

ARTIFACT DESCRIPTION

A. Name of Artifact

Metrics Table used in Maintenance Effort Estimation for Open Source Software

B. Artifact License

Copyright (c) [2016] [Metrics used in Maintenance Effort Estimation for Open Source Software] Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SCORECARD

C. Insightful

Open Source Software has become increasingly popular in the past decade. According to Gartner, open source will be part of mission-critical software portfolios across 99% of Global 2000 enterprises by 2016. They also predict that at least 70 percent of new enterprise Java applications will be deployed on an open source Java application server by the end of 2017. Referring to findings of Black Duck Software and North Bridge Partners survey, key open source software trends that will shape the future of enterprise IT include 41.70% of people plan to deploy an open source solution in 1-2 years; more than 56% of companies using open source software will collaborate with competitors; more than 50% of all purchased software will be open source in 5 years. Since a large portions of software development effort goes into software maintenance, our artifact provides researchers and practitioners in the open source community with a thorough reference of what others have done in this area in the past 15 years. The artifact contains metrics from the most recent works in the area of open source software maintenance effort estimation. We extracted open source software maintenance effort estimation metrics from a carefully conducted systematic mapping study. The study consists 29 papers, which were selected from 3,312 papers that were published between 2000-2015 on the relevant topics.

There are 85 metrics classified according to their categories and content.

D. Useful

The purpose of this artifact is to provide full access for any researchers or practitioners to learn the most recent technique and approaches to estimate maintenance effort for open source software. Going through papers and then analyzing the metrics mentioned in them is very complex and requires lots of effort. Since the metrics table covers the most relevant studies in the area of maintenance effort estimation for open source software, researchers and practitioners won't need to reproduce and reclassify metrics that were included in the table. This artifact is free and available online.

E. Usable

All the metrics are organized in a table and each metric is explained with the following:

- **Level** describes the category of the metric.
- **Content** describes the content of the metric.
- **Name** describes the name of the metric.
- **Description** explains the metric in details.
- **#** records the number of occurrences of the metric.
- **Indirect effort, Direct effort, Code review, Duplicates, Bug fixing, Contribution, Guideline** are the topics from the selected studies.
- **T** indicates that the metric was used in the paper.

The metrics table is available via: <http://goo.gl/0qMHQp>.

ACKNOWLEDGMENT

This material is based upon work supported by National Natural Science Foundation of China No.91318301, 61432001, 91218302. It is also supported by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. It is also supported by the National Science Foundation grant CMMI-1408909, Developing a Constructive Logic-Based Theory of Value-Based Systems Engineering.

Metrics used in Maintenance Effort Estimation for Open Source Software

Hong Wu^{1,2}, Lin Shi^{1,4}, Celia Chen⁴, Qing Wang^{1,2,3}, Barry Boehm⁴

¹Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing, China

²University of Chinese Academy of Sciences, Beijing, China

³State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

⁴Center for Systems and Software Engineering, University of Southern California, Los Angeles, U.S.A.

Email: {wuhong, shilin, wq}@itechs.iscas.ac.cn, {qianqiac, boehm}@usc.edu

I. INTRODUCTION

Open Source Software (OSS) is distributed and maintained collaboratively by developers all over the world. However, frequent personnel turnover and lack of organizational management makes it difficult to capture the actual effort from their developers. Various OSS maintenance effort estimation approaches have been developed to provide a way to understand and capture the actual development effort. The goal of our study is to identify the current state of art of the existing maintenance effort estimation approaches for OSS. We performed a systematic literature review on the relevant studies published in the period between 2000-2015 by both automatic and manual searches from different sources. 29 out of 3,312 papers were selected based on a well designed selection process for this study. We extracted data and synthesized it against the research questions. Our study allowed identifying the rigor of the evidence with respect to the different estimation methods towards maintenance effort of OSS projects. One of artifact of the study is the extracted metrics that have been used by current literature. In Section II, we will describe the artifact, and in Section III we will describe how one can use it.

II. THE ARTIFACT: METRICS USED IN MAINTENANCE EFFORT ESTIMATION FOR OPEN SOURCE SOFTWARE

Current studies used various metrics or factors as inputs of OSS maintenance effort estimation methods. We extracted 85 metrics that have been used in estimating maintenance effort for open source software from 29 careful selected studies. The list of the 29 studies can be found on:

<http://goo.gl/gUii20>.

The 85 metrics are classified according to the category and their content as follows.

- **Project.** This category includes metrics that statistically described the characteristics of the entire project. There are six aspects in terms of content: size, time, task, bugs, commits, and developers.
- **Changes.** This category includes quantitative measurements that gave insight into characteristics of source code in patches. There are eight aspects: Cyclomatic complexity (CC), function, LOC, file, class, operator, parameter, and return.

- **Issue report.** This category includes metrics derived from issue reports. There are six aspects: basic information (info), stakeholder, time, similarity, process, and quality.
- **Participant.** This category includes metrics that can quantitatively represent the experience and the capability of the participants involved in the OSS maintenance activity. There are four roles: bug reporter, bug owner, bug trigger, and bug collaborator.
- **Community.** This category includes metrics about the OSS community characteristics. There are three aspects: contributor, activity, and workload.

For each metric, we provide the name and the description along with the studies that have used this metrics before. The full table of the 85 metrics can be found on: <http://goo.gl/NR3BEh>

III. HOW TO USE THE METRICS

The 85 metrics can benefit the both the Open source studies and industries that adopt open source software.

A. Researchers

For researchers who devote to build the quality models of open source software, the 85 metrics can serve as candidate measurements according to different goals. For example, active programmers, community activity, and open bugs can be used as single or composed measurements to evaluate the quality to the community of the open source software.

For researchers who dedicate to estimate the cost of open source software, the 85 metrics can be used as candidate drivers.

B. Practitioners

Open source software are widely used by industries as a part of their in-house product. However, open source software are not as free as zero-cost when integrating. Extra cost such as maintenance cost and integration cost need to be considered by practitioners. For practitioners who need estimate the cost of adopting open source software, the 85 metrics can serve as candidate factors.

However, the difficulties that people might encounter in using the 85 metrics is that the definition of one metric may slightly different among existing literature, and they need trace back to the literature according to the ID to confirm.