The Use of the Software Metrics in Practice

LUKA PAVLIČ, University of Maribor
MOJCA OKORN, University of Maribor
MARJAN HERIČKO, University of Maribor

In this paper, we address the field of software quality, software quality measurement and software metrics. The aim of our research was to check if and in what extend developers in Slovenia use software metrics. We base our research on the surveys, which were conducted not only in Slovenia, but also abroad. This is how we were able in addition to collect the data, also to compare the results. Primarily, we learnt the current state of software quality and quality assurance in Slovenia. Based on comparisons with the two previous surveys we find that the results for Slovenia do not differ significantly with them. We were able to found larger deviations only by combining obtained quality estimate into overall quality estimate, which is a commoner practice abroad than in Slovenia.

1. INTRODUCTION

The software has been present since 1842, when Ada Lovelace wrote the first computer program [Hiskey 2011]. Since then, the use of software has expanded into different areas of application, making it increasingly complex due to the continuous increase in user requirements and expectations. The software is used in a variety of critical areas, which are crucial for business success and / or security of people [ISO/IEC 9126 2011], for example for business purposes, in ubiquitous systems, health systems, education, and the military, therefore a good quality of software is crucial. Key factors in ensuring the proper quality of the software are a comprehensive description and quality evaluation, which can be achieved by determining the expected, required, quality attributes in which the purpose of using the software must be taken into account, and each of the mentioned properties must then be evaluated using validated and generally accepted metrics [ISO/IEC 9126 2011]. There are many different models to measure software quality.

The concept of software quality has many different definitions. According to ISO / IEC 9126, the quality is a set of the characteristics of the entity that relates to the ability to meet the requirements [ISO/IEC 9126 2011]. In the literature, various authors determine different quality factors, while the ISO / IEC 9126 standard determines functionality, reliability, usability, efficiency, maintenance and portability as quality factors. The listed factors are divided into sub-factors, which can be measured with internal or external metrics. The metric is a measurement method and a measurement scale [ISO/IEC 9126 2011], which gives quantitative results. The step towards quality software is measuring it, and it is measured using metrics, the results of which need to be combined in a general quality assessment later. When measuring, it is also important to choose the appropriate quality model or the standard after which quality measurement is carried out. There are many different models and quality standards that differ in terms of use. It is not appropriate to use every model or standard in all areas. Models and quality standards need to be adapted to each project or program according to needs.

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0057).

Author's address: University of Maribor, Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, SI-2000 Maribor, Slovenia; email: luka.pavlic@um.si.

In: Z. Budimac (ed.): Proceedings of the SQAMIA 2018: 7th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications, Novi Sad, Serbia, 27–30.8.2018. Also published online by CEUR Workshop Proceedings (http://ceur-ws.org, ISSN 1613-0073)

The aim of this paper is also to determine the status of measuring and quality assurance of software in companies in Slovenia. We are also interested in finding out the importance of the quality of to the companies in Slovenia. Furthermore, we would like to see, what approaches they use, how much of the companies in Slovenia measure the quality of the software. Finally, we would like to compare the results of companies in Slovenia with existing research abroad.

The structure of this paper is as follows. In the next section, we shortly summarize software quality and software quality metrics. This section gives a reader context in which we execute our research. Section 3 references similar research already conducted abroad. Existing surveys are a base for our survey. In section 4, we summarize main results and discuss them in final section 5.

2. SOFTWARE QUALITY AND SOFTWARE QUALITY METRICS

According to ISO standard, the quality is a set of the characteristics of the entity that relates to the ability to meet the requirements and requirements [ISO/IEC 9126 2011]. Quality Measurements, however, indicate if the techniques used really improve the software. Importantly, how the quality of the process affects the quality of the software [Kitchenham et. al. 1996]. Key factors in ensuring the quality are comprehensive specifications and software quality measurement, which can be achieved by considering the purpose of using the software and by defining the appropriate quality properties that need to be determined and evaluated by validated and generally accepted measurements, if only possible [ISO/IEC 9126 2011].

In the industry, we do not have a consensus about an effective, generally accepted or established basic definition of productivity and quality of software. The definition of software quality must be unambiguous, so that it can be predicted and measured before issuing, so that it can be defined and cannot be completely subjective [Jones 2016].

Software quality is typically understood as a hierarchical structure consisting of factors, sub-factors that aggregate attributes. The hierarchy is not balanced, as some attributes can contribute more than one sub-factor. [ISO/IEC 9126 2011] In doing so, we divide internal and external attributes and quality factors. E.g., reliability can be measured externally by observing the number of errors within a certain period of implementation in the software test period, and internally by reviewing the detailed specifications and source code to assess the degree of error tolerance. Internal attributes should be external attribute indicators. An internal attribute can affect one or more factors and more than one attribute can affect one factor.

Sub-factors are measured by internal or external metrics. The correlation between internal attributes and external metrics is never balanced, and the effect an internal attribute has on an associated external metric will be determined based on experience and will depend on the particular context in which the software is used. In the same way, external properties (such as relevance, accuracy, fault tolerance or time response) will affect the observed quality. The impact on quality failure in use (for example, the user cannot complete the task) can be traced to external quality attributes (such as suitability or operability) and related internal attributes that need to be changed [ISO/IEC 9126 2011].

Internal metrics can also be used for non-executable software (such as specifications or source code) at the design and coding stage. The primary purpose of internal metrics is to ensure that the required level of external quality and quality is achieved. Internal metrics allow users, assessors, testers, and developers to evaluate software quality and address quality questions before software becomes feasible. Internal metrics measure internal attributes or, by analysing static properties, intermediate or affordable software points out external attributes. Measurements of internal metrics use numbers or frequencies of compound components of software that occur, for example, in the source code, in the control graph, in the data flow, and in the representations of the transitions between states.

External metrics apply software criteria derived from the system behaviour measures it is part of, by testing, operating, and observing feasible software or system. Before purchasing or using the software, it must be assessed using metrics based on business goals related to the use and management of software in a particular organizational and technical environment. These are mainly external metrics. External metrics provide users, assessors, testers and developers to be able to evaluate software quality during testing or operation. Some of the examples of these metrics are functional adequacy, functional implementation completeness, precision, fault density, etc. [ISO/IEC 9126 2011].

3. RELATED RESEARCH

The aim of our research is to determine, using the survey, how and to what extent the quality of software is measured in Slovenia. This is why we performed a literature review. We captured about thirty papers using Google Scholar, ScienceDirect and IEEE Xplore databases. Firstly we excluded all that were older than 10 years in order to compare the results as best as possible. Then, we excluded studies that did not contain surveys, as our purpose was to compare the results of the surveys. We then eliminated all those who did not contain questions about the use of models, which we labelled as the most commonly used or those that did not contain questions about the use of metrics. We also eliminated all the studies that were not general enough and covered only one of the models or standards, such as the study Quality Measurement of a Software Product: an ISO / IEC 9126 survey [Jung et. al. 2004] covering only questions on the ISO standard / IEC 9126. To improve interpretation and subsequent comparison of the results, we finally selected two studies in which they conducted a survey on the same topic. We selected the study, conducted by Gary Lynn Wilson Jr. as part of his master's thesis [ISO/IEC 9126 2011] and the survey of Software Quality Models in practice, which was carried out with the help of a survey at the Technical University of Munich at the Institute of Informatics [Kitchenham et. al. 1996].

In order to make a better comparison, we used questions in the survey on the basis of the already conducted surveys. We tried to change the questions as little as possible, so that the results of the comparison were as correct as possible, and we supplemented them with our own questions. We changed the questions about the metrics that were in use, by using the literature review and adding other metrics from existing surveys.

This paper presents a preliminary part of results, acquired during master thesis research "Measuring Software Quality" [Okorn et. al. 2016].

4. THE RESULTS

We conducted the survey online using Google forms. The questionnaire contained 28 questions. We received 61 responses over a period of five weeks. Let us highlight some the most interesting questions and results.

First, we asked respondents how many years of experience they have with the development of software or information solutions. The average age of the respondents' experience is 13.5. The minimum number of years of work experience they have entered is 0 and the highest is 35 years.

The majority of respondents who participated in the survey in Slovenia responded that they work in the company with up to five people. Another most frequent answer is that companies share up to 50 employees and up to 250 employees.

In the question "What type of software are you developing?" (See figure 1) respondents had more options. Most respondents who participated in the survey in Slovenia are developing business information solutions, which are 73.8%, followed by mobile or web solutions with 59%, followed by third-party software with 16.4%, software for embedded systems.

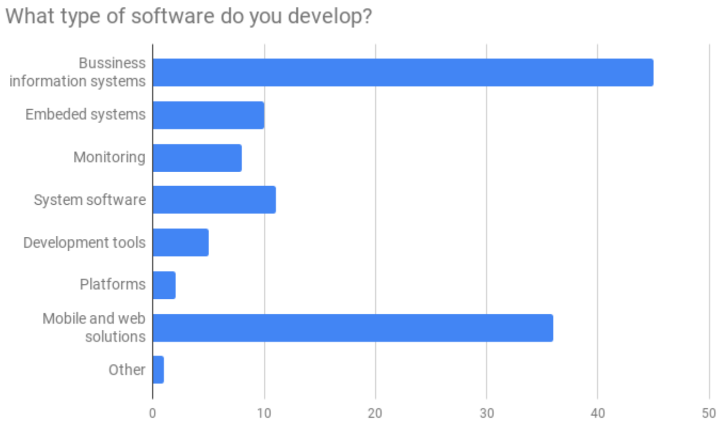

Fig. 1. Majority of developers were in business, mobile and web divisions.

To the question "Does your team use metrics (such as code range, nesting depth, collision) in order to improve the quality of the code?" respondents could respond with "yes", "no", or "do not know". Most, 50.8% of the respondents who participated in the survey in Slovenia replied that they do not use metrics, for the purpose of improving the quality of the code. 37.7% of respondents replied that they were using, while 11.5% of the respondents did not know. Shares are shown in Figure 2.

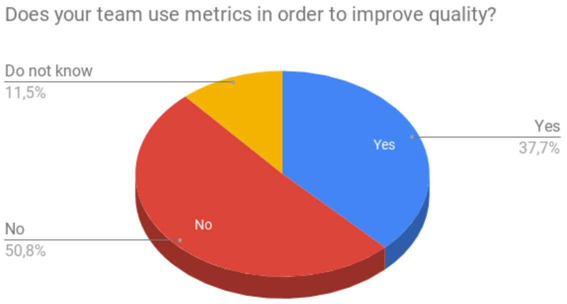

Fig. 2. Less than half of developers use metrics.

The next question was "How often does your team use metrics to evaluate the following areas?". The respondents had the following areas: software scope, error tracking, test cases, documentation suitability, software complexity, maintenance effort, software modularity. The rankings included ratings from 1 - Never to 7 - Continuous, meanwhile the average score was 4 - Average.

The results are shown in Figure 3, which shows that most respondents answered, in all areas, that they never use an evaluation metric. The second most common choice was the grade 4 - Average.

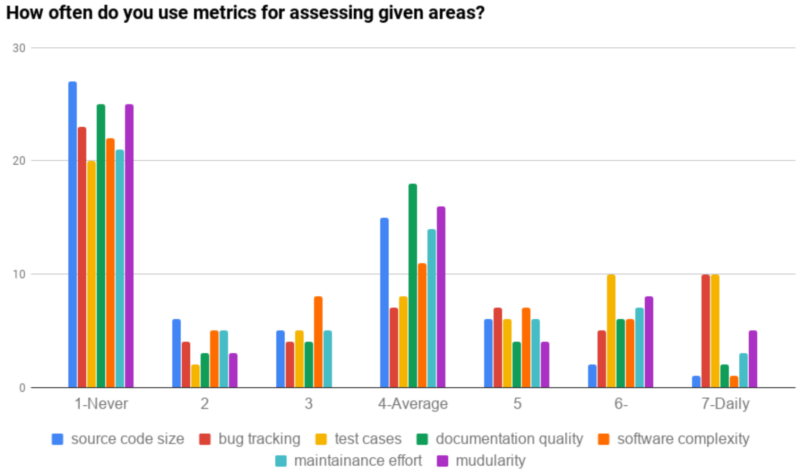

Fig. 3. Metric types and frequency of use.

Let us highlight a question that was dealing with concrete metrics - "Knowledge and usage of metrics for object - oriented structural attributes and measurements". The respondents had the following metrics: Coupling, Cohesion or Lack of Cohesion of Methods, Depth of the inheritance tree, Number of Children, Response for a Class. The scale included ratings: I've never heard of, or do not know, I know, but I do not use them, I've already used them, I regularly use it.

The results are shown in Figure 4, which shows that the majority of respondents answered that they never heard of metrics or did not know them, followed by answers that respondents know metrics but do not use them. The fewest respondents were those who replied that metrics were regularly used. Those who regularly use metrics use mostly the inheritance depth metric and coupling metric.

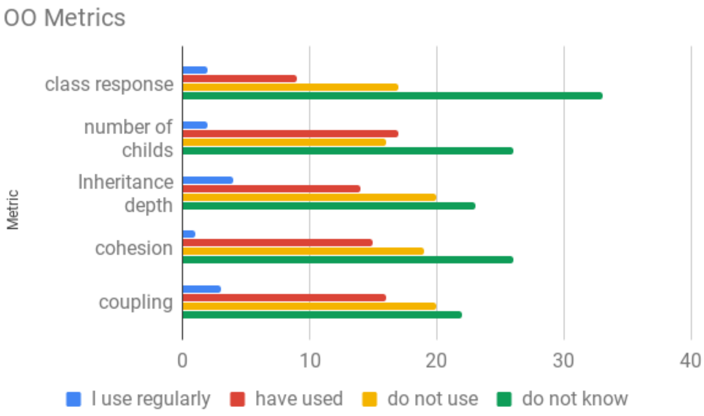

Fig. 4. OO metrics in practice.

5. DISCUSSION

In this paper, we presented the concept of software quality and the importance of software metrics. By reviewing related works, we selected and adapted existing experiments on research on the use of metrics in practice. In the survey, we collected general information about the respondents. The average experience of respondents in the field of software development is 13.5 years, the majority of respondents are software developers, most of them are surveyed in companies with up to five employees, and the development team counts up to five members on average. Most respondents answered that they are developing business information solutions, and the most frequently mentioned domain are telecommunications.

A little over half of the respondents do not use metrics in order to improve the quality of the code, a good third of the surveyed metrics are used. Respondents mostly answered that they never use metrics to evaluate areas such as test cases, error tracking, and software modularity. The metrics most frequently used by the respondents are related to unit testing or the ratio of successful / failed tests, the lines of code, the percentage of errors detected by users, the average time between failures / errors, the percentage of duplicate code, number of classes, number of methods. We also found that there is little knowledge of metrics in Slovenia.

The results obtained with the survey carried out in Slovenia are similar to those of the Technical University of Munich [Wagner et. al. 2014] and the results of the Wilson survey [Wilson 2013]. However, there have been differences in some issues. The population in which the survey was conducted in Slovenia is similar to those of foreign ones. Therefore, it was reasonable to make a comparison between the results.

In the survey in Slovenia, the majority of respondents adapt the models for each product / project. In the survey in Slovenia and in the Technical University of Munich, we found that testing is the most important technique for measuring all attributes of quality. According to the results of the Technical University of Munich, testing is the most important for portability and maintenance, while in our country for reliability and performance. The Wilson Survey found that for respondents the most important measure were deadlines, while respondents in Slovenia marked quality as the most important attribute of development. In the two polls, the respondents considered the cost to be the least significant attribute. In the University of Munich Technical Survey, compared to ours, many respondents answered that they combine the acquired quality ratings into a general quality assessment. As in the Wilson Survey, we also obtained similar results in the survey in Slovenia regarding the most frequently used metrics. The respondents regularly use the same metrics as in the Wilson survey. The biggest difference is in the metric for the average time between failures / errors and metrics for the percentage of errors detected by users, as they are one of the most commonly used on the scale in Slovenia. In Wilson it is among the least used.

In comparison, we allow the possibility that there have been differences due geographical context. In Wilson, they carried out the survey in the United States, while the Technical University of Munich's survey was carried out in German-speaking areas. Likewise, differences in results can be due to differences in years, Wilson passed the master's thesis in 2013, but the years of the survey of the Technical University in Munich are not traceable, but the latest source they said were from 2009.

In the future, the survey could be carried out on a wider population, correlations could be used to process and compare results, and the questionnaire could include even more questions about approaches to measuring quality. Interviews would be very suitable for obtaining the results. Additionally, questions about the visualization of software metrics could be added.

REFERENCES

D. Hiskey, “In 1842, Ada Lovelace Wrote the World’s First Computer Program.” [Online]. Available: http://www.todayifoundout.com/index.php/2011/02/in-1842-ada-lovelace-wrote-the-worlds-first-computer-program/.

ISO Standard, “ISO/IEC 9126 Software engineering - Product quality,” vol. 2001, 2001.

B. Kitchenham and S. L. Pfleeger, “Software quality: The elusive target,” 1996.

C. Jones, “The Mess of Software Metrics,” 2014. [Online]. Available: http://namcookanalytics.com/mess-software-metrics/. [Accessed: 17-Mar-2016].

H. W. Jung, S. G. Kim, and C. S. Chung, “Measuring software product quality: A survey of ISO/IEC 9126,” IEEE Softw., vol. 21, pp. 88–92, 2004.

S. Wagner, K. Lochmann, S. Winter, A. Goeb, M. Kläs, S. Wagner, S. Winter, and A. Goeb, “Software Quality Models in Practice: Survey Results,” 2012. [Online]. Available: https://mediatum.ub.tum.de/doc/1110601/1110601.pdf. [Accessed: 17-Mar-2016].

G. Wilson Lynn, “An Empirical Study on Software Quality: Developer Perception of Quality, Metrics, and Visualizations,” 2013. [Online]. Available: https://repositories.lib.utexas.edu/handle/2152/22606. [Accessed: 17-Mar-2016].

M. Okorn, M. Heričko, “Merjenje kakovosti programske opreme”, Master Thesis, Faculty of Electrical Engineering and Computer Science, 2016. [Online]. Available: https://dk.um.si/IzpisGradiva.php?lang=slv&id=58029