*SUMARRY1*

*Measuring Software Testability Modulo Test Quality*

***1:Authors:***

*Valrio Teragani*

*Pasquale salsa*

*Mauro puzze*

***2:Universities:***

*USI Università della Svizzera italiana Switzerland* [*valerio.terragni@usi.ch*](mailto:valerio.terragni@usi.ch)

*University of Zurich Switzerland* [*salza@ifi.uzh.ch*](mailto:salza@ifi.uzh.ch)

*USI Università della Svizzera italiana Schaffhausen Institute of Technology Switzerland* [*mauro.pezze@usi.ch*](mailto:mauro.pezze@usi.ch)

***DATE:*** *ICPC ’20, October 5–6, 2020, Seoul, Republic of Korea*

***INTRODUCTION:*** *Software testing is an essential, labor-intensive and time-consuming activity of the software lifecycle. Making testing easier is important for many software companies, as it lowers development costs while increasing the number of detected faults.* *A software system with a high degree of test ability results in slow test effort. In their recent comprehensive literature review of 208 papers on software test ability.* *Most studies on measuring and predicting test ability investigate on the relation between class-level metrics in object oriented systems* *, but suffer from two limitations: (i) the data sets are of small size, and (ii) mostly ignore the quality of the test suites.*

***Small sample size****. Previous studies involved at most eight software projects [22]. Such a small number of analyzed projects does not guaran teeth ability of there results: specific development styles, frameworks, and practices can influence the correlation results and produce different results for different projects[3]. Ignoring the test quality. Previous studies measured the test effort in terms of the size of the test classes, while mostly ignoring.*

***Design properties and test effort;*** *• publicly releasing our data set for further studies 2. The paper is organized as follows. Section 2 presents the objective of this study, introduces the considered metrics, and motivates and explain sour normalization procedure. Section3 presents the results, which address the main research questions that validate the hypothesis that “ normalizing by the test quality” achieves better correlation than not using the normalization. Section4 discusses there work. Section5 summarizes the main results presented in the paper.*

***ABSTRACT:***

*Comprehending the degree to which software components support testing is important to accurately schedule testing activities, train developers, and plan effective refactoring actions. Software test ability estimates such property by relating code characteristics to the test effort.* *The results confirm that normalizing the test effort with respect to the test quality largely improves the correlation between class metrics and the test effort. Better correlations result in better prediction power, and thus better prediction of the test effort. This paper contributes to a better comprehension of software test ability by:*

*• presenting the by-far largest study on the correlation of class and test-effort metrics in terms of analyzed metrics, classes and projects;*

*• extending the test ability measurements by normalizing the test effort, with respect to the quality of the test suites;* ***\*****showing that the proposed normalization improves the correlation between class metrics and test effort.*

*• giving important in sight son software test ability that confirm some of the findings of previous studies as well as uncover.*

***CONCLUSIONS:*** *This paper proposes a new software test ability approach that extends current practice with the novel idea of normalizing test effort with respect to test quality. It also presented the results of an extensive study that involves 9,861 pairs of Java classes (with a total of 1,594,309 lines of code ) and corresponding Unit test cases taken from 1,186 GitHub projects. Our results indicate that normalizing test effort with test quality largely increases the correlation between class metric and test effort. An improved correlation between class metric and test effort means a better prediction of test effort.* *In this paper, we introduced the normalization process under the assumption optional growth of the test effort with respect to the test quality. For example, if five test cases (T-NOT = 5) achieve a branch coverage of 50%, our normalization assumes we need ten test cases (T-NOT = 10) to have a branch coverage of 100%. One avenue for future work is to study the impact of this assumption on the correlation between class and test-effort metrics.*