# Maintenance Effort Estimation for Open Source Software: Current trends

Chaymae Miloudi [1], Laila Cheikhi [1], Alain Abran[2] and Ali idri [1]

[1] *Software Project Management Team, ENSIAS, Mohammed V University in Rabat, Morocco*
[2] *Department of Software Engineering & Information Technology, École de Technologie Supérieure Montréal, Canada*

**Abstract**
Software maintenance of Open Source Software (OSS) has gained more attention in recent years and facilitated by the help of the Internet. Since volunteers in OSS do not record the effort of their contribution in maintenance tasks, researchers have to indirectly estimate the maintenance effort of such software. A review of the published OSS-MEE models has been performed using a set of 65 selected studies in a Systematic Mapping Study (SMS). This study analyses, discusses the state of the art about O-MEE and identifies trends through five additional Mapping Questions (MQs). In summary, various maintenance effort estimation (MEE) models were developed for OSS or industrial software. Researchers have mostly expressed the maintenance effort in terms of bug fixing, bug resolution time and severity in conjunction with bug report attributes. Regression Analysis and Bayesian Networks were most used estimation techniques, Recall, Precision, $R^2$ and F-measure evaluation criteria in addition to k-fold cross validation method. Most of the models were implemented using WEKA, R software and MATLAB. More than half of the selected studies lacked of any validity analysis of their results. Trends are also discussed to identify a set of implications for researchers.

**Keywords**
Maintenance, effort estimation, open source software, models

## 1. Introduction

Software maintenance "sustains the software product throughout its operational life cycle. Modification requests are logged and tracked, the impact of proposed changes is determined, code and other software artifacts are modified, testing is conducted, and a new version of the software product is released" [1]. Therefore, the maintenance activities consume the major part of software lifecycle effort and cost [2], which motivated researchers to propose a number of Maintenance Effort Estimation (MEE) models to manage this effort and provide easily modifiable software at less cost.

With the help of the Internet, open source software (OSS) development has emerged and has been evolving through the availability of high quality of software professionals in different countries [3]. Although, for several decades, there has been a growing number of software development effort estimation models, it is hard to find models proven suitable for MEE since most maintenance participants in OSS projects do not record their effort in effort recording systems. Therefore, a number of researchers have had to substitute of effort as inputs to their O-MEE models. A recent review of the published O-MEE models was performed in terms of a Systematic Mapping Study (SMS) [4] in 2022, with 65 studies from 2000 to June 2020. This SMS looked at five mapping questions (MQs) related to the publications channels and venues, datasets, research approaches, estimation techniques, and metrics used as independent variables as well as dependent variables.

The new study reported in this paper uses the same set of studies selected in the 2022 SMS and additional MQs related to the following aspects: OSS software types, estimation techniques with respect to accuracy criteria and validation methods, relationships independent-dependent variables, tools used

IWSM 2022, September 28–30, 2022, Izmir, Turkey
EMAIL: chaymae_miloudi@um5.ac.ma (A. 1); laila.cheikhi@um5.ac.ma (A. 2); alain.abran@etsmtl.ca (A. 3); ali.idri@um5.ac.ma (A. 4)

to generate estimates, and validity approaches. The main contribution of this study is to classify the selected studies based on the above aspects, provides a broader state of the art in O-MEE and discusses trends by means of five additional MQs.

The rest of this paper is organized as follows. Section 2 presents the related work, Section 3 the mapping questions and the data extraction form of this study, Section 4 the analysis of the findings for each MQs, and Section 5 a discussion on these findings and trends. Section 6 concludes this study.

## 2. Related work

In the literature, two review studies have focused on O-MEE; the systematic literature review (SLR) [5] was published in 2016 and includes 29 studies collected from 2000 to 2015, and the recent SMS [4] published in 2022, with 65 studies from 2000 to June 2020. This section summarizes the initial version of the SMS performed in [4].

The methodology adopted for the SMS includes four main steps. The first one consists on selecting the MQs with respect to the studied aspects. The second step consists on the identification of key terms used to construct the search string, such as: Open source, Maintenance, Effort, Estimation, Software, and Empirical. For each key term, a set of synonyms were identified. This search string was used to search automatically in the following digital databases: IEEE Xplore, ACM Digital Library, Science Direct, Springer Link, and JSTOR to identify candidate primary studies. Next, citation searching was also performed to identify more relevant studies. The third step consists on the selection of studies: included ones were empirical primary studies addressing O-MEE whether by proposing new models or performing empirical evaluations or comparing models. All the studies that are out of the scope of this study were rejected. The fourth step consists on extracting data and summarizing it in a MS Excel Sheet. Next, to facilitate the results analysis, the extracted data was investigated using different synthesis methods such as tables and graphs.

The selection process results is summarized as the following. First, a set of 18064 primary studies was retrieved by applying the search string on the digital databases by performing an automated search. Next, manual selection and duplication removal resulted in 410 studies. Then, studies out of the scope were excluded, which resulted in 54 studies. The citation searching process resulted in 11 additional studies. The final set is 65 studies.

With respect to the addressed MQs, the findings are as the following: Conferences were most targeted channels, followed by journals and workshops. Proposing new models and evaluating empirically their performance were the purpose of all selected studies, while only a few focused on comparing models performance. Eclipse, Mozilla, Firefox Apache, and Gnome open source software have been used as sources for building the datasets. Regression analysis, Bayesian Networks and Decision tree techniques were the most used. The most used independent variables are related to bug report attributes and size metrics, while most dependent variables are expressed in terms indirect maintenance effort based on bug reports.

## 3. Mapping questions and Data extraction form

### 3.1. Mapping questions

The five mapping questions (MQs) addressed in this paper are summarized as the following:

- Which types of OSS projects were used in selected studies (MQ1)? To identify the OSS project types used to build the datasets used in the O-MEE selected studies.
- Which estimation techniques, accuracy criteria and validation methods were used in O-MEE selected studies (MQ2)? To examine the estimation techniques and provide a discussion with respect to accuracy criteria and validation methods used in O-MEE studies.
- Which relationships between dependent and independent variables were most investigated in the selected studies (MQ3)? To identify the relationships between the independent variable and the dependent variables used in the O-MEE selected studies.
- What tools were most frequently used to generate estimates (MQ4)? To identify the estimation tools used in the selected studies to support practitioners with O-MEE tools.

- What were the research validity approaches used in the selected studies to address their limitations (MQ5)? To identify the threats to validity to assess the relevance of the results of the O-MEE studies.

## 3.2. Data Extraction form

A data extraction form (see Table 1) was established to collect relevant information per selected study. This data is useful and helps in answering the MQs of this study. This task was performed by two researchers and checked by another researcher in case of disagreement, by reading full text.

**Table 1**
Data extraction form

| MQs | Extracted data |
|---|---|
| **MQ1** | The OSS software types: 1) Open Source Software (OSS): Studies that discuss MEE techniques in OSS only.2) Other Types of Software (OTS): Studies that discuss MEE for both OSS and proprietary software. |
| **MQ2** | The estimation techniques, the accuracy criteria, the validation methods, and their relationships. |
| **MQ3** | The relationship most used between dependent and independent variables were identified |
| **MQ4** | The tools used to perform estimations includes those used in [6]: WEKA, RapidMiner, SPSS, proprietary, etc. |
| **MQ5** | The research validity approaches includes the threats to validity proposed in [7]: Internal, External, Conclusion, and Construct validity. |

The collected data for each study with respect to the extraction form was grouped in an MS Excel sheet. However, this data is not presented in this paper due to the conference number of pages limit; it can be provided upon request.

# 4. Findings related to the MQs

## 4.1. OSS project types (MQ1)

The analysis identified two software types: some studies focused on MEE for OSS only, and others focused on MEE for both OSS and other types of software (OTS). Figure 1 presents the distribution of the software types over the years: before 2008, the interest was directed toward MEE for OSS; the interest in MEE for OSS and OTS is present in 2008, 2010, 2014, 2016 and 2017 when some researchers investigated to which extent the background and knowledge acquired from MEE research can be useful for OSS. Furthermore, since we are interested in OSS software, Eclipse, Mozilla, Apache, Gnome, Open office and Linux kernel were the most frequently used projects (more than three times).

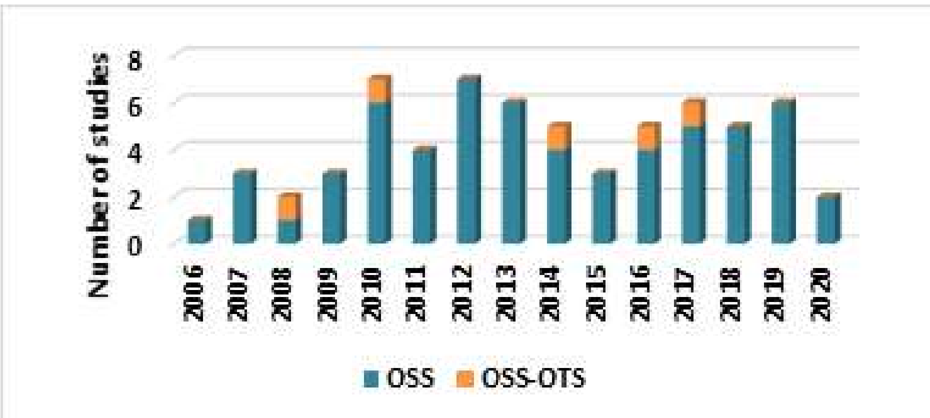


**Figure 1:** Distribution of software types focus per publication year

Besides, variety of data sources were used for the purpose of datasets. For instance: for Eclipse, Firefox, Mozilla, Apache and Linux kernel projects, the researchers used the code source [8] and the bug reports [9], while for Open office and Gnome they used only bug reports [10].

## 4.2. Estimation techniques, accuracy criteria and validation methods (MQ2)

### 4.2.1. Estimation techniques

Two categories of estimation techniques were identified: machine learning (ML) as well as Statistical (ST).

For **ML techniques**: Bayesian Networks (BN) was the most investigated techniques with different variants such as Naïve Bayes and Multinomial Naïve Bayes. Decision Tree (DT) techniques were investigated by means of J48, M5P, C4.5, Chi-squared Automatic Interaction Detector, Random Tree and an alternating decision tree. Support vector Machine (SVM) was also investigated. Instance Based Reasoning (IBR) was investigated with one variant technique, which is k Nearest Neighbor. Apriori Algorithm, Classification Rules, Zero Rule and One Rule were the Rule System (RS) investigated techniques. Ensemble Techniques (ET), RS, and Artificial Neural Networks (ANN) were less used techniques with six studies each. Moreover, AdaBoost, bagging, voting and Random Forest were used for ET, ANN techniques were used by means of Multilayer Perceptron and Feed-Forward Neural Networks with Multinomial Log-Linear Models. Hybrid techniques (HT) were the least investigated in two studies, while Deep Neural Network (DNN) in only a single study.

For **ST techniques**: Regression Analysis (RA) techniques were the most investigated with different variants such as Linear Regression and Logistic Regression. Statistical Models (SM) were investigated using Markov model and Rayleigh SRM. Stochastic Models (SA) were investigated using Weibull distribution and Multivariate Bernoulli distribution.

### 4.2.2. Accuracy criteria:

For any empirical O-MEE study, the estimated values differ from the actual values. To determine how accurate the estimation technique, Figure 2 presents the most frequently used accuracy criteria in O-MEE studies (more than three times). It should be noted that some studies may use more than one accuracy criteria, thus they are counted as many times as they were used. As shown, Recall and Precision are the dominant accuracy criteria, then $R^2$ and F-measure, followed by Accuracy and PRED (25).

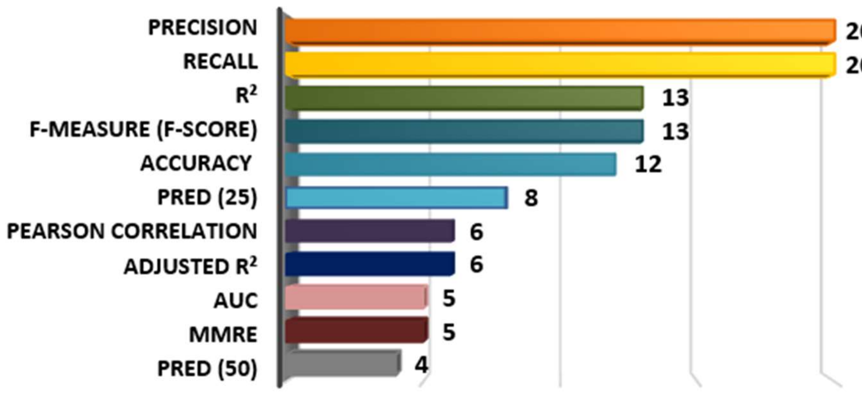


**Figure 2**: Most used accuracy criteria per selected studies

## 4.2.3. Cross validation methods:

Once an O-MEE technique is evaluated, its accuracy needs to be validated. In the selected studies, only 40% (26 out of 65) used different validation methods - see Figure 3.

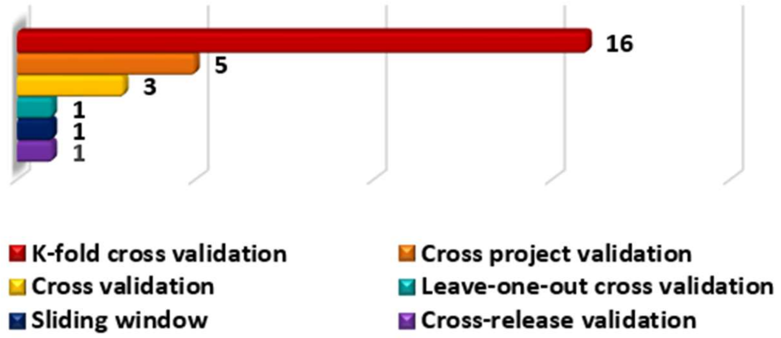


**Figure 3**: Validation methods used in the selected studies

K-fold cross validation (KFCV) was the most used where a known single parameter k refers to the number of groups that a data sample is to be split into. The k-1 groups are used for training and the 1 group left is used for testing; this process is performed k times. For instance, 10FCV was used in [11], and 3FCV in [10]. Cross project validation (CPV) was used in [9], Cross validation (CV) was used in [3]. The least used validation methods are: Leave-one-out cross validation (LOOCV), which is a variation of 1FCV. Cross-release validation (CRV), when "a training dataset was built from a past release of a project, and a test dataset was built from the following release" as reported in [12]. Sliding window (SW) is based on splitting the change-history of a system into a specific time period [13].

## 4.2.4. Relationship estimation techniques-accuracy criteria and cross validation methods

Table 2 summarized the most used accuracy criteria and validation methods grouped by the 12 estimation techniques identified previously. Regarding the accuracy criteria: F-measure was the most used with 11 out of 12 techniques: RA, BN, DT, SVM, IBR, ET, RS, SM, ANN, HT and DNN, followed by Recall, Precision and Accuracy were used together with RA, BN, DT, SVM, IBR, HT and DNN. It was also remarked that Recall, Precision, and F-measure were used together with RA, BN, DT, SVM, IBR, ET, HT and DNN. With respect to the validation methods used: KFCV was the most used with ten out of 12 techniques: RA, BN, DT, SA, SVM, IBR, ET, RS and ANN, followed by CPV with BN, DT, SVM, IBR, RS, ANN and DNN, then CV with RA, BN, DT, SA, ET and RS.

**Table 2**

Estimation techniques, accuracy criteria and validation methods

| Technique | Popular accuracy criteria | Validation methods |
|---|---|---|
| RA | Precision, Recall, $R^2$, RMSE Adjusted $R^2$, PRED (25), MMRE, PRED (50), MdMRE, Spearman correlation, Pearson correlation, Kappa, F-measure, Accuracy, AUC | KFCV, CPV, CV, CRV, SW, LOOCV |
| BN | Precision, Recall, Accuracy, F-measure, AUC, Kappa | KFCV, CPV, CV, SW |
| DT | Precision, Recall, Accuracy, F-measure, AUC, Pearson correlation, MAE, Kappa | KFCV, CPV, CV, CRV, SW |
| SA | Precision, Recall, $R^2$ | KFCV, CV |
| SVM | Precision, Recall, Accuracy, F-measure, | KFCV, CPV |
| IBR | Recall, Precision, Accuracy, F-measure, PRED (25) | KFCV, CPV |

| Technique | Popular accuracy criteria | Validation methods |
|---|---|---|
| ET | Precision, Recall, F-measure | KFCV, CV, CRV |
| RS | Accuracy, F-measure, Kappa | KFCV, CPV, CV |
| SM | F-measure | -- |
| ANN | Accuracy, F-measure, MRE, MAE, Pearson correlation | KFCV, CPV, SW |
| HT | Accuracy, Precision, Recall, F-measure, AUC, AAR, PRED (25), PRED (50), Feedback | SW |
| DNN | Accuracy, Precision, Recall, F-measure and MCC | CPV |

## 4.3. Dependent and independent variables relationships (MQ3)

Before answering this question, a summary of independent variables and the dependent variables identified in [4] is provided first. For independent variables, the seven metrics suites identified, from the most to the least used, are: bug reports attributes, Size metrics, Chidamber and Kemerer metrics, Li and Henry metrics, McCabe metrics, Lorenz and Kidd metrics, and Henry and Kafura metrics. Furthermore, two categories were identified in [4] for dependent variables: Indirect maintenance effort (IME) where researchers have expressed the effort in term of bug fixing time, bug priority and severity, distribution and number of bug, etc. Direct maintenance effort (DME) data was collected by surveying OSS administrators and developers involved in the projects, in order to gather the actual effort in man – days.

**Table 3**
O-MEE independent-dependent variables

| Category | Sub-categories | Metrics suites |
|---|---|---|
| **IME** | Bug fixing/ resolution time prediction | Bug reports attributes, Size metrics |
| | Bug severity prediction | bug reports attributes |
| | Bug priority prediction | Bug reports attributes |
| | Bug prediction | Size metrics, Chidamber and Kemerer, Li and Henry, McCabe, bug reports attributes |
| | Code changes or churn estimation/ prediction | Size metrics, Chidamber and Kemerer |
| | Maintenance changes estimation | Size metrics |
| **DME** | Man – days | Bug reports attributes |

With respect to **independent–dependent variables** relationship, as it can be noticed from Table 3.

For **IME**, Bug reports attributes were widely used with bug fixing/ resolution time prediction in 23 studies, followed by Bug severity prediction in seven studies, Bug priority prediction in four studies and bug prediction in two studies. Size metrics were also popular as they were used for bug prediction in five studies, code changes or churn estimation/ prediction in four studies, bug fixing/ resolution time prediction in three studies and maintenance changes estimation in two studies. Chidamber and Kemerer were used for bug prediction in three studies then code changes or churn estimation/ prediction in two studies. Li and Henry, and McCabe, were used with bug prediction in two studies.

For **DME** category, Bug report attributes were used with two studies out of four, and due to the limited number of studies within this category, no conclusion can be drawn.

## 4.4. Tools used to generate estimates (MQ4)

Six automated software tools were used in 28 out of 65 selected studies to generate estimates (see Figure 4), while some studies used their proprietary tools. In fact, 46% used WEKA tool, 21% used the R software system, 14% used MATLAB, 14% used their own proposed tools, 7% used RapidMiner and 7% used SPSS. The least used tool is Statistica (in one study).

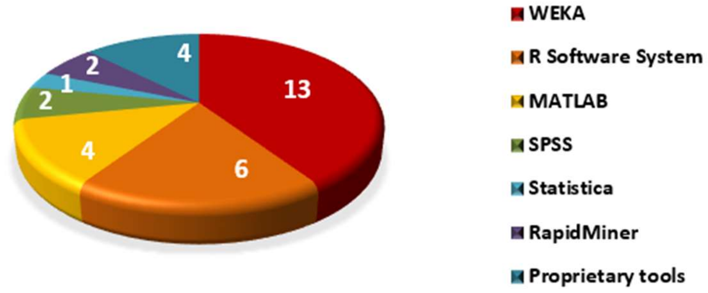


**Figure 4**: Tools used to generate estimated in selected studies

## 4.5. Research validity approaches (MQ5)

Of the 65 O-MEE studies: 42 reported the threats to validity of their research results, 23 lacked of any analysis of validity threats to their empirical results. The main types of validity threats proposed in [7] were extracted and synthesized in Figure 5 with the corresponding number of studies. As shown the most tackled threats to validity are: generalization of the results outside the scope of study (external) with 86%, the outcome affected by the change done (internal) with 81%, the relation between the theory behind the experiment and the observation (construct) with 60%, and conclusion with 12%. Some studies classified their threats to validity based on the study terminology grouped in other category: for instance, the set of experimental projects, field selection, OSS, etc.

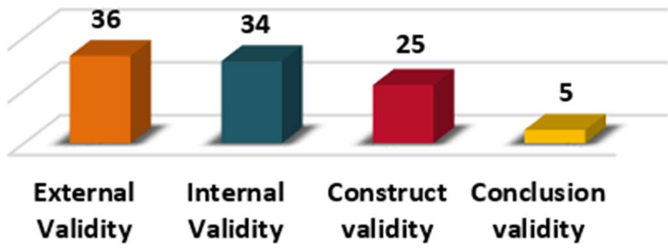


**Figure 5**: Research validity approaches used in the selected studies

# 5. Discussion and trends

This section presents a discussion of this SMS main findings about O-MEE and identifies a set of implications for researchers.

**OSS projects types (MQ1):** While several MEE models were developed for OSS or industrial software, only a few models were developed for both types (only five models) (Figure 1). It would be of great interest to investigate adaptation of OSS models for OTS and reciprocally: this may save a lot of effort and resources for the OSS community as well as for industrial societies interested in OSS. Although, many researchers have empirically evaluated their proposed techniques with the main objective to evaluate their performance, devising new O-MEE models is required as well as comparative studies in order to identify the best models for reliable estimations. It should be noted that some OSS projects are very large[1] and produce different data sources related to their development and maintenance such as Bugzilla[2], JIRA[3], etc. Moreover, with the diversity of the data sources, the format of the data

[1] For instance, according to the latest statistics in 2019, Eclipse is composed of 68.1 million LOC.
[2] https://www.bugzilla.org
[3] https://www.atlassian.com/software/jira

also differs among them varying from structured to unstructured ones, which must be refined to be used in empirical studies. Therefore, data preprocessing of such data sources by proposing a standardized format to support data collection from these sources is required to encourage comparability of the results.

**Estimation techniques, accuracy criteria and validation methods (MQ2):** a panoply of estimation techniques has been used in the O-MEE selected studies: ST techniques with the most used RA and ML ones where the most used in BN. In general, the use of techniques has evolved over the years. ML techniques have gained interest of researchers since they provide accurate models [14], while ST techniques are more acknowledged and simpler to use [15]. Many characteristics of the ML techniques have been reported by researchers in the selected studies. For instance: DT and RS provided explainable and simple models [16]**.** BN, IBR and ANN provided flexible models [17], ANN led to precise results and had the ability of learning [18]. However, IBR could have some weaknesses when employed to cost estimation [19], DT is considered less accurate than the other ML techniques [16], ANN depends on many parameters such the size of the training set [18], on the architecture chosen [20], and ANN does not perform very well in identifying linear relations [21]. Therefore, no single technique is performing well in all circumstances. To deal with this, a few researchers investigated the use of ET which combines various single techniques [10] in O-MEE models. Such scarcity of ET brings an opportunity to explore them for relevant models since they have proved to produce more accurate estimates than the single techniques [22]. Also HT were recommended in the literature, considering their performance for effort estimation over single techniques [13]. Deep learning techniques were less investigated in the selected studies (one single study); therefore, researchers are encouraged to devise more studies in this context.

Widespread accuracy criteria have been used in the selected studies to evaluate the performance of the O-MEE models (Figure 2). The accuracy criteria were not identified for each selected studies in the SLR [5], but the accuracy values of some common accuracy criteria among selected studies were used to perform accuracy analysis. In this SMS, Recall, Precision, $R^2$ and F-measure were the most used accuracy criteria, which means that the classification problems were the most addressed. MMRE and PRED accuracy criteria have been less used; therefore, prediction models are needed in order to estimate the effort for maintenance operations in OSS.

A variety of validation methods have been used in the selected studies such as those based on KFCV method (K = 3 or K= 10) (Figure 3). SLR [5] also reported that KFCV (K > 1) was the dominant validation method among the selected studies. The "K-fold cross-validation considers the variation among data points, and provides less biased predictive accuracy results. Leave-one-out cross-validation is a specific variation of k-fold cross-validation, typically used when building effort estimation with small datasets" [23]. Moreover, new validation methods such as CRV, CPV and SW were less frequently used. CPV based models "provide similar (or even better) results than those of intra-projects"[24], CRV performs an evaluation using more practical setting [12], and SW is adopted when using process metrics as predictors but it's not recommended for product metrics [13].Therefore, researchers are encouraged to investigate those new validations methods.

**Dependent and independent variables (MQ3):** With regards to relationships between independent and dependent variables, the bug report attributes were commonly used with bug fixing/ resolution time prediction the popular combination in the selected studies- as well as with bug severity and bug priority prediction, followed by size metrics mainly used with bug prediction (Table 3). However, code source metrics such as McCabe and Chidamber and Kemerer were rarely used. Additionally, some metrics have not been investigated with all the dependent variables such the case of Chidamber and Kemerer and bug fixing/ resolution time prediction. Therefore, new O-MEE models could be based on code source data sources as well as different types of metrics in order to obtain more relevant results by comparing different type of models. Moreover, researchers are encouraged to tackle these gaps in further validation studies.

**Tools used to generate estimates (MQ4):** a variety of automated software tools were used in the O-MEE (Figure 4). Only four of them can be applied to ML techniques: WEKA, SPSS, MATLAB and RapidMiner, while the other ones are specialized in statistical techniques. All these tools have their own peculiarities in terms of implementation and each has its own merits. Therefore, the choice of the software tool is very dependent on the tool features and the purpose of the experiments.

**Research validity approaches (MQ5):** more than half of the selected studies lacked of any analysis of validity threats to their empirical results (Figure 5), and almost 36 out of the 42 suffers from external validity and thus suffers from generalizability of the results. Moreover, the conclusion validity is vital, and all studies were expected to discuss it. However, it is not the case for this set of 65 selected studies. Therefore, researchers are encouraged to address the threats or the limitations of their studies and more studies should be considered in order to identify what kind of quantifiable measures have been used in the selected studies to minimize those threats to be able to compare the results of the studies.

## 6. Conclusion

This study purpose was to provide a state of the art and research needs about O-MEE topic by answering a set of five MQs based on the set of 65 studies selected in [4]. The main findings of this study are summarized as the follows. The most used OSS projects are bug reports from Eclipse, Mozilla, Firefox Apache, and Gnome. The most used techniques are BN and RA. The frequent used accuracy criteria are: Recall, Precision, $R^2$ and F-measure. KFCV was the most adopted validation method. The most used combinations of independent and dependent variables are bug report attributes with bug fixing/ resolution time prediction followed by bug severity. The most used tools were WEKA, R software and MATLAB. The most current threats to validity were external, then internal and construct.

The findings of this study were based on the set of 65 O-MEE selected studies and the data extracted using the established form in section 3. To mitigate the threat of extracting good data, two authors have performed this task keeping in mind the MQs purposes and without altering any data. Another third researcher reviewed the results and all disagreement were discussed until the authors reached a consensus. Furthermore, to draw the conclusions of this study, each study was investigated using data extracted using the established extraction form. These findings were next checked by a fourth researcher for results consistency and reliability. To summarize, the findings of this SMS should be beneficial to researchers conducting O-MEE studies as well as practitioners interested in using these models. Section 5 also provides recommendations to carry out future research in the O-MEE topic.

## References


[1] P. Bourque and R. E. Fairley, IEEE Computer Society, in : Guide to the software engineering body of knowledge, 2014.

[2] A. Abran and H. Nguyenkim, Measurement of the maintenance process from a demand-based perspective, J. Softw. Maint: Res. Pract. 5, no. 2, (1993) 63–90.

[3] A. Capiluppi and D. Izquierdo-Cortázar, Effort estimation of FLOSS projects: a study of the Linux kernel, Empirical Software Engineering 18 (2013) 60–88.

[4] C. Miloudi, L. Cheikhi, A. Abran, and A. Idri, Open source software maintenance effort estimation: a systematic mapping study, Journal of Engineering Science and Technology (2022), To appear.

[5] H. Wu, L. Shi, C. Chen, Q. Wang, and B. Boehm, Maintenance Effort Estimation for Open Source Software: A Systematic Literature Review, in: Proceedings of the IEEE Conference on Software Maintenance and Evolution, 2016, pp. 32–43.

[6] S. Elmidaoui, L. Cheikhi, A. Idri, and A. Abran, Empirical Studies on Software Product Maintainability Prediction: A Systematic Mapping and Review, E-Informatica Softw. Eng. J. 13 (2019), p. 62.

[7] C. Wohlin, Experimentation in software engineering. New York: Springer, 2012.

[8] P. Bhattacharya and I. Neamtiu, Assessing programming language impact on development and maintenance: a study on c and c++, in: Proceedings of the 33rd. International Conference on Software Engineering, 2011, pp. 171–180.

[9] M. Sharma and A. Tondon, Developing Prediction Models to Assist Software Developers and Support Managers, Computational Science and Its Applications 10408 (2017) 548–560.

[10] S. Guo, R. Chen, M. Wei, H. Li, and Y. Liu, Ensemble Data Reduction Techniques and Multi-RSMOTE via Fuzzy Integral for Bug Report Classification, IEEE Access 6 (2018).

[11] K. K. Sabor, M. Hamdaqa, and A. Hamou-Lhadj, Automatic prediction of the severity of bugs using stack traces and categorical features, Information and Software Technology 123 (2020).

[12] Y. Kamei, S. Matsumoto, A. Monden, K. Matsumoto, B. Adams, and A. E. Hassan, Revisiting common bug prediction findings using effort-aware models, in: Proceedings of the International Conference on Software Maintenance, 2010, pp. 1–10.

[13] D. Di Nucci, F. Palomba, G. De Rosa, G. Bavota, R. Oliveto, and A. De Lucia, A Developer Centered Bug Prediction Model, IEEE Transactions on Software Engineering 44, no. 1 (2018) 5–24.

[14] M. Sharma, M. Kumari, R. K. Singh, and V. B. Singh, Multiattribute Based Machine Learning Models for Severity Prediction in Cross Project Context, in Computational Science and Its Applications 8583 (2014).

[15] M. R. M. Talabis, R. McPherson, I. Miyamoto, J. L. Martin, and D. Kaye, Analytics Defined, Information Security Analytics (2015) 1–12.

[16] H. Valdivia Garcia and E. Shihab, Characterizing and predicting blocking bugs in open source projects, in: Proceedings of the 11th. Working Conference on Mining Software Repositories 2014, pp. 72–81.

[17] S. Bibi, A. Ampatzoglou, and I. Stamelos, A Bayesian Belief Network for Modeling Open Source Software Maintenance Productivity, Open Source Systems: Integrating Communities 472 (2016) 32–44.

[18] A. Kaur and D. S. Singh, Comparison of Maintenance Activity for Effort Estimation in Open Source Software Projects, International Journal of Advanced Research in Computer Science (2017), p. 5.

[19] C. Weiss, R. Premraj, T. Zimmermann, and A. Zeller, How Long Will It Take to Fix This Bug?, in: Proceedings of the Fourth International Workshop on Mining Software Repositories 2007, pp. 1–1.

[20] L. Wang, X. Hu, Z. Ning, and W. Ke, Predicting Object-Oriented Software Maintainability Using Projection Pursuit Regression, in: Proceedings of the First International Conference on Information Science and Engineering, 2009.

[21] S. Karus and M. Dumas, Code churn estimation using organisational and code metrics: An experimental comparison, Information and Software Technology 54, no. 2 (2012) 203–211.

[22] M. Azzeh, A. B. Nassif, and L. L. Minku, An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation, Journal of Systems and Software 103 (2015) 36–52.

[23] A. Hira and B. Boehm, COSMIC Function Points Evaluation for Software Maintenance, in: Proceedings of the 11th. Innovations in Software Engineering Conference, 2018, pp. 1–11.

[24] H. Wang and H. Kagdi, A Conceptual Replication Study on Bugs that Get Fixed in Open Source Software, in: Proceedings of the International Conference on Software Maintenance and Evolution, Sep. 2018, pp. 299–310.