop on CESI, 2013, pp. 239-248.

[6] Mika V. Mantyla, Juha Itkonen, “More testers – The effect of crowd size and time restriction in software testing”, In Journal of Information and Software Technology, 2013, pp. 986-1003.

[7] Anca Deak, “Understanding the influence of social and technical factors testers in software organizations”, In International Conference on ICST, 2013, pp. 511-512.