Does class size matter? An in-depth assessment of the effect of class size in software defect prediction

Amjed Tahir[1] · Kwabena E. Bennin[2] · Xun Xiao[1] · Stephen G. MacDonell[3,4]

Abstract

In the past 20 years, defect prediction studies have generally acknowledged the effect of class size on software prediction performance. To quantify the relationship between object-oriented (OO) metrics and defects, modelling has to take into account the direct, and potentially indirect, effects of class size on defects. However, some studies have shown that size cannot be simply controlled or ignored, when building prediction models. As such, there remains a question whether, and when, to control for class size. This study provides a new in-depth examination of the impact of class size on the relationship between OO metrics and software defects or defect-proneness. We assess the impact of class size on the number of defects and defect-proneness in software systems by employing a regression-based *mediation* (with bootstrapping) and *moderation* analysis to investigate the direct and indirect effect of class size in *count* and *binary* defect prediction. Our results show that the size effect is not always significant for all metrics. Of the seven OO metrics we investigated, size consistently has significant mediation impact only on the relationship between Coupling Between Objects (CBO) and defects/defect-proneness, and a potential moderation impact on the relationship between Fan-out and defects/defect-proneness. Other metrics show mixed results, in that they are significant for some systems but not for others. Based on our results we make three recommendations. One, we encourage researchers and practitioners to examine the impact of class size for the specific data they have in hand and through the use of the proposed statistical mediation/moderation procedures. Two, we encourage empirical studies to investigate the indirect effect of possible additional variables in their models when relevant. Three, the statistical procedures adopted in this study could be used in other empirical software engineering research to investigate the influence of potential mediators/moderators.

Keywords Defect prediction · Class size · Metrics · Software quality

Extended author information available on the last page of the article.

1 Introduction

Software defect prediction aims to identify potentially defective software modules which in turn aids in effective allocation and prioritisation of scarce testing resources. Constructing defect prediction models is dependent on the availability and use of historic software project data. Metrics extracted from software components have been used for predicting either the number of defects or defect-proneness in numerous software defect prediction studies (D'Ambros et al. 2012a; Jureczko and Madeyski 2010). The performance and stability of prediction models have been extensively assessed and discussed in prior studies (Ghotra et al. 2015). Among the metrics often considered is class size, an attribute that is generally thought to strongly influence the relationship between all other metric features and defects. Conventionally, it is assumed that larger classes are more likely to be 'defective' or be 'more defective' than smaller classes (Mende and Koschke 2009).

The jury is still out on the true impact of class size on the relationship between OO metrics and the defect-proneness of software components. Empirical evaluation of the indirect effect of class size on the relationship between software characteristics and defects has been undertaken by few researchers. Prior studies (El Emam et al. 2001; Zhou et al. 2014; Gil and Lalouche 2017) have evaluated the effect of class size on binary models in predicting defect-proneness (i.e. if a file or module is buggy or not). Notwithstanding, the conclusions of these prior studies do not apply to other defect prediction models, specifically count models built to predict the number of defects (Fenton and Neil 1999; Zhang 2009). Additionally, the Type 1 error control and power of the statistical tests employed by prior studies (El Emam et al. 2001; Zhou et al. 2014) (especially when causal-steps or Sobel tests are used) have been shown to be inferior to other statistical methods (Hayes 2013).

The need for the use of more robust statistical tests in validating the results of empirical analyses conducted in software engineering has been discussed in recent studies (e.g., (Kitchenham et al. 2016)). In the context of defect prediction, there have been recent calls to revisit many of the previous conclusions with the use of larger datasets (Majumder et al. 2020). In the two previous studies (El Emam et al. 2001; Zhou et al. 2014), only a limited number of systems drawn from a single dataset (a single telecommunications framework with 174 classes in El Emam et al. (2001) and 6 open-source systems in Zhou et al. (2014)) have been used. To overcome such limitations, we therefore utilise a much larger set of systems obtained from two different datasets (with a total of 23 systems) in an attempt to provide a more general conclusion.

A two-part process has been followed in the past to examine whether a group of metrics might be useful in predicting defects/defect-proneness (Basili et al. 1996b; Olague et al. 2007; Zhou and Leung 2006). In the first step, which can be referred to as "model specification" (Tantithamthavorn and Hassan 2018), one must define which metrics should be included in the model. Each individual OO metric is used as an independent variable to build a univariate model against the dependent variable (being the number of defects or defect-proneness). The main purpose of this process is to establish the validity of each of the selected metrics in predicting defects.

El Emam et al. (2001) and Zhou et al. (2014) found that, in this first part of the process, and for most OO metrics, class size has a significant indirect (confounding) effect on the validity of the studied OO metrics. The recommendation of both studies is that model specification using individual OO metrics should always control for class size *before* researchers go on to the second step of the process, building multivariate prediction models.

The importance of such empirical hypothesis testing when considering whether to control for the indirect effect of metrics in defect models has also been highlighted in the work of Tantithamthavorn and Hassan (2018). In that research (Tantithamthavorn and Hassan 2018), the authors pointed out that defect models, besides predicting defects, are also critical in carefully examining phenomena and hypotheses of interest in an effort to derive empirical theories. Similarly, in this study we are interested in studying if the OO metrics are indirectly influenced by class size as a variable in the model (i.e., the potential indirect effect of size). Thus, the main motivation of this paper is to provide a more complete understanding of the indirect effect of class size on the OO metrics' defect prediction ability. We reassess the effect of class size on the association between OO metrics and defects in a class using robust statistical methods through detailed *mediation* and *moderation* analyses on 23 projects (including both open source and industrial projects).

Mediation and moderation statistical analysis methods examine the indirect effect of *additional* variables on the relationship between a predictor and response variables in a model. In short, *mediation* analysis tests whether the effect of the predictor (X) on the outcome (Y) operates through a third variable called the mediator (M). *Moderation* analysis tests whether a moderator variable (Z) affects the direction or the strength of the relationship between the predictor (X) and the outcome (Y) variables. In other words, mediation investigates the *conditional effect* of a third variable whereas moderation investigates the *interaction effect* of a third variable in the model.

The main contribution of this study is an in-depth understanding of the true indirect effect (through *mediation* and *moderation* analysis) of class size in defect prediction models across both count and binary models. We also provide an easy-to-follow statistical procedure to assess the *mediation* and *moderation* effects before building prediction models. The paper extends our early work (Tahir et al. 2018) in that it examines not only the effect of size in count models (i.e., the number of defects), but also in binary models (i.e., defect-proneness), and does so on a larger number of systems (23 systems in total). The paper addresses the following two main research questions:

RQ1: does class size have an effect on the relationships between OO metrics and the number of defects?

RQ2: does class size have an effect on the relationships between OO metrics and defect-proneness?

RQs 1 and 2 can be then divided into two sub-questions each:

RQ1.1: does class size have a *mediation* effect on the relationships between OO metrics and the number of defects?

RQ1.2: does class size have a *mediation* effect on the relationships between OO metrics and defect-proneness?

RQ2.1: does class size have a *moderation* effect on the relationships between OO metrics and the number of defects?

RQ2.2: does class size have a *moderation* effect on the relationships between OO metrics and defect-proneness?

We test the following four Null Hypotheses:

H01.1: class size has no *mediation* effect on the relationship between OO metrics and the number of defects.

H01.2: class size has no *mediation* effect on the relationship between OO metrics and defect-proneness.

H02.1: class size has no *moderation* effect on the relationship between OO metrics and the number of defects.

H02.2: class size has no *moderation* effect on the relationship between OO metrics and the defect-proneness.

Our empirical and statistical analyses show that the evidence regarding the *mediation* effect of size is inconsistent across different systems. Only the effect of size on the relationship between CBO and the number of defects or defect-proneness can be considered to be significant across most systems. Size appears to have a stronger *moderation* effect on the relationship between Fan-out and defects. To explain this inconsistency of results that are obtained from different systems, we conducted further analysis to explore the similarity between different systems in our dataset and then compare the similarity outcomes with the mediation and moderation analysis results obtained from these systems. This analysis reveals that data collected from different systems exhibit rather different patterns, and that systems that are highly correlated share the same mediation outcomes, but differ in their moderation outcomes.

We encourage software teams to consider the indirect effect of class size during defect prioritisation activities, by applying the bootstrapping mediation and moderation analysis methods that have been used in this study. If size is found to have a strong effect, then statistical methods to control that effect should be used. The statistical mediation and moderation analysis method employed here could also have wider application in other software engineering areas such as in cost and effort estimation and change prediction.

The remainder of this paper is organised as follows: Section 2 presents related work in defect prediction with a focus on analysing the relationship between OO metrics, size and prediction performance. The research methodology is presented in Section 3. We present the statistical analysis and results in Section 4, followed by a discussion of the practical implications of the results. Potential threats to validity are discussed in Section 5 and we conclude the study in Section 6.

2 Related Work

Defect prediction for software components has been extensively investigated to date. Previous studies have used various metrics extracted from source code or design artifacts to predict defects (Basili et al. 1996a; Zimmermann et al. 2007; Schröter et al. 2006). Other studies have used change metrics (e.g., number of changes, commit history, dependencies, code churn) as predictors of defects (Hassan 78; Ying et al. 2004; Zimmermann et al. 2008).

Gyimothy et al. (2005) found LOC to be a good low-consistency predictor of defects, as it is easier to collect compared to many other metrics. Fenton and Neil (1999), Hall et al. (2011), Shepperd et al. (2014) and D'Ambros et al. (2012b) all provide comprehensive overviews of defect prediction models, their structure and validity. Research in defect prediction has resulted in models for both binary/classification (e.g., Song et al. (2011)) and count (e.g., Osman et al. 2018) prediction.

To improve prediction performance, a range of factors have been studied, among them feature and model selection (Catal and Diri 2009; Hayakawa et al. 2021) and class imbalance (Bennin et al. 2018; Bennin et al. 2018). Additionally, cross-project defect prediction proposed to aid software practitioners who need to test new software projects with no historical data has also been studied (Turhan et al. 2009). Several efforts (Kamei et al. 2013; Kamei et al. 2016; Pascarella et al. 2019) on building prediction models for identifying

defect-inducing changes, known as Just-In-Time defect prediction, have been investigated. The relationships between OO metrics and defect-proneness, in particular, have been investigated by prior studies (El Emam et al. 2001; Zhou et al. 2014). El Emam et al. (2001) empirically investigated the relationships between OO metrics and class defect-proneness. The authors built two logistic regression models for each OO metric considered with LOC (as the size measure) included in one of the models. To measure the magnitudes of the associations between the OO metric and defect-proneness with and without controlling for size, two odds ratios were computed for the two models. A comparison between the two odds ratios with a large percentage difference suggested that class size had a confounding effect. The results of the study indicated that the associations between most of the metrics investigated (namely, Weighted Methods per Class (WMC), Response for Class (RFC) and Coupling Between Objects (CBO)) and defect-proneness were significant before controlling for size but were not so after controlling for size. The authors thus recommended that defect prediction models should always control for size.

The approach adopted by El Emam et al. (2001) mirrors the Baron and Kenny Causal-steps (joint significance) test (Hayes 2009). While simple and widely used, the causal-steps test is known to have less power than other available mediation analysis approaches (Fritz and Mackinnon 2007; MacKinnon et al. 2002). Additionally, this approach is not based on a quantification of the intervening indirect effect, but examines a series of links in a causal chain between variables (e.g., $X \rightarrow M \rightarrow Y$). A similar approach was used by Gil and Lalouche (2017), which also followed a causal-steps approach to determine the effect of size on a number of OO metrics. However, their study found that, when controlling for size, metrics may lose their predictive power. The findings of El Emam et al. have been argued by other works. Evanco (2003) argued that introducing class size as an additional independent variable in a defect prediction model can result in misspecified models that lack internal consistency. Another study reported that complexity metrics are found to have a strong correlation with LOC, and therefore the more complexity metrics provide no further information that could not be measured simply with LOC (Herraiz and Hassan 2010).

More recently, Zhou et al. (2014) conducted a study on the effect of class size on the association between OO metrics and defect-proneness. In contrast to the study by El Emam et al. (2001), the authors employed the Sobel test (Sobel 1982), which examines the significance of the mediation effect by calculating the standard error of XY, evaluating the null hypothesis that the true effect of a mediating variable is zero. Their results showed that size has a mediation effect on the relationship between OO metrics and defect-proneness. Positively, the Sobel test is known to perform better (in terms of power) than the Causal-steps approach (Fritz and Mackinnon 2007); however, there are a number of issues with this test, particularly its weakness due to the assumption of normality of sampling. In most cases the sampling distribution of $X \rightarrow Y$ tends to be asymmetric, with high skewness and kurtosis (Bollen and Stinet 1990; MacKinnon et al. 2002). In our analysis of one of the two datasets used by Zhou et al. (2014) (originally generated in D'Ambros et al. 2012b), we found that the most metrics data do not follow a normal distribution (see Section 3.4). In addition, the Sobel test has also been shown to score low in power and Type I error control (MacKinnon et al. 2002). Note that all previous studies have investigated the size effect from only one perspective, the indirect effect of size; that is, where size is a third variable and that the relationship between the metrics and defect-proneness is transmitted through this variable. As such, size is in this case a *mediator* variable, and such a variable is analysed using a mediation analysis technique.

The known issues with both techniques (i.e., the Causal-steps and Sobel tests), combined with a level of conclusion instability across different studies, led us to seek alternative

approaches to investigate the true indirect effect of size. In this work we assess whether class size has a mediation effect on the relationships between OO metrics and defects (in count models) using robust statistical analysis. In addition, we assess another size effect phenomenon known as the *moderation* effect - we study whether class size moderates (i.e., affects the size or strength of) the relationship between OO metrics and the number of defects. We are not aware of any other study that has investigated the moderation effect of class size in defect prediction models. As such our results should inform the design of future models that may - or may not - employ class size as one of the predictors of defects.

3 Research Setting and Empirical Analysis

3.1 Estimating the Indirect Effect of Variables

This work aims to estimate either mediation or moderation effects of size, assuming that size might have one of the effects on the relationship between OO metrics and defects in a class. Note that the previous studies in this space (e.g., El Emam et al. 2001 and Zhou et al. 2014) have investigated the *mediation* effect of size in defect prediction models. This is what has been referred to as the "confounding effect". While it is important to confirm whether size has a mediation effect in defect prediction models, we also believe that it is important to investigate the potential *moderation* effect of size (that is, the impact of size on the *strength* of the relationship between OO metrics and the number of defects). Understanding the nature of the indirect effect of size is quite important especially when selecting methods to control for such an effect (i.e., methods to control for a mediator could be different than those used to control for moderators).

Given an independent variable X and a dependent variable Y, the effect of X on Y may be transmitted through a third intervening (mediating) variable M. That is, X affects Y because X affects M and M in turns affects Y (i.e., $X \rightarrow Y$ is the result of the indirect relationship $X \rightarrow M \rightarrow Y$). Figure 1 depicts a path diagram showing a direct relationship between an independent variable (IV, an OO metric) and a dependent variable (DV, number of defects or defect-proneness). Figure 2 depicts the possible indirect relationship between an OO metric and the number of defects through a mediator: class size. If a mediator is controlled (i.e., held constant) and that mediator was the reason for the relationship between the independent and dependent variables (i.e., there is complete mediation), this indicates that there is no direct relationship between the independent and dependent variables.

On the other hand, a *moderator* is a variable that impacts the power of the relationship between the independent and dependent variables. It can affect the sign or the strength/magnitude of the relationship between those two variables. Figure 3 shows the potential moderation effect of size on the relationship between an OO metric and defects. The relationship between X (independent variable) and Y (dependent variable) is said to be moderated if its size or direction depends on a third variable M (Hayes 2013). In other words, a moderator is a third variable that affects the zero-ordered correlation between the
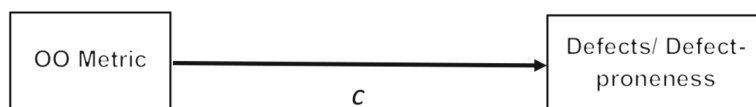

Fig. 1 Path diagram illustrating a simple model for the relationship between OO metrics and defects
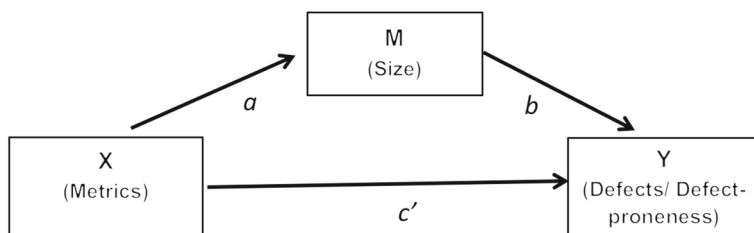
Fig. 2 Path diagram illustrating a simple *mediation* effect of size on the relationships between OO metrics and defects

independent and dependent variables. Unlike mediation, which affects the extent that it accounts for the relationship between the independent and dependent variables, a moderation variable does not necessarily explain the relationship between variables but rather it explains *the strength of* this relationship (Baron and Kenny 1986). A moderator effect is indicated by the interaction of X and M in explaining the outcome Y.

3.2 Methods to Detect Mediators and Moderators

A number of candidate approaches can be employed to determine the true effect of a mediating variable on the relationship between two other variables, and MacKinnon et al. (2002)
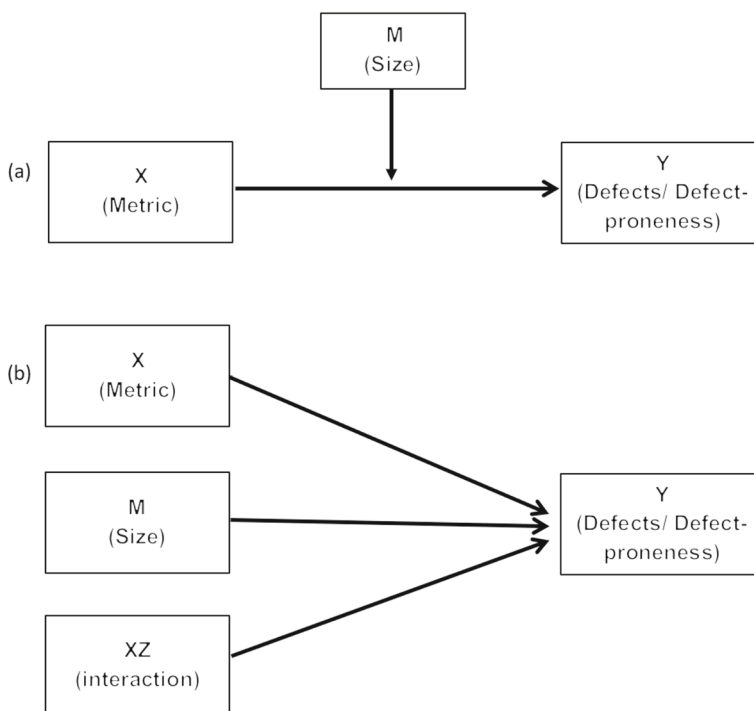

Fig. 3 A path diagram (**a**) and a statistical model (**b**) illustrating a simple *moderation* effect of size on the relationships between OO metrics and defects/defect-proneness

provide a detailed comparison of several well-known methods. Of those methods, there is a strong and quite recent recommendation in the literature to use bootstrapping-based mediation tests over other tests such as the causal-steps and Sobel-based tests, since it tends to have better power and Type I error control than other approaches (MacKinnon et al. 2002; Hayes 2009). Bootstrapping-based methods for estimating indirect effects have been discussed in the literature for some time (e.g., Bollen and Stinet 1990) and the methods have received more attention in recent years (MacKinnon et al. 2004; Preacher and Hayes 2008; Hayes 2009).

For mediation analysis, we adopted a bootstrapping-based method to test for different effects multiple times using a regression-based approach. However, the approach used to estimate mediation effects for count models (i.e., where the dependent variable is a count, in our case, that is the number of defects) and binary models (the dependent variable is defect-proneness) is slightly different - this is discussed in more detail in Section 3.3.1.

In moderation analysis, we study whether the effect of the increase/decrease in the independent variable X does not depend on a moderator M. The moderator M interacts with X in predicting Y if the regression weight of Y on X varies as a function of M. We measure the change in the R^2 value due to the interaction of the moderator variable $X \cdot M$. In case that such an interaction is significant, we then used simple slope analysis to probe the interaction of the moderator. Simple slope can visually portray how another variable moderates the association between the independent and dependent variable. This is done when the dependent variable is either count (number of defects) or dichotomous (defect-proneness).

3.3 Conducting Mediation and Moderation Analysis

3.3.1 Mediation Analysis

In conducting our mediation analysis, we produced a series of regression models. As explained in Section 3.1, the mediation effect is estimated through three levels of effect: the *direct*, *indirect* and *total* effects (illustrated in Figs 1 and 2). To investigate the *indirect effect* of size on the number of defects, we first build a linear model where the dependent variable (the number of defects) is explained directly by a single OO metric (step 1). This is called the *direct effect* (see Fig. 1). Then we build another linear model where the mediator (transformed LOC) is explained by the independent variable (the same single OO metric) (step 2). Using Poisson regression (in case where the outcomes is a non-negative integer value), we then built a new model to explain variance in the dependent variable (number of defects) using both the independent variable and the mediator (transformed LOC) (step 3) - this is the path *ab* in Fig. 2. The *indirect effect* is then estimated. When the outcome variable is defect-proneness (i.e., the dependent variable is dichotomous), we use logistic regression models instead in steps 1 and 3.

The reasons for those choices are rather straightforward. Defect-proneness data follows a binary-value Bernoulli distribution, and logistic regression is the standard approach for modelling binary-value random variables with possible covariates. For the number of defects, our normality analysis (discussed in Section 3.4) shows that the number of defects obtained from different datasets used in this study tend not to follow a Gaussian distribution, but in fact it is closer to a Poisson distribution. Given a sufficiently large class size with M LOC with a small probability λ of defect occurrence in each line, the Poisson limit theorem guarantees that the number of defects will follow a Poisson distribution with mean $\mu = M\lambda$ approximately. Thus, we decided to use Poisson regression as it is widely used in

modelling random counts with covariates. The detailed math formulations of these models are presented next.

To model the *total effect*, and in case of a Poisson regression model where the outcome and number of defects Y, follow a Poisson distribution with mean μ, we link the log mean with each individual OO metric X as:

$$\log(\mu) = \gamma_0 + \gamma_1 X \tag{1}$$

On the other hand, and in the case of logistic regression where the outcome, defect-proneness Z, follows a Bernoulli distribution with a probability p, to model the *total effect* we link the log odds with each individual OO metric X as:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X \tag{2}$$

To model the *direct effect*, we use a linear regression model with the mediator $\log(M)$, i.e., log class size as the outcome and each individual OO metric X as the covariate

$$\log(M) = \theta_0 + \theta_1 X \tag{3}$$

To model the *indirect effect* of size, we include two covariates in the model, with an individual OO metric as the independent variable and class size M as the mediator. However, Poisson regression imposes a log-linear relationship between the mean of Y and the covariates, which can be further formulated as:

$$\text{mean}[Y] = \mu = \exp\left(\gamma_0 + \gamma_1 X + \gamma_2 M\right) \tag{4}$$

Equation (4) implies that the average number of defects increases with increasing class size M at an exponential rate. This contradicts the Poisson limit theorem which implies that there is a linear relationship between class size M and the average number of defects μ. In fact, the class size data obtained from the datasets used in our study were far from the Gaussian distribution needed for linear models (see details in Section 3.4), while the log class size fits better into a bell-shaped density. Therefore, rather than using class size directly, we use the log class size $\log(M)$ as the mediator in both Poisson and logistic regression models, as follows:

$$\log(\mu) = \gamma_0' + \gamma_1' X + \gamma_2' \log(M) \tag{5}$$

$$\log\left(\frac{p}{1-p}\right) = \beta_0' + \beta_1' X + \beta_2' \log(M) \tag{6}$$

Note that we estimate the indirect effect (at step 3) only if the results of the first two steps are found to be significant (i.e., the *total* and *direct* effects are both significant). If the results of all three steps are found to be significant, then we establish that data are consistent with the hypothesis that variable M mediates the $X[\rightarrow]Y$ relationship. On the other hand, if the results of either of the first two steps were not found to be significant, then we conclude that the mediation effect of M is unlikely. For defect-proneness (i.e., when Y is dichotomous), we estimate the indirect effect of ab only when the direct effect c' is significant. Similarly, if both are significant, we then establish that M mediates the $X[\rightarrow]Y$ relationship.

The mediation analysis was carried out using the `mediation` package for R.[1] This package provides a mechanism to conduct mediation analysis using various standard models such as linear or generalized linear models and returns the estimates of the average causal mediation effects. The `mediate` function in `Mediation` calculates point estimates, CIs

---

[1]https://cran.r-project.org/web/packages/mediation/

and the p-values, for the average *direct*, *indirect*, and *total* effects. The algorithm and formal definitions of the models used in `mediation` are explained in Tingley et al. (2014).

In applying bootstrapping mediation analysis, we employ 5000 bootstrap resamples and set the confidence intervals to 95% for the mediation analysis. Tingley et al. (2014) suggested to use a bootstrap with a minimum of 1000 resamples. Other studies recommended 5000 bootstrap resamples for effective resampling, as long as there is enough computation resources (Preacher and Hayes 2008). We first experimented with various bootstrap resamples (i.e., using 10,000 bootstrap) to see if an increase in the resampling would in fact impact the results. We compared the results obtained from running both 5000 and 10,000 bootstraps for 3 of the systems (total of 6 analyses) for each of two datasets we used in this study. We did not observe any changes in the overall results (i.e., no significance changes in p-values or CI levels). Therefore, we decided to conduct all mediation analysis using 5000 bootstrap resamples, following the recommendation in Preacher and Hayes (2008).

3.3.2 Moderation Analysis

Moderation analysis is also estimated via regression modelling by building a regression model to fit the outcome (number of defects or defect-proneness) using the independent variable (an OO metric), the moderator and the *interaction term*, which tests whether the relationship between the OO metric and the outcome changes based on the moderator. The interaction term is computed as the product of the independent variable and the moderator. If the interaction term has $p < .05$, the moderation effect is then declared as significant.

Initially, we investigated those interaction terms at two levels, one for each individual metric (i.e., single predictor) and with all metrics combined (i.e., multivariate predictors). Although we investigated both approaches, the interactions between the individual metrics and the moderators, tested individually, would provide a clearer view of the true effect of the moderator. If we build a model with all interactions put together, and that model has a significantly better fit than the model without interactions, we will likely not know for sure which interaction is independently significant and will have more impact; thus, the decision to use univariate models to investigate the moderation effect of size.

Similar to the mediation analysis explained in Section 3.3.1, we used Poisson and logistic regressions to model the moderation effect of size, using log class size $\log(M)$ as the moderator. In models where the outcome is the number of defects Y and in which the size data follows a Poisson distribution with mean μ, we link the log mean with the OO metric X, the moderator $\log(M)$ and their interaction term $X \cdot \log(M)$ as:

$$\log(\mu) = \gamma_0 + \gamma_1 X + \gamma_2 \log(M) + \gamma_3 X \cdot \log(M) \tag{7}$$

For models where the outcome is defect-proneness Z and size follows a Bernoulli distribution with a probability p, we link the log odds with the OO metric X, the moderator $\log(M)$ and their interaction term $X \cdot \log(M)$ as:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X + \beta_2 \log(M) + \beta_3 X \cdot \log(M) \tag{8}$$

3.4 Datasets

To examine the potential size effect on defect models, we selected two publicly available open source datasets (D'Ambros et al. 2012b (DAMB), and Jureczko and Madeyski 2010 (JURE)). We select these datasets and their constituent systems for the following reasons:

1) they contain systems of large size which have been developed by popular open-source communities such as Eclipse and Apache, 2) the JURE dataset contain data obtained from both open-source and and proprietary software (6 industrial systems) so our data sample can be *slightly* representative, 3) these datasets have been widely used in previous defect prediction studies (e.g., He et al. 2012; Bennin et al. 2018)) – in particular the DAMB dataset was used in the study of Zhou et al. (2014), which offers an opportunity to compare results. We also considered other datasets such as Zimmermann et al. (2008) and the NASA metrics program data, but we found that those datasets do not report the same metrics as those already selected for our study (i.e., Zimmermann et al. 2008) or they are too small (i.e., KC1), therefore they are not suitable for our study. Both datasets are publicly available.

The D'Ambros et al. (2012b) dataset includes metrics data collected from five large open source projects from two well-established ecosystems: Eclipse (JDT Core, PDE UI, Mylyn and Equinox) and Apache (Lucene). For simplicity, we refer to this dataset as DAMB in the rest of this paper. The Jureczko and Madeyski (2010) dataset contains data from open source and proprietary projects (although from relatively small systems). We excluded systems comprising small number of modules (fewer than 30) as we consider them too small for our analysis. We included 18 systems from this dataset in our study. This includes 11 well-known Apache projects (including Tomcat, Ant, Log4j and Xalan) and jEdit (a popular text/code editor). We refer to this dataset as JURE.

We thus considered 23 systems in our study, containing a total of 11.4k files and 2.5m LOC . Note that all those systems from the DAMB and JURE datasets are Java based. The systems we selected here contain different levels and proportions of defective modules. General information about the systems in the datasets is shown in Table 1. Figure 4 shows the distribution of defective modules across all systems in both datasets. More detailed information about the selected systems, including versioning information, are provided in the original studies (D'Ambros et al. 2012b; Jureczko and Madeyski 2010).

To inform the models' selection, we examined the normality of the various data distributions for and LOC and defects using the Shapiro-Wilk test. For the DAMB and JURE datasets, we test the null hypothesis that the data follow a normal distribution. The results of this test lead us to reject the null hypothesis that the samples come from a normal distribution. We provide two examples of the shape of defects from our datasets in Fig. 5 (for defects in Log4J (JURE dataset) and Eclipse Equinox (DAMB dataset)). These histograms visually confirm that those systems (and other systems that exhibit a similar patter) tend to follow a distribution that is very far from Gaussian distribution; most defects are concentrated in certain classes or files, and most files contain few defects. We also use probability plots (a graphical technique to assess if the data follows a given distribution) to plot the distribution of class size. Figures 6a and 7a show probability plots for LOC in Eclipse PDE (DAMB dataset) and Apache Xalan (JURE dataset), respectively. In general, classes with very prominent modes are larger than lower mode classes, while the majority of classes are under 500 LOC. Figures 6b and 7b show that the shape of the LOC data tends to follow

Table 1 Details of the systems used in the empirical study

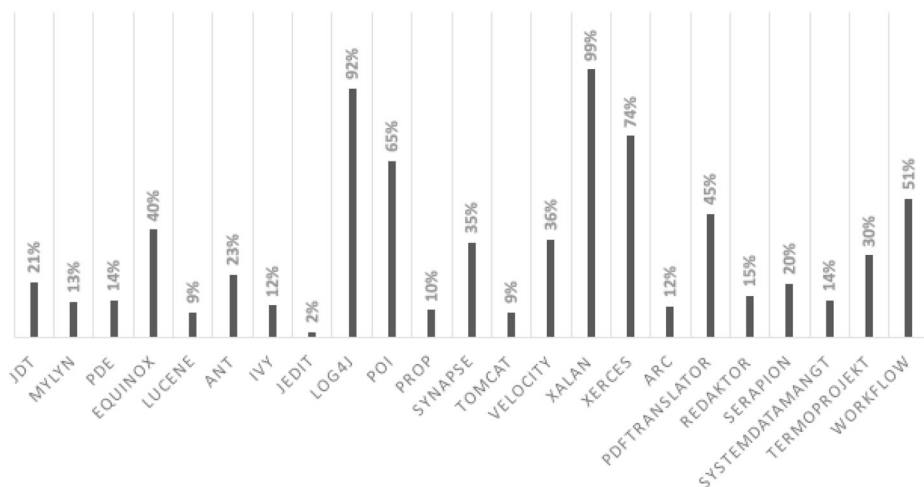| Dataset	#projects	NOC	LOC	#defects	#defective classes	%defective classes
DAMB	5	5,238	639,826	1,740	843	16%
JURE	18	6,206	1,879,894	4,863	2,419	39%
Total	23	11,444	2,519,720	6,603	3,262	29%

Fig. 4 Percentage of defective modules (files) across all systems

a normal (or near normal) distribution after transformation (discussed in more details in Section 3.3).

We studied the effect of class size (using the transformed LOC (i.e., $\log(LOC)$)) on the relationship between each of the following OO metrics and the number of defects/defect-proneness: RFC, WMC, CBO, DIT, LCOM, Fan-in and Fan-out as shown in Table 2. These metrics cover several OO design quality criteria representing the internal and external complexity of the class. These include coupling, cohesion, inheritance and complexity. RFC, CBO, WMC, LCOM and DIT are part of the CK metrics suite (Chidamber and Kemerer 1994).

We apply our mediation and moderation analysis procedures to seven OO metrics, LOC and defects. For each system in our datasets, the total number of tests conducted is $4 \times 7 = 28$ tests for both moderation and mediation analysis. We conducted all analysis on R^2 on a machine running on a 2.6 GHz Intel Core i7 processor, and 32 GB RAM. Each of our analysis scripts ran in under 1 hour. For reproducibility and replication, we provide all results and scripts in our online repository (Tahir 2021).

4 Results and Discussion

As explained in Section 3.3, we followed a three-steps approach to investigate the potential effect of size. We first examine the associations between OO metrics and the number of defects. Given that the data are not normally distributed, we use the non-parametric Kendall's tau (τ) correlation coefficient, and correlation strength is interpreted using Cohen's classification (low correlation when $0 < \tau \le .29$, medium when $.30 \le \tau \le .49$ and high when $\tau \ge .50$). We also examine the correlation between the size metric (measured in LOC), OO metrics and the number of defects.
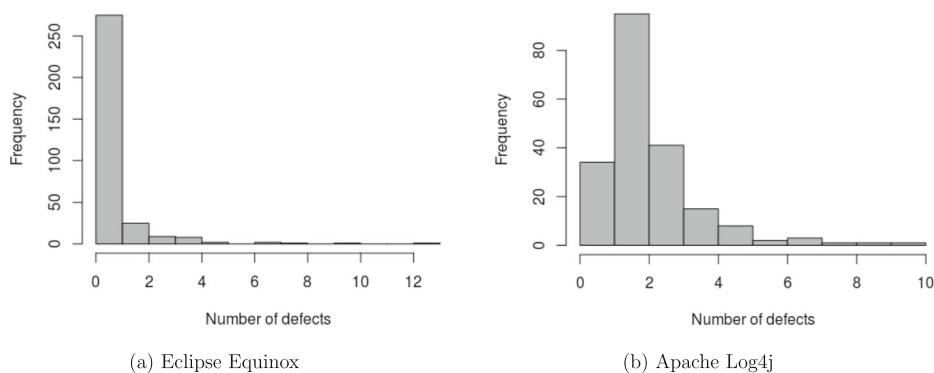
---

[2]Version 3.6.0

(a) Eclipse Equinox   (b) Apache Log4j

Fig. 5 Distribution of defects in two selected systems from DAMB and JURE datasets

(a) before   (b) after
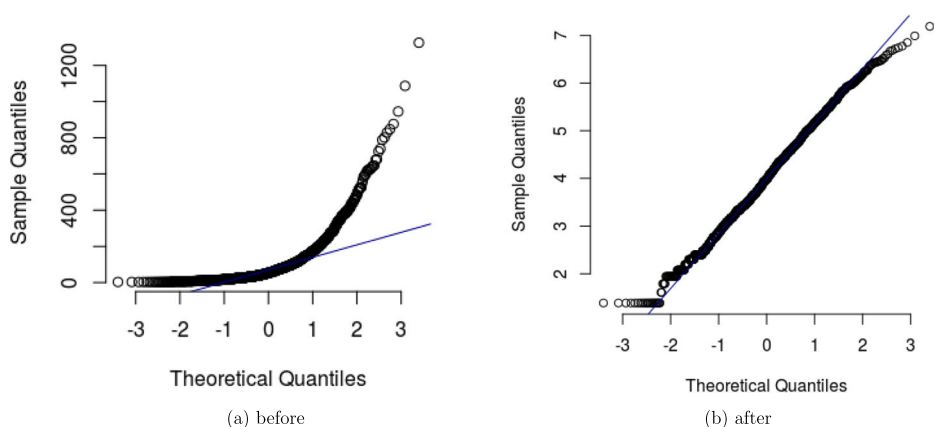
Fig. 6 Probability plots of LOC in Eclipse PDE before and after transformation

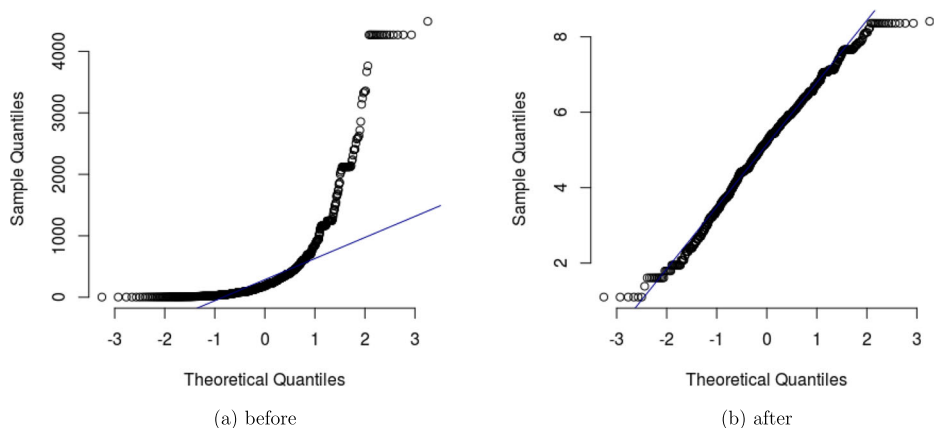(a) before   (b) after

Fig. 7 Probability plots of LOC in Apache Xalan before and after transformation

Table 2 List of metrics used

Metric	Full name	Metric	Full name
LOC	Lines of Code	CBO	Coupling between Objects
RFC	Response for Class	DIT	Depth of Inheritance Tree
WMC	Weighted Method Count	LCOM	Lack of Cohesion in Methods
Fan-in	No. of other classes that reference the class	Fan-out	No. of classes referenced by the class

4.1 The Correlation Between Size, Defects and OO Metrics

Several previous studies used bivariate correlations and univariate logistic regressions to establish the validity of those metrics in predicting defects (Binkley and Schach 1998; Harrison et al. 1998) or defect-proneness (Briand et al. 2000; Tang et al. 1999). In this section, we examined the bivariate correlations between size, OO metrics and the number of defects across for each of the systems in both datasets. We compute Kendall's τ (Tau) correlation test between the individual metrics and defects for both datasets' (Tables 3 and 4). The correlation values for both count of defects and binary defects were mostly significant. Positive and significant correlations between the pairs (metric and defects) are observed across almost all datasets. Few insignificant correlation values (highlighted values in grey) are observed in the JURE datasets. The Fan-in metric has most of the insignificant correlation values whereas LOC, Fan-out, RFC and WMC metrics had more significant correlation values across all datasets. There were no insignificant correlation values for the DAMB datasets indicating the strong correlation between all individual metrics and defects.

The results show that class size is significantly correlated with several of our OO metrics. LOC is also significantly correlated with the number of defects. It was also found that all of the OO metrics except DIT are strongly correlated with the number of defects and defect-proneness.[3] We then used regression to fit models that would enable us to assess the mediation and moderation effects of size on the number of defects. In this analysis regression is used to explain the strength of the effect of independent variables on the dependent variable, rather than to predict outcomes. We used each OO metric as an independent variable and the number of defects as the dependent variable.

4.2 The Mediation Effect of Class Size (RQ1)

Having confirmed that the OO metrics, size metric and defects are all significantly correlated, and that some of these OO metrics can explain (to varying degrees) the number of defects, we turn our attention to investigate whether size plays a mediating role in the relationships between the OO metrics and the number of defects.

4.2.1 Mediation Effect On the Number Of Defects(RQ1.1)

As we explained in Section 3.3.1, the mediation effect of variables is calculated based on three sequential steps - first by calculating the *total effect c* (Fig. 1), then the *direct effect c'*, and finally the *indirect effect ab* (Fig. 2).

---

[3]all associations are statistically significant at an $p < 0.05$.

Table 3 Kendall τ rank correlation analysis between each individual metric and count/binary defects for DAMB dataset

Metric	Project	Defects	Defective	Metric	Project	Defects	Defective
CBO	JDT	0.31	0.30	Fan-in	JDT	0.17	0.16
	Equinox	0.44	0.41		Equinox	0.37	0.36
	Lucene	0.15	0.15		Lucene	0.15	0.15
	Mylyn	0.20	0.20		Mylyn	0.11	0.10
	PDE	0.18	0.17		PDE	0.05	0.05
Fan-out	JDT	0.33	0.32	LCOM	JDT	0.24	0.23
	Equinox	0.38	0.34		Equinox	0.45	0.42
	Lucene	0.11	0.11		Lucene	0.15	0.15
	Mylyn	0.23	0.23		Mylyn	0.12	0.11
	PDE	0.21	0.21		PDE	0.17	0.17
RFC	JDT	0.32	0.31	WMC	JDT	0.34	0.33
	Equinox	0.41	0.38		Equinox	0.44	0.41
	Lucene	0.12	0.12		Lucene	0.14	0.14
	Mylyn	0.19	0.19		Mylyn	0.14	0.14
	PDE	0.22	0.22		PDE	0.19	0.19
LOC	JDT	0.33	0.32				
	Equinox	0.42	0.40				
	Lucene	0.13	0.13				
	Mylyn	0.19	0.18				
	PDE	0.21	0.21				

All values are significant (p >0.05)

Note that we did not conduct any mediation or moderation analysis for DIT. While DIT is significantly correlated with the number of defects (with a rather weak negative correlation), it does not significantly explain the number of defects. This violates the first and second steps of our three-step procedure. Therefore, DIT was eliminated from any further analysis.

As there are results from 23 different systems acress two different datasets, we present here a detailed explanation from only two selected systems (one taken from each dataset), and we include detailed results for all other individual systems in our online repository (Tahir 2021)

We present the results of the mediation analysis of class size on the number of defects from two systems: Eclipse Mylyn (DAMB) and Apache Xerces (JURE). These two systems were selected because they are well-known, developed by well-established communities and they are two of the largest systems in each of the datasets. The results of the mediation effect of class size on the relationship between OO metrics and the number of defects in Mylyn are presented in Table 5. The *total effect* (c) regression model for RFC explaining the number of defects, ignoring the mediator, was significant ($p < 0.001$, Estimate= 1.68e-03). The direct effect (c') of RFC on defects - by introducing the mediator LOC as a further explanatory factor in the model - was insignificant ($p < 0.91$, Estimate= -3.49e-05). Failing in this second step suggests that a mediation effect of class size is unlikely, although the indirect effect was significant ($p < 0.001$, Estimate= 1.71e-03), but quite weak.

Table 4 Kendall τ rank correlation analysis between each individual metric and count/binary defects for JURE dataset

Metric	JURE DATASETS Project	# Defects	Defective	Project	# Defects	Defective	Project	# Defects	Defective	Project	# Defects	Defective
CBO	Ant	0.29	0.28	PDFTranslator	0.45	0.55	Synapse	0.24	0.24	WorkFlow	−0.20	−0.20
	arc	0.24	0.24	poi-3.0	0.40	0.35	SystemData	0.31	0.32	Xalan	−0.01	0.07
	Ivy	0.25	0.25	Prop	0.14	0.14	TermoProjekt	0.27	0.25	Xerces	0.50	0.53
	jEdit	0.07	0.07	Redaktor	0.05	0.06	Tomcat	0.25	0.25			
	log4j	0.28	0.13	Serapion	0.31	0.30	Velocity	0.26	0.27			
Fan-in	Ant	0.12	0.11	PDFTranslator	0.43	0.42	Synapse	0.21	0.20	WorkFlow	−0.27	−0.27
	arc	−0.11	−0.11	poi-3.0	0.20	0.13	SystemData	0.24	0.24	Xalan	−0.08	0.00
	Ivy	0.07	0.07	Prop	0.01	0.01	TermoProjekt	0.13	0.10	Xerces	0.32	0.41
	jEdit	0.04	0.04	Redaktor	−0.06	−0.05	Tomcat	0.16	0.16			
	log4j	0.27	0.14	Serapion	0.40	0.40	Velocity	0.01	0.01			
Fan-out	Ant	0.30	0.29	PDFTranslator	0.46	0.52	Synapse	0.21	0.20	WorkFlow	0.20	0.20
	arc	0.28	0.28	poi-3.0	0.40	0.39	SystemData	0.25	0.25	Xalan	0.03	0.07
	Ivy	0.17	0.16	Prop	0.12	0.12	TermoProjekt	0.39	0.40	Xerces	0.55	0.55
	jEdit	0.11	0.11	Redaktor	0.14	0.14	Tomcat	NA	NA			
	log4j	0.12	0.01	Serapion	0.05	0.05	Velocity	0.31	0.33			
LCOM	Ant	0.33	0.33	PDFTranslator	0.57	0.63	Synapse	0.15	0.15	WorkFlow	0.08	0.08
	arc	0.18	0.18	poi-3.0	0.28	0.33	SystemData	0.29	0.27	Xalan	0.11	0.05
	Ivy	0.24	0.24	Prop	0.12	0.12	TermoProjekt	0.33	0.33	Xerces	0.12	0.01
	jEdit	0.05	0.05	Redaktor	−0.13	−0.13	Tomcat	0.18	0.18			
	log4j	0.12	0.07	Serapion	0.51	0.50	Velocity	0.08	0.12			

Table 4 (continued)

JURE DATASETS

Metric	Project	# Defects	Defective	Project	# Defects	Defective	Project	# Defects	Defective	Project	# Defects	Defective
RFC	Ant	0.40	0.39	PDFTranslator	0.50	0.54	Synapse	0.34	0.34	WorkFlow	0.31	0.31
	arc	0.18	0.18	poi-3.0	0.42	0.40	SystemData	0.28	0.27	Xalan	0.14	0.09
	Ivy	0.29	0.29	Prop	0.16	0.16	TermoProjekt	0.47	0.48	Xerces	0.35	0.23
	jEdit	0.05	0.05	Redaktor	0.00	0.00	Tomcat	0.26	0.26			
	log4j	0.16	0.05	Serapion	0.34	0.33	Velocity	0.23	0.25			
WMC	Ant	0.35	0.34	PDFTranslator	0.48	0.53	Synapse	0.25	0.25	WorkFlow	0.19	0.19
	arc	0.19	0.19	poi-3.0	0.37	0.40	SystemData	0.25	0.24	Xalan	0.07	0.07
	Ivy	0.28	0.28	Prop	0.14	0.15	TermoProjekt	0.46	0.47	Xerces	0.23	0.10
	jEdit	0.05	0.05	Redaktor	-0.10	-0.10	Tomcat	0.20	0.20			
	log4j	0.22	0.13	Serapion	0.52	0.50	Velocity	0.21	0.25			
LOC	Ant	0.39	0.38	PDFTranslator	0.44	0.48	Synapse	0.33	0.33	WorkFlow	0.27	0.27
	arc	0.15	0.15	poi-3.0	0.38	0.37	SystemData	0.25	0.24	Xalan	0.40	0.10
	Ivy	0.29	0.29	Prop	0.16	0.16	TermoProjekt	0.48	0.48	Xerces	0.43	0.32
	jEdit	0.05	0.05	Redaktor	0.02	0.02	Tomcat	0.26	0.26			
	log4j	0.18	0.01	Serapion	0.35	0.35	Velocity	0.24	0.23			

Non-significant correlation results (p >0.05) are highlighted in gray

Table 5 The mediation effect of class size with the number of defects in Mylyn

Metrics	Indirect Effect		Direct Effect		Total Effect		Significant
	Estimate	p-value	Estimate	p-value	Estimate	p-value	
RFC	1.71e-03	**<0.001**	-3.49e-05	0.91	1.68e-03	**<0.001**	
WMC	0.004267	**<0.001**	-0.000529	0.29	0.003737	**<0.001**	
CBO	0.005006	**<0.001**	0.001439	**<0.002**	0.006445	**<0.001**	✓
LCOM	3.43e-04	**<0.001**	-3.87e-05	0.37	3.04e-04	**<0.001**	
Fan-in	0.003800	**<0.01**	0.001145	0.10	0.004945	**<0.001**	
Fan-out	0.00854	**<0.001**	0.00716	**<0.001**	0.01570	**<0.001**	✓

The same inference applies to the WMC, LCOM and Fan-in metrics (i.e., the total and indirect effects are significant, but the direct effect is insignificant). Therefore, it is concluded that class size *does not fully mediate* the relationship between the number of defects and these metrics: RFC, WMC, LCOM, Fan-in and Fan-out in Mylyn. On the other hand, for CBO, the total effect (c) of CBO on the number of defects was found to be significant ($p < .001$, Estimate= 0.0027). The direct effect (c'), as we introduce the mediator LOC in the model, was also significant ($p < .001$, Estimate= 0.0013). We then estimate the indirect effect which was also significant ($p < .001$, Estimate= 0.0014) - the indirect effect of CBO on defects interval does not include zero (LLCI= 0.0009, ULCI= 0.0001); i.e., zero does not fall between the bootstrapping LLCI and ULCI values. The same applies to the Fan-out metric, which exhibits a significant total, direct and indirect effects (all at $p < 0.001$). Therefore, it is inferred that *size fully mediates* the relationship between CBO and Fan-out, and the number of defects.

The results of the mediation analysis for Xerces (JURE dataset) are shown in Table 6. The regression model that uses RFC to explain the variance in the number of defects while ignoring the mediator (*total effect*) was significant ($p < 0.001$, Estimate= 0.0325). The *direct effect* (c') of RFC on defects - by introducing the mediator ($\log(LOC)$) as a further explanatory factor in the model - is also significant ($p < 0.001$, Estimate= 0.0113). A concurring outcome was evident for the *indirect effect* of RFC on the number of defects, which was also significant ($p < 0.001$, Estimate= 0.0212) and zero does not fall between the interval values (LLCI= .0013, ULCI= 0.03), which indicates that (unlike the results in Mylyn) size actually *mediates* the relationship between RFC and number of defects. This is the same for the following metrics: CBO (total effect= 0.1008, direct effect= 0.0279, indirect effect = 0.0630, all significant at $p < 0.001$), Fan-in (total effect= 0.0993, direct effect= 0.02959, indirect effect = 0.0697, $p < 0.05$) and Fan-out (total effect= 0.1783, direct effect= 0.0671, indirect effect = 0.1112, significant at $p < 0.001$). These results indicate that class size indeed *mediates* the relationships between RFC, CBO, Fan-in and Fan-out and the number of defects in Xerces. However, for the other two metrics for Xerces (i.e., WMC and LCOM), class size does *not* mediate their relationships with the number of defects. For example, WMC shows a significant total effect ($p < 0.001$, Estimate= 0.0744) but an insignificant direct effect ($p < 0.21$, Estimate= 0.0099), suggesting that the mediation effect of size is unlikely.

The same analysis process was enacted on all 23 systems. For enhanced readability, we provide only summary results of this analysis in Table 7 (though detailed results from each system are available externally (Tahir 2021)). The table shows a summary of whether class size has a mediation effect for each single metric. As shown in Table 7, there are

Table 6 The mediation effect of class size with the number of defects in Xerces

Metrics	Indirect Effect		Direct Effect		Total Effect		Significant
	Estimate	p-value	Estimate	p-value	Estimate	p-value	
RFC	0.02124	**<0.001**	0.01129	**<0.001**	0.03253	**<0.001**	✓
WMC	0.06455	**<0.001**	0.00988	0.21	0.07442	**<0.001**	
CBO	0.0630	**<0.001**	0.0379	**<0.001**	0.1008	**<0.001**	✓
LCOM	0.003550	**<0.001**	0.000439	0.59	0.003989	**<0.001**	
Fan-in	0.06969	**<0.001**	0.02959	**<0.05**	0.09928	**<0.001**	✓
Fan-out	0.1112	**<0.001**	0.0671	**<0.001**	0.1783	**<0.001**	✓

mixed results regarding the true mediation effect of class size across all systems examined. Considering the results for the DAMB dataset, we observed that size appears to have a mediation effect on the relationship between CBO and the number of defects in 4/5 systems (all systems except PDE), and on Fan-out in 3/5 systems. For the other metrics, there is no consistent evidence of a mediation effect for class size, given that there was no significant mediation effect in more than two systems.

A similar pattern was found in the analysis of the JURE dataset, where only a few systems show a significant effect of class size on the relationship between metrics and the number of defects. Some systems such as JDT, Ivy and Xerces show a mediation effect of size for 4 or 5 metrics, where other systems such as Arc and PDFTranslator do not show a mediation effect for any of the metrics examined. The mediation effect of size on the relationship between CBO and defects appears in 9/18 systems. A number of systems also show a mediation effect of size on a other coupling metrics, i.e., Fan-out and RFC, where the mediation effect of size appears in 7 systems each. Another interesting observation from the results is that, for all the proprietary systems we included from the JURE dataset, none of the metrics seemed to be mediated by size (except for CBO in Serapion). However, given the still limited number of systems we investigated and the size of these systems we are not able to generalise these findings to other proprietary systems. Further investigation is needed for such systems.

Overall, WMC, LCOM and Fan-in are the metrics with the least evidence of a mediation effect of size. We could not find evidence of a significant mediation effect of size on the relationship between those two metrics and the number of defects in the majority of systems we investigated (only in 2, 3 and 4 systems, respectively). The mediation effect on all six metrics seems to be different between systems as well. When comparing the results we obtained by analyzing the DAMB dataset with those found prior in the study of Zhou et al. (2014) (using the same dataset), we found similar evidence of a mediation effect of size for CBO only, as the evidence of the mediation effect of size was consistent in 4 of the 5 systems. However, unlike the previous analysis (Zhou et al. 2014), we could not confirm if the effect of size is significant for the other metrics: RFC, WMC, LCOM, Fan-in or Fan-out metrics, simply because the results were largely inconsistent across different systems.

In summary, it is observed that evidence regarding the mediation effect of size is *inconsistent* and does not follow a pattern across all systems. For all metrics, class size is not shown to have a significant mediation effect across all systems. Only CBO shows a more consistent significant mediation effect of size than other metrics, where a significant indirect effect was found in 13 out of 23 systems. There is less evidence of the effect of size on Fan-out, where the mediation effect is significant in 10 systems across both datasets. However, we still consider these results inconclusive in determining if the indirect effect is

Table 7 Summary of the mediation effect of class size in count models

Systems		RFC	WMC	CBO	LCOM	Fan-in	Fan-out
DAMB	JDT	✓	✓	✓	✓		✓
	Mylyn			✓			✓
	PDE						
	Equinox		✓	✓	✓	✓	
	Lucene			✓		✓	✓
JURE	Ant	✓					✓
	Ivy	✓		✓		✓	✓
	jEdit						
	Log4j			✓			
	POI	✓		✓			✓
	Prop				✓		
	Synapse	✓		✓			✓
	Tomcat	✓		✓			NA
	Velocity			✓			✓
	Xalan	✓		✓			✓
	Xerces	✓		✓		✓	✓
	Arc						
	PDFTranslator						
	Redaktor						
	Serapion			✓			
	SystemDataMangt						
	TermoProjekt						
	WorkFlow						

✓ the *mediation* effect of size is significant

NA: metric data is not available

truly and always significant. In returning to our hypothesis (H01.1), based on the evidence resulting from our analysis of the individual systems in both datasets, we reject the null hypothesis that class size has no mediation effect on the relationship between the number of defects and the following OO metrics: RFC, WMC, LCOM, Fan-in metrics. For CBO, while the mediation effect appears in the majority of systems, the evidence of this effect is more consistent in DAMB dataset (4/5) than in JURE dataset (9/18).

4.2.2 Mediation Effect on Defect-Proneness (RQ1.2)

Having investigated the effect of class size on the number of defects in the previous Section (4.2.1), we now investigate the mediation effect of class size on the presence of defects in software modules (e.g., defect-proneness models). Similar to the previous section, we present the results of mediation analysis for a system from each dataset: Eclipse Mylyn and Apache Xerces in Tables 8 and 9, respectively.

For the mediation analysis conducted on Mylyn (Table 8), the *total* and *indirect effects* of RFC on defect-proneness intervals were significant and zero does not lie between the LLCI

Table 8 The mediation effect of class size with defect-proneness in Mylyn

Metrics	Indirect Effect		Direct Effect		Total Effect		Significant
	Estimate	p-value	Estimate	p-value	Estimate	p-value	
RFC	4.51e-04	**<0.001**	2.20e-04	0.11	6.71e-04	**<0.001**	
WMC	1.33e-03	**<0.001**	-2.92e-05	0.97	1.31e-03	**<0.001**	
CBO	0.001426	**<0.001**	0.001298	**<0.001**	0.002724	**<0.001**	✓
LCOM	1.06e-04	**<0.001**	-2.40e-05	0.31	8.19e-05	**<0.01**	
Fan-in	0.001076	**<0.001**	0.000744	0.09	0.001821	**<0.01**	
Fan-out	0.001349	0.11	0.006553	**<0.001**	0.007903	**<0.001**	

and ULCI (indirect effect: $p < 0.01$, Estimate= 4.51e-04), however, the *direct effect* (c') regression model for RFC explaining defect-proneness was not significant ($p < 0.11$, Estimate= 4.51e-04), which inferred that size does not fully mediate the relationship between RFC and defect-proneness. The same conclusion is observed for the relationships between defect-proneness and WMC (*direct effect*: $p < 0.97$, Estimate= -2.92e-05), LCOM (*direct effect*: $p < 0.31$, Estimate= -2.40e-05) and Fan-in (*direct effect*: $p < 0.09$, Estimate= 0.0007=). For Fan-out, while both the *direct* and *total effects* were found to be significant at $p < 0.001$ (with an Estimate of 0.0066 and 0.0079, respectively), the *indirect effect* was insignificant ($p < 0.11$, Estimate= 0.0013, LLCI = -0.0003, ULCI = 0). This also indicates the the mediation effect of size is questionable. On other hand, we observed a mediation effect of size on the relationship between CBO and defect-proneness in Mylyn as all three effects were found to be significant at $p < 0.001$. The indirect effect was significant with Estimate= 0.0014. This observation shows that size has a significant mediation effect on the relationship between CBO and class defect-proneness.

In Xerces (Table 9), for the RFC, CBO, Fan-in and Fan-out metrics, all three *direct*, *indirect* and *total effects* regression models for those metrics explaining defect-proneness were significant at $p < 0.001$. For example, size has an *indirect effect* on the relationship between CBO and defect-proneness with $p < 0.001$ and Estimate= 0.0630. The mediation effect of class size is considered to be significant for four metrics. However, for WMC and LCOM, while the *total* and *indirect* effects were significant, the *direct effect* of size was insignificant (violating the second step). In the case of LCOM, the *indirect effect* is insignificant with $p < 0.59$, and Estimate= 0.0004, indicating that the mediation effect of class size is improbable.

Table 9 The mediation effect of class size with defect-proneness in Xerces

Metrics	Indirect Effect		Direct Effect		Total Effect		Significant
	Estimate	p-value	Estimate	p-value	Estimate	p-value	
RFC	0.02124	**<0.001**	0.01129	**<0.001**	0.03253	**<0.001**	✓
WMC	0.06455	**<0.001**	0.00988	0.21	0.07442	**<0.001**	
CBO	0.0630	**<0.001**	0.0379	**<0.001**	0.1008	**<0.001**	✓
LCOM	0.003550	**<0.001**	0.000439	0.59	0.003989	**<0.001**	
Fan-in	0.06969	**<0.001**	0.02959	**<0.05**	0.09928	**<0.001**	✓
Fan-out	0.1112	**<0.001**	0.0671	**<0.001**	0.1783	**<0.001**	✓

When looking at the two datasets combined (Table 10), we observed that the mediation effect of size is significant for CBO in 13 systems, with particularly strong evidence in the DAMB dataset (being evident in all systems except PDE). The size effect on CBO is also observed in 9 other systems in JURE dataset. This is similar to the pattern reported for the count models in Section 4.2.1. None of the other metrics have shown similar/consistent results in either of the datasets. Unlike the results from the count models, there is less evidence of the mediation effect of size on RFC and Fan-out (i.e., significant in only 5 systems across both datasets). LCOM is the metric with the second highest number of systems where the effect of size is found to be significant (found in 7/23 systems). Therefore, the results from all metrics except CBO are considered to be diverse and do not follow a consistent pattern across all datasets/systems.

To summarise the results from the defect-proneness analysis, which are consistent with the results from the count defect models, it is observed that evidence regarding the mediation effect of class size is still discordant and does not follow a specific pattern across all systems. Again, only CBO shows a more consistent result, as size appears to have indirect

Table 10 Summary of the mediation effect of class size in binary models

Systems		RFC	WMC	CBO	LCOM	Fan-in	Fan-out
DAMB	JDT	✓	✓	✓	✓		✓
	Mylyn			✓			
	PDE						
	Equinox		✓	✓	✓	✓	
	Lucene			✓		✓	✓
JURE	Ant	✓					✓
	Ivy						
	jEdit						
	Log4j						
	POI			✓			
	Prop		✓	✓	✓		
	Synapse			✓			
	Tomcat	✓		✓			NA
	Velocity		✓	✓	✓		
	Xalan	✓	✓	✓	✓		✓
	Xerces	✓		✓		✓	✓
	Arc						
	PDFTranslator			✓	✓		
	Redaktor						
	Serapion			✓	✓		
	SystemDataMangt						
	TermoProjekt						
	WorkFlow						

✓ the *mediation* effect of size is significant

NA: metric data is not available

effects on the relationship between CBO and class defect-proneness in 13 systems (4 systems in DAMB and 9 systems in JURE). Therefore, we consider the mediation effect of class size on defect-proneness to have a potential impact only on the relationship between CBO and defect-proneness. In returning to our research hypothesis (H01.2), we reject the null hypothesis for the following metrics RFC, WMC, LCOM, Fan-in and Fan-out but we cannot reject the null hypothesis for CBO as the results are mixed and we could not find strong evidence of the absence of the mediation effect of class size.

When looking at the results from both sets of models- i.e., count (Section 4.2.1) and binary (Section 4.2.2), we observe that only CBO resulted in the same significant mediation effect of class size on the relationship with defects/defect-proneness. Eleven (out of thirteen) of the systems that show a mediation effect of size in the count models also show a mediation effect in the binary models. This suggests that CBO is indeed the metric that is mostly impacted by the mediation effect of class size. Our results demonstrate that there is either a lack of statistical evidence (or in many cases, no strong evidence) of a significant mediation effect of size in defect count and binary defect prediction models. The only metric that looks vulnerable to the effect of size is CBO.

4.3 The Moderation Effect of Class Size (RQ2)

We now look at the moderation (interaction) effect of class size. Similar to the mediation analysis, we study the moderation effect of size in models explaining both the number of defects (count) and defect-proneness (binary). To investigate the moderation effect of class size on the number of defects, we empirically assessed the impact of all six OO metrics on the number of defects under the influence of class size, and report the results for two systems separately (one selected from each dataset to enhance readability) - Mylyn and Xerces. As explained in Section 3.2, the moderation effect is estimated using multiple regression analysis. We considered the moderation effect of class size on the relationship between individual OO metrics and the number of defects/defect-proneness to be significant when the interaction effect is significant ($p < .05$).

To graphically illustrate the moderation effect of class size on the relationship between the object-oriented metrics and defects, three levels of the moderator (mean, one standard deviation below the mean (-SD) and one standard deviation above the mean (+SD)) are computed and then plotted using simple slopes. The plots are generated using the rockchalk function (Johnson and Grothendieck 2012) available in R.

4.3.1 Moderation Effect on the Number of Defects (RQ2.1)

Table 11 presents the moderation effect results for Mylyn and Xerces from the DAMB and JURE datasets, respectively. In Table 11, our moderation analysis reveals that the interaction effects of size on defects for Mylyn were significant for four metrics (RFC, LCOM, Fan-in and Fan-out), all with significant interaction terms ($p < 0.05$). For WMC and CBO, this indicates that there is no significant relationship between these metrics and the number of defects. Similarly, for Xerces, we observed significant moderation effects of LOC on the individual OO metrics excluding the LCOM ($p = 0.229$) and Fan-in ($p = 0.452$) metrics. Looking at both systems, the moderation effect of size appears to be evident only for the RFC and Fan-out metrics.

We then present examples of the moderation effects of size on the relationship between Fan-out and defects for Eclipse Mylyn (DAMB dataset) in Fig. 8, and Fan-out and defects

Table 11 Moderation effect of size on the relationship between individual OO metrics and defects for Mylyn (DAMB Dataset) and Xerces (JURE Dataset): count models

Eclipse Mylyn				Apache Xerces			
Metrics	Estimate	p-value	Significant	Estimate	p-value	Significant
RFC	−0.0023606	**<0.001**	✓	−0.0025988	**<0.001**	✓
WMC	−0.0005414	0.425		0.004101	**<0.001**	✓
CBO	0.002356	0.122		0.003280	**<0.01**	✓
LCOM	−1.695e−04	**<0.01**	✓	−7.379e−05	0.229	
Fan−in	0.005272	0.0481	✓	0.001427	0.452	
Fan−out	−0.009447	**<0.01**	✓	−0.009595	**<0.001**	✓

for Apache Xerces (JURE dataset) in Fig. 9. We include graphs from all other systems in our online repository (Tahir 2021).

As shown in Fig. 8, we observe that small class size values (black line) with small Fan-out values resulted in identifying more defects . Large size values (green dotted line) together with small Fan-out values initially resulted in identifying more defects than the average (blue dashed line). However, as the Fan-out values increased, the number of defects identified by the small and large size values interchanged - i.e., more defects were identified than average when size was small, and more defects were identified also when size was large, but it was less than the number identified by the average size values. Regarding the Xerces results, the simple slope plot in Fig. 9 shows a similar trend to that of the Mylyn project in Fig. 8. From Fig. 9, we observe that small size values (the black line) leads to more defects being found with large Fan-out values but leads to less defects found overall on average (the blue dashed line). Large size values (the green dotted line) with large Fan-out values help find more defects than average. However, as Fan-out increases beyond 35, the number of defects identified by the small size and large size values interchange where the small size values identify more defects than the large size values. This suggests that the Fan-out metric has a high probability of explaining defects count in large classes (where LOC is high). The difference in the slopes for small and large values of size confirms that size does moderate the relationship between Fan-out and the number of defects.

The moderation analysis results for all systems in all datasets are shown in Table 12. We only provide the summarised results, which show whether size has a moderation effect on the relationship between each single metric and defects. Overall, the results were mixed. Significant moderation effects are observed for all metrics for Lucene. Three systems show significant moderation effects for four metrics (i.e., Mylyn, jEdit and Xerces), although for different combinations of metrics. On the other hand, seven systems (i.e., Equinox, Ivy, Log4j, Velocity, PDFTranslator, TemoProjekt and Workflow) show no moderation effect of class size on any of the examined metrics (TemoProjekt and Workflow also shows no mediation effect - see Tables 7 and 10). The moderation effect of size on RFC and Fan-out metrics was significant in 10 systems across both datasets, while the moderation effect of size on LCOM was significant in only five systems. The moderation effect of size on the other four remaining metrics was significant in fewer than 5 systems. Class size seems to moderate the relationship between LCOM and defects in 3/5 systems in the DAMB dataset, but this was not the case in the JURE dataset (only 2/18 systems). For Fan-in, the moderation effect was also significant in two systems in the DAMB dataset but was significant in only two of the eighteen systems in the JURE dataset. While we have found evidence of a
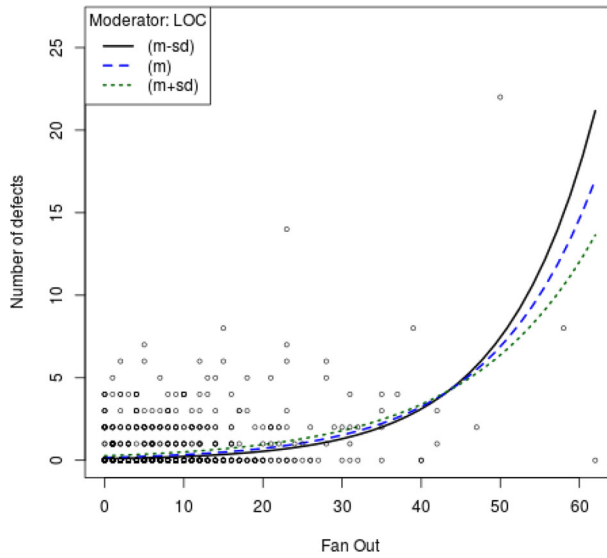
Fig. 8 Moderation effect of size on the relationship between Fan-out and the number of defects in Eclipse Mylyn

mediation effect of size on the relationship between CBO and the number of defects, we can confirm that there is no *moderation* effect of size for the same metric.

In summary, it is observed that evidence regarding a moderation effect of size is *inconsistent* and does not follow a pattern across all systems. Only RFC and Fan-out show a consistent and more significant moderation effect of size than other metrics, where a significant moderation effect was found in 10 systems. Returning to our research hypothesis (H02.1), we reject the null hypothesis that class size has no significant moderation effect

Fig. 9 Moderation effect of size on the relationship between Fan-out and the number of defects in Apache Xerces
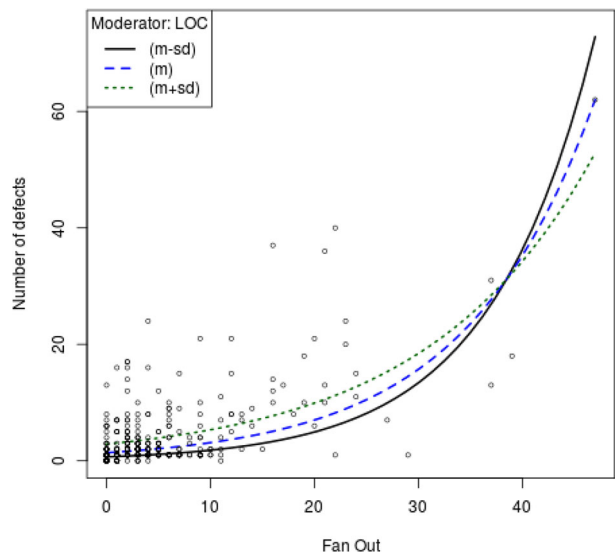
Table 12 Summary of the moderation effect of class size in count models

Systems		RFC	WMC	CBO	LCOM	Fan-in	Fan-out
DAMB	JDT	NA					✓
	Mylyn	✓			✓	✓	✓
	PDE	✓			✓		
	Equinox						
	Lucene	✓	✓	✓	✓	✓	✓
JURE	Ant	✓					
	Ivy						
	jEdit	✓	✓		✓		✓
	Log4j						
	POI	✓			✓		✓
	Prop		✓				
	Synapse						✓
	Tomcat	✓		✓			NA
	Velocity						
	Xalan	✓					✓
	Xerces	✓	✓	✓			✓
	Arc	✓					✓
	PDFTranslator						
	Redaktor						✓
	Serapion					✓	
	SystemDataMangt					✓	
	TermoProjekt						
	WorkFlow						

✓ the *moderation* effect of size is significant

NA: metric data is not available

on the relationship between the number of defects and the following OO metrics: WMC, CBO, LCOM and Fan-in. For RFC and Fan-out, the evidence available for this effect is not consistent across both datasets. Again, and similar to the mediation effect, we are unable to confirm if a moderation effect will always be present.

4.3.2 Moderation Effect on Defect-Proneness (RQ2.2)

For binary models (i.e., models of defect-proneness), we apply the same analysis conducted for count models (see Section 4.3.1). A significant $p < .05$ with positive estimate value indicates a significant and positive association between the metric and defect-proneness (i.e., an increase in the OO metric corresponds to an increase in the defect-proneness of modules) whereas a negative effect value indicates that the relationship between the metric and defects is negative (i.e., when the value of the OO metric increases, the defect-proneness decreases). For an insignificant p-value, it indicates that there is no significant relationship between the OO metric and defect-proneness.

Table 13 presents the moderation effect results for Eclipse Mylyn and Apache Xerces from the DAMB and JURE datasets, respectively. The moderation analysis conducted for the two systems reveals that the interaction effects for Mylyn were insignificant for all but two metrics (RFC and CBO). However, for Xerces, we observed significant moderation effects of size on the individual OO metrics excluding the CBO ($p = 0.68$) and Fan-in metrics ($p = 0.50$).

The simple slope analysis shown in Figs. 10 and 11 visually illustrates the moderation effects of size on the relationship between CBO and defect-proneness in Mylyn (DAMB dataset) and the moderation effect of size on the relationship between LCOM and defect-proneness in Xerces (JURE dataset), respectively. We provide all remaining graphs for all metrics from all other systems in our online repository (Tahir 2021).

Figure 10 shows that small size values (the black line) with large CBO values leads to a less probability of defect-prone modules being found in comparison to the average size values (the blue dashed line). Large size values (the green dotted line) with large CBO values leads to a higher probability of defect-prone modules being found than average (the blue dashed line). Similarly, in Fig. 11, we observe that an initial increase in LCOM with small size values (black line) leads a less probability of defect-prone modules being found which is a less probability than average size (the blue dashed line). However, large size values (green dotted line) with large LCOM values leads to higher probability of defect-prone modules being found which is more than the average. Furthermore, as the LCOM values increases, the probability of defect-prone modules found for small size values and average size values are almost the same. Also, as the LCOM values increase, the large size values leads to the same probability of defect-prone modules being detected as previously detected by the small LCOM values. The difference in the slopes for large or small size values shows that size actually *moderates* the relationship between LCOM and defect-proneness.

Table 14 presents the summarised moderation analysis for all systems. Overall, the moderation effect of size on the relationship between the individual metrics and defect-proneness was insignificant for most of the systems. Only one system (JDT) shows significant moderation effects for 5/6 metrics. The moderation effect of size on Fan-out was significant in 9 systems across both datasets, while the effect on CBO was significant in only two systems. On the other hand, the effect on Fan-in was not significant for any of the systems. A further analysis of this effect across all systems revealed little significant moderation effect of size on the relationship between WMC, CBO and Fan-in metrics and defect-proneness. Class size seems to moderate the relationship between CBO and defect-proneness in two out of 5 systems in DAMB dataset, but this was not the case in the JURE

Table 13 Moderation effect of size on the relationship between individual OO metrics and defects for Mylyn (DAMB Dataset) and Xerces (JURE Dataset): binary models

Metrics	Eclipse Mylyn			Apache Xerces		
	Estimate	p-value	Significant	Estimate	p-value	Significant
RFC	−0.004251	**<0.01**	✓	0.025058	**<0.001**	✓
WMC	−0.0004594	0.761		0.033668	**<0.001**	✓
CBO	0.007743	0.0331	✓	−0.01608	0.68074	
LCOM	−5.576e−05	0.695		0.004701	**<0.001**	✓
Fan−in	0.007078	0.169		−0.01890	0.500953	
Fan−out	−0.007564	0.38306		−0.29659	**<0.001**	✓

Fig. 10 Moderation effect of size on the relationship between CBO and defect-proneness in Mylyn
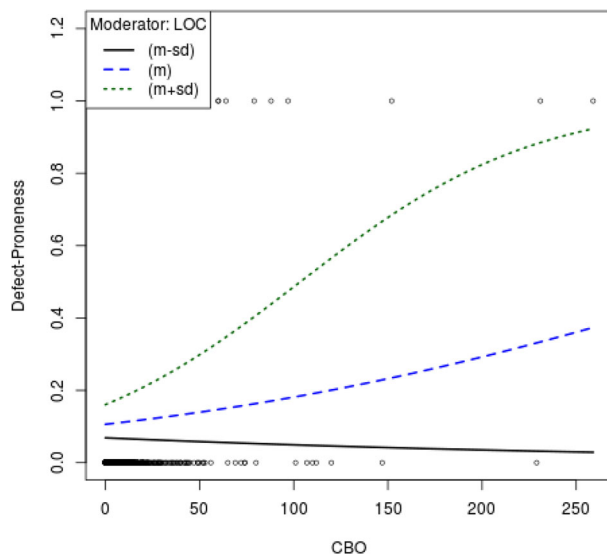
dataset (0 out of the 18 systems). Indeed, the moderation effects were inconsistent across the individual systems, and consequently, we note the evidence regarding the moderation effect of size is not conclusive with regard to these datasets. In summary, it is observed that evidence regarding the moderation effect of size is also *inconsistent* across all systems. The moderation effect of size was significant and considerably consistent on the relationship between only Fan-out and defect-proneness, although this is only found in 9 systems.

Based on our research hypothesis (H02.1), we cannot reject the null hypothesis that class size has no significant moderation effect on the relationship between defect-proneness and the following OO metrics: RFC, WMC, CBO, LCOM and Fan-in. For Fan-out, the evidence

Fig. 11 Moderation effect of size on the relationship between LCOM and defect-proneness in Xerces
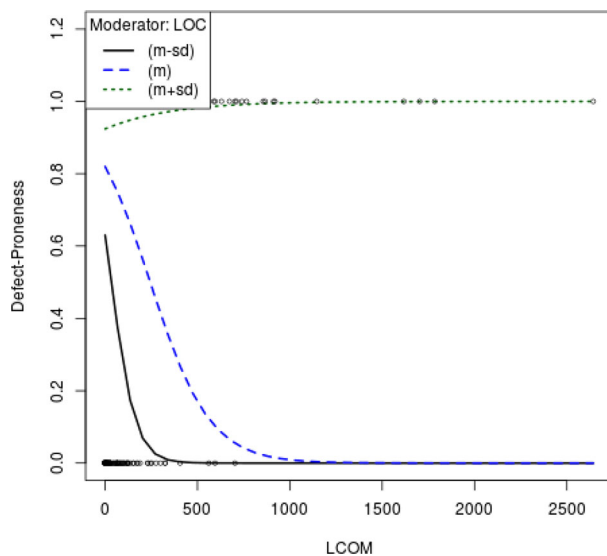
Table 14 Summary of the moderation effect of class size in binary models

Systems		RFC	WMC	CBO	LCOM	Fan-in	Fan-out
DAMB	JDT	✓	✓	✓	✓		✓
	Mylyn	✓		✓			
	PDE	✓					✓
	Equinox						
	Lucene				✓		✓
JURE	Ant						
	Ivy						
	jEdit	✓	✓		✓		
	Log4j						✓
	POI	✓	✓		✓		✓
	Prop						
	Synapse						✓
	Tomcat						NA
	Velocity						✓
	Xalan						
	Xerces	✓	✓		✓		✓
	Arc						
	PDFTranslator						
	Redaktor						✓
	Serapion						
	SystemDataMangt						
	TermoProjekt						
	WorkFlow						

✓ the *moderation* effect of size is significant

NA: metric data is not available

of this effect is not consistent across the two datasets. We are unable to confirm if the moderation effect will always be present.

4.4 A Comparison of the Systems Used

In the previous results section, the concepts of moderation and mediation have been used to interpret the effects of class size and the validity of several OO metrics. All the mediation and moderation analysis results indicate that there is a considerable level of inconsistency between different datasets/systems. A quick look at the simple descriptive statistics (i.e., mean, variance, and correlation of various OO metrics, size and defects) indicates that data collected from different systems exhibit rather different patterns of instability, and that different systems tend to be distinct in their results.

One way to compare any two different systems is by comparing the discrepancy between those systems through statistical hypotheses testing. Most statistical hypothesis tests are designed to accept/reject the pairwise equivalence of the underlying multivariate probability distributions. For example, it is possible to test if data obtained from two different

systems follow the same multivariate distributions (a strict equivalence) or share the same correlation matrix (a weak equivalence). Although conducted, we did not report the results of these hypothesis tests here as both of them reject any possible equivalence between any two systems with extremely small p-values. A direct comparison between any two systems is also not practical given that those systems may contain different number of classes/files and highly-heterogeneous metrics values. The mean and variance of a single metric, size or number of defects in a system are not that informative in our case as we focus on the relationship between different variables. We are seeking a method to measure the similarity between any two systems while considering all metrics (i.e., size, OO metrics and defects).

A scale-free measure is needed to quantify the differences between systems. More specifically, the proposed metric should measure the distance of the inherent relationships of any two systems. Here, we make an effort to compare the systems in our dataset by constructing a suitable measure based on the correlation matrix and matrix norm. The correlation matrix of a system is a straightforward proxy of the internal relationships within the system. A matrix norm will induce a distance between two correlation matrices to compare the internal relationships of any two systems.

For a system with n classes, we collect the OO metrics and the number of defects in an $n \times p$ data matrix X. Here, $p = 9$ variables includes the log size, log number of defects, and seven OO metrics. We used a scale-free correlation matrix, as the correlation coefficients are normalised and fall within a range between -1 and 1. Let's consider the data matrix X of two systems A and B denoted by X_A and X_B. The correlation matrices are $\Sigma_A = \left(\sigma_{ij}^{(A)} \right)$ and $\Sigma_B = \left(\sigma_{ij}^{(B)} \right)$, $1 \leq i, j \leq p$ accordingly. To measure the difference between Σ_A and Σ_B, we use the distance induced by the Frobenius norm as:

$$d(\Sigma_A, \Sigma_B) = \sqrt{\sum_{i,j} \left(\sigma_{ij}^{(A)} - \sigma_{ij}^{(B)} \right)^2} \tag{9}$$

This distance is actually a pairwise measure which can be calculated for any two systems. The results are summarised in a distance matrix D shown in the heatmap in Fig. 12. Given the distance matrix D, we applied a hierarchical clustering algorithm to group similar systems based on the distance between each pair of systems, and then order them by similarity. We linked those clustering and correlation results with the results obtained from our mediation and moderation analysis, with the goal of investigating whether systems that are similar (i.e., in the same cluster group) have comparable results in terms of the mediation or moderation effect of size.

Table 15 provides a summary of the shared mediation and moderation effects of size on the metrics for systems in the same clusters (as shown in Fig. 12). In this table we focus on the top 12 clusters (i.e., clusters with the strongest correlations). As shown in the dendrogram in Fig. 12 and the mediation analysis results in Tables 7 and 10, we observed that the majority of system pairs within the same cluster (high correlation) share the same mediation outcomes (with the majority of them either having no mediation effect 0f size at all, or the mediation effect appears for CBO only). We also observed that systems that are clustered together and having strong correlation tend to share similar mediation more than moderation outcomes. For example, PDE and jEdit appear to be similar as they are in the same cluster and also have strong correlation. In both systems, size does not have any significant mediation effect on any of the OO metrics in both count or binary models. However, the results for the moderation effect of size for those two systems are dissimilar as

Table 15 Shared Mediation and Moderation effect of size on the OO metrics between systems in the same cluster

System A	System B	Shared Mediation		Shared Moderation	
		Count	Binary	Count	Binary
PDE	jEdit	No	No	RFC, LCOM	RFC
Mylyn	Velocity	CBO, Fan-out	CBO	No	No
Serapion	Synapse	CBO	CBO	No	No
PDE	Mylyn	No	No	RFC, LCOM	RFC
Velocity	jEdit	No	No	No	No
Ant	Workflow	No	No	No	No
Redaktor	SystemDataMangt	No	No	No	No
Prop	Lucene	No	CBO	WMC	No
Log4J	PDFTranslator	No	No	No	No
Equinox	POI	CBO	CBO	No	No
Xerces	Prop	No	CBO	WMC	No
Xerces	Lucene	CBO, Fan-in, Fan-out	CBO, Fan-in, Fan-out	RFC, WMC, CBO, Fan-out	LCOM, Fan-out
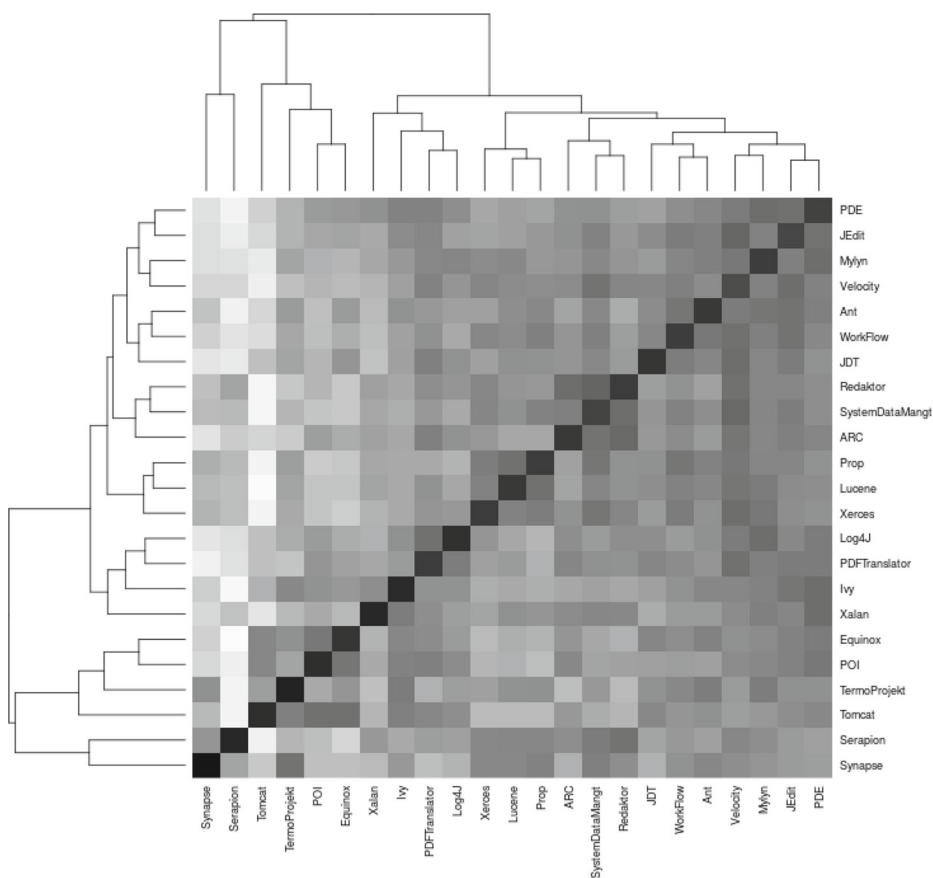
Fig. 12 Matrix heatmap for pairwise significance similarity measure with clustering

the moderation effect of size appears only in RFC and LCOM (refer to Table 15). This is also the case for Mylyn and Velocity, where both systems share a significant mediation effect of size for CBO and Fan-out in count models, and CBO in binary model, but also dissimilar in their moderation results. However, for systems with lower correlation coefficients, the mediation and moderation results appear to be even more distinct. We conclude from this that systems with similar distance measures to some extent share the same mediation results for their individual metrics.

4.5 Implications of the Results

A number of prior studies have looked into the indirect effect of class size in defect models, with the assumption that the confounding effect of size is strong and impacts the performance of such models (El Emam et al. 2001; Zhou et al. 2014). The recommendation of those studies was that size should be controlled to avoid such an impact on the performance of the newly built models. Previous studies that looked into the effect of class size focused on binary defect models (i.e., fit a model to predict defect-proneness) and none looked into the impact of size in count models (i.e., fit a model to predict the number of defects). Our study has sought to address 1) the miscellany of the results from previous studies to confirm the indirect effect of size and 2) the effect of class size when looking at count models. We classified the indirect effect into a *mediation* and *moderation* effect, and we selected suitable statistical procedures to assess both.

The results presented here provide some evidence (although inconclusive) that size is in fact a problem for some OO metrics, and so should at least be considered. When looking into the OO metrics that we studied, the evidence regarding the impact of size is only evident for CBO (in both types of models, count and binary); and even then, it is shown in the majority of systems but not all of them. Other metrics show mixed results; and those results are far from conclusive. In fact, the results presented in this paper are somehow contradictory to those of the previous studies (i.e., El Emam et al. 2001; Zhou et al. 2014). In terms of the moderation effect, another coupling metric, Fan-out, show a moderation effect of size in more systems that other metrics. Based on our in-depth investigation of the conditional (mediation) and interaction (moderation) effect of size, we cannot confirm if size is actually the reason for the relationship between most OO metrics and defects. We consider the effect of size to depend largely on the systems examined. For some systems, the effect of size affected all metrics, whereas for other systems there was no significant effect. This may be due to the nature of each specific system (i.e., type and domain of the system), or the dynamics of each development team (e.g., team experience, number of developers). We did not investigate these additional factors any further as such information are not always accessible for open-source projects. We provided an in-depth similarity analysis of the systems used in Section 4.4.

The main implication of these findings is that, rather than recommending that size should *always* be controlled when building prediction models, we recommend that one should consider controlling for size only when significant evidence of mediation or moderation effects is observed, and that this should be tested for each system individually. Size remains an important metric for predicting defects (and other quality artefacts), but without proper investigation of the mediation and moderation effect of size for individual systems, size should not be omitted or controlled as a rule.

The results presented here provide a complementary perspective on the effect of class size, indicating that there is no strong evidence of a significant effect of size in defect prediction models. We note that the size effect depends on the systems examined. This is also the

case for the moderation effect of size, where the findings regarding such effects are inconsistent across the systems studied. A further investigation of the nature of the systems used reveals that the data collected from different systems exhibit different patterns and therefore it is hard to generalise any results from this analysis. We expect the same outcomes from the previous studies as systems are quite different in nature. It does appear that the selection of datasets (the systems examined) plays an important role in determining whether size has a mediation or moderation effect on the association between OO metrics and defects.

We therefore recommend that if evidence of a significant mediation or moderation effect of size is found (which, as this paper advocates, should be examined using robust statistical methods as explained in Section 3), then one should consider controlling for size before looking at the prediction power of other OO metrics. Statistical procedures to control for the mediation or moderation effects are provided in the literature and should be followed if strong evidence regarding the indirect effect of size is found. There have been a number of methods proposed to control for mediation or moderation effects, including a number of regression-based approaches (Dickinson et al. 2001; Zhou et al. 2014). Stratification methods have also been suggested as an approach to control for mediators and other confounders (McNamee 2005). In such cases (where there is evidence of an indirect effect of size), failing to control for size might result in models that are unstable and misleading in interpreting the prediction power of OO metrics.

Our study fails in finding any strong and consistence evidence of a mediation or moderation effect of size across the systems that we investigated and for most metrics we included. Still, this does not mean that we should completely ignore (or even overestimate) the role of class size in such models. One may argue that size (LOC) can be used solely to predict the likelihood of defects given that size is correlated with the number of defects (Gyimothy et al. 2005; Herraiz and Hassan 2010). However, we believe that predicting defects using size alone can also be misleading. Other metrics should also be incorporated when such models are built. The mediation and moderation analysis can reveal the true impact of size on other metrics. However, class size can be used, with caution, as an initial indicator of the presence of defects in a class. LOC, for example, could be used as a good and quick predictor of defects, as it is generally easier to collect compared to other complexity metrics (Gyimothy et al. 2005).

The potential application of the *mediation* and *moderation* analysis presented in the paper goes beyond defects models, to other areas in software engineering. To the best of our knowledge, the concepts of *mediation* and *moderation* analysis have not been typically studied in software engineering research. The regression-based procedures we used in this study can potentially have a wider application in other software engineering areas such as cost and effort estimation studies, software analytic studies, and social studies in software engineering (for example, in studies where there are human aspects with some potential indirect factors that may impact the outcomes, such as experience or familiarity with a technology).

5 Threats to Validity

This study has a number of validity threats. We discuss possible threats to the validity of our study below.

The selected datasets pose a threat to our results. We selected data from two publicly available datasets. Most of the data collected are mainly from open-source Java-based

projects (mainly Eclipse and Apache projects). However, we sought to select projects that have been used in previous defect prediction studies to allow us to compare our results with previous findings. Both datasets have been widely used in defect prediction studies in the past. In particular, the DAMB dataset has been used in the study of Zhou et al. (2014).

Also, we did not consider factors such as the domain of the program, and timeline of the development when we analysed our results. We believe that such information is important when building predication models.

The method used for determining the mediation and moderation effects of size on defects prediction models can be a potential threat to the validity of the results. There are a number of statistical methods that are available to test for mediation and moderation effects. Previous studies have also utilised different methods to examine the confounding (indirect) effect of size. Those methods may show different conclusion due to their different power and robustness. We decided to use a bootstrapping-based CI methods following the recommendation in the literature (MacKinnon et al. 2004; Hayes 2013), as those methods have been shown to be more robust compared to other available approaches.

6 Conclusion

Previous studies on defect-proneness have shown that class size has an indirect effect on the relationships between OO metrics and defect-proneness. However, no studies have attempted to study the same phenomenon in count defect prediction models. Our study reports on the analysis of the *mediation* and *moderation* effect of class size on the relationships between OO metrics and the number of defects/defect proneness. We investigated the potential indirect effect of size in count defect prediction models and re-examined the evidence on the indirect effect of size on defect-proneness. When gauging indirect effects, bootstrapping-based techniques have several advantages over other methods, including the causal-steps and Sobel tests.

By applying bootstrapping mediation we have shown that there is no conclusive evidence that size has a significant mediation effect on the relationships between most OO metrics and the number of defects, with evidence of a mediation effect of size on the relationship between CBO metric and defects. We also found inconsistent evidence that size has a moderation effect (i.e., affects the magnitude of the relationships between OO metrics and the number of defects), with a significant moderation effect found for RFC and Fan-out metrics. The findings for binary defect prediction models (defect proneness) are not different - our statistical analyses showed that there were inconsistent results regarding the mediation and moderation effects of class size on the relationship between individual OO metrics and defect proneness (again, only the CBO metric shows a significant results for mediation and the Fan-out metric for moderation).

Unlike previous studies (e.g., El Emam et al. 2001; Zhou et al. 2014), our results do not fully support the assertion that the size effect *always exists* in count or binary models. In fact, our results suggest a contrary conclusion - that class size has no consistent significant mediation or moderation effect on the relationships between most OO metrics (all but CBO and Fan-out) and the number of defects or defect-proneness. We contend that bootstrapping-based techniques provide more powerful means for analysing mediator and moderator variables and we therefore recommend that empirical software engineering researchers employ such techniques when studying mediation and moderation effects. The approach we employed here can be applied to many problems in software engineering

studies, and we encourage research to follow the steps we have employed here to study the the potential indirect effect of additional variables in their models.

References

Baron RM, Kenny DA (1986) The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. J Pers Soc Psychol 51(6):1173

Basili VR, Briand LC, Melo WL (1996a) A validation of object-oriented design metrics as quality indicators. IEEE Trans Softw Eng 22(10):751–761

Basili VR, Briand LC, Melo WLWL (1996b) How reuse influences productivity in object-oriented systems. Commun ACM 39(10):104–116

Bennin KE, Keung J, Phannachitta P, Monden A, Mensah S (2018) Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Trans Softw Eng 44(6)

Bennin KE, Keung J, Monden A (2018) On the relative value of data resampling approaches for software defect prediction. Empir Softw Eng 24(2):1–35

Binkley AB, Schach SR (1998) Validation of the coupling dependency metric as a predictor of run-time failures and maintenance measures. In: Proceedings of the 20th International conference on software engineering. IEEE

Bollen K, Stinet R (1990) Direct and indirect effects : classical and bootstrap estimates of variability. Sociol Methodol 20:115–140

Briand LC, Wüst J., Daly JW, Porter DV (2000) Exploring the relationships between design measures and software quality in object-oriented systems. J Syst Softw 51(3):245–273

Catal C, Diri B (2009) Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. Inform Sci 179(8):1040–1058

Chidamber S, Kemerer C (1994) A metrics suite for object oriented design. IEEE Trans Softw Eng 20(6):476–493

D'Ambros M, Lanza M, Robbes R (2012a) Evaluating defect prediction approaches: a benchmark and an extensive comparison. Empir Softw Eng 17(4-5):531–577

D'Ambros M, Lanza M, Robbes R (2012b) Evaluating defect prediction approaches: a benchmark and an extensive comparison. Empir Softw Eng 17(4-5):531–577

Dickinson W, Leon D, Fodgurski A (2001) Finding failures by cluster analysis of execution profiles. In: Proceedings of the 23rd International conference on software engineering. IEEE, pp 339–348

El Emam K, Benlarbi S, Goel N, Rai SN (2001) The confounding effect of class size on the validity of object-oriented metrics. IEEE Trans Softw Eng 27(7):630–650

Evanco WM (2003) Comments on "the confounding effect of class size on the validity of object-oriented metrics". IEEE Trans Softw Eng 29(7):670–672

Fenton NE, Neil M (1999) A critique of software defect prediction models. IEEE Trans Softw Eng 25(5):675–689

Fritz MS, Mackinnon DP (2007) Required sample size to detect the mediated effect. Psychol Sci 18(3):233–239

Ghotra B, McIntosh S, Hassan AE (2015) Revisiting the impact of classification techniques on the performance of defect prediction models. In: Proceedings of the 37th International conference on software engineering. IEEE, pp 789–800

Gil Y, Lalouche G (2017) On the correlation between size and metric validity. Empir Softw Eng 22(5):2585–2611

Gyimothy T, Ferenc R, Siket I (2005) Empirical validation of object-oriented metrics on open source software for fault prediction. IEEE Trans Softw Eng 31(10):897–910

Hall T, Beecham S, Bowes D, Gray D, Counsell S (2011) A systematic review of fault prediction performance in software engineering. IEEE Trans Softw Eng 38(6):1276–1304

Harrison R, Counsell S, Nithi R (1998) Coupling metrics for object-oriented design. In: Proceedings of 5th International software metrics symposium. IEEE

Hassan AE (78) Predicting faults using the complexity of code changes. In: Proceedings of the International conference on software engineering

Hayakawa T, Tsunoda M, Toda K, Nakasai K, Tahir A, Bennin KE, Monden A, Matsumoto K (2021) A novel approach to address external validity issues in fault prediction using bandit algorithms. IEICE Trans Inf Syst 104(2):327–331

Hayes AF (2009) Beyond Baron and Kenny: statistical mediation analysis in the new millennium. Commun Monogr 76(4):408–420

Hayes AF (2013) Introduction to mediation, moderation and conditional process analysis: a regression-based approach. Guilford Press, New York

He Z, Shu F, Yang Y, Li M, Wang Q (2012) An investigation on the feasibility of cross-project defect prediction. Autom Softw Eng 19(2):167–199

Herraiz I, Hassan AE (2010) Beyond lines of code: Do we need more complexity metrics?. Making software: what really works, and why we believe it, pp 125–141

Johnson PE, Grothendieck G (2012) rockchalk: Regression estimation and presentation. R package version 1(2)

Jureczko M, Madeyski L (2010) Towards identifying software project clusters with regard to defect prediction. In: Proceedings of the 6th international conference on predictive models in software engineering. ACM

Kamei Y, Fukushima T, McIntosh S, Yamashita K, Ubayashi N, Hassan AE (2016) Studying just-in-time defect prediction using cross-project models. Empir Softw Eng 21(5):2072–2106

Kamei Y, Shihab E, Adams B, Hassan AE, Mockus A, Sinha A, Ubayashi N (2013) A large-scale empirical study of just-in-time quality assurance. IEEE Trans Softw Eng 39(6):757–773

Kitchenham B, Madeyski L, Budgen D, Keung J, Brereton P, Charters S, Gibbs S, Pohthong A (2016) Robust statistical methods for empirical software engineering. Empir Softw Eng 21(1):212–259

MacKinnon DP, Lockwood CM, Hoffman JM, West SG, Sheets V (2002) A comparison of methods to test mediation and other intervening variable effects. Psychol Methods 7(1):83–104

MacKinnon DP, Lockwood CM, Williams J (2004) Comparison of approaches in estimating interaction and quadratic effects of latent variables. Multivar Behav Res 39(1):37–67

Majumder S, Mody P, Menzies T (2020) Revisiting process versus product metrics: a large scale analysis. arXiv:2008.09569

McNamee R (2005) Regression modelling and other methods to control confounding. Occup Environ Med 62(7):500–506

Mende T., Koschke R (2009) Revisiting the evaluation of defect prediction models. In: Proceedings of the 5th International conference on predictor models in software engineering. ACM

Olague HM, Etzkorn LH, Gholston S, Quattlebaum S (2007) Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes. IEEE Trans Softw Eng 33(6):402–419

Osman H, Ghafari M, Nierstrasz O (2018) The impact of feature selection on predicting the number of bugs. arXiv:1807.04486

Pascarella L, Palomba F, Bacchelli A (2019) Fine-grained just-in-time defect prediction. J Syst Softw 150:22–36

Preacher KJ, Hayes AF (2008) Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. Behav Res Methods 40(3):879–891

Schröter A., Zimmermann T, Zeller A (2006) Predicting component failures at design time. In: Proceedings of the International symposium on empirical software engineering. ACM, pp 18–27

Shepperd M, Bowes D, Hall T (2014) Researcher Bias: The Use of Machine Learning in Software Defect Prediction. IEEE Trans Softw Eng 40(6):603–616

Sobel ME (1982) Asymptotic confidence intervals for indirect effects in structural equation models. Sociol Methodol 13(1982):290–312

Song Q, Jia Z, Shepperd M, Ying S, Liu J (2011) A general software defect-proneness prediction framework. IEEE Trans Softw Eng 37(3):356–370

Tahir A (2021) Does class size matter? online repository and replication package. https://github.com/amjedtahir/size-effect-defect-prediction

Tahir A, Bennin KE, MacDonell SG, Marsland S (2018) Revisiting the size effect in software fault prediction models. In: Proceedings of the 12th International symposium on empirical software engineering and measurement. ACM

Tang M.-H., Kao M.-H., Chen M.-H. (1999) An empirical study on object-oriented metrics. In: Proceedings 6th International software metrics symposium. IEEE

Tantithamthavorn C, Hassan AE (2018) An experience report on defect modelling in practice: Pitfalls and challenges. In: Proceedings of the 40th International conference on software engineering: software engineering in practice. IEEE

Tingley D, Yamamoto T, Hirose K, Keele L, Imai K (2014) Mediation: R package for causal mediation analysis. J Stat Softw 59(5)

Turhan B, Menzies T, Bener AB, Di Stefano J (2009) On the relative value of cross-company and within-company data for defect prediction. Empir Softw Eng 14(5):540–578

Ying AT, Murphy GC, Ng R, Chu-Carroll MC (2004) Predicting source code changes by mining change history. IEEE Trans Softw Eng 30(9)

Zhang H (2009) An investigation of the relationships between lines of code and defects. In: Proceedings of the international conference on software maintenance. IEEE

Zhou Y, Leung H (2006) Empirical analysis of object-oriented design metrics for predicting high and low severity faults. IEEE Trans Softw Eng 32(10):771–789

Zhou Y, Xu B, Leung H, Chen LIN (2014) An in-depth study of the potentially confounding effect of class size in fault prediction. ACM Trans Softw Eng Methodol 23(1)

Zimmermann T, Nagappan N, Zeller A (2008) Predicting bugs from history. In: Software evolution. Springer, pp. 69–88

Zimmermann T, Premraj R, Zeller A (2007) Predicting defects for eclipse. In: Proceedings of the 3rd International workshop on predictor models in software engineering

Amjed Tahir is a Senior Lecturer in Software Engineering at Massey University, New Zealand. He obtained a PhD degree from the University of Otago, NZ, in 2016. Amjed research is mainly in empirical software engineering, with special interest in software maintenance and testing problems. He has served in the organizing and program committee of a several software engineering conferences. Amjed is an Executive member of Software Innovation New Zealand.

Kwabena E. Bennin is an Assistant Professor in the Information Technology Group of Wageningen University and Research. He received his PhD in Computer Science from City University of Hong Kong and his Bachelor degree (Hons) in Computer Science and Statistics from University of Ghana. His research focuses on empirical software engineering, software quality and software analytics.

Xun Xiao is a Lecturer in Statistics at School of Fundamental Sciences, Massey University since 2017. He received his B.Sc. degree in Statistics from the University of Science and Technology of China in 2011 and his Ph.D. degree in Systems Engineering and Engineering Management from the City University of Hong Kong, Hong Kong in 2016. His research focuses on various statistical problems arising from industrial products and systems, including reliability engineering and statistical quality control.

Stephen G. MacDonell is Professor of Software Engineering at the Auckland University of Technology and Professor in Information Science at the University of Otago, both in New Zealand. Stephen was awarded BCom(Hons) and MCom degrees from the University of Otago and a PhD from the University of Cambridge. His research has been published in IEEE Transactions on Software Engineering, ACM Transactions on Software Engineering and Methodology, ACM Computing Surveys, Empirical Software Engineering, Information & Management, the Journal of Systems and Software, Information and Software Technology, and the Project Management Journal, and he has presented his research findings at numerous international conferences. He is a Fellow of IT Professionals NZ, Senior Member of the IEEE and the IEEE Computer Society, Member of the ACM, and he serves on the Editorial Board of Information and Software Technology. He is also Deputy Director of New Zealand?s National Science Challenge Science for Technological Innovation.

Affiliations

Amjed Tahir[1] 🆔 **· Kwabena E. Bennin[2] · Xun Xiao[1] · Stephen G. MacDonell[3,4]**

Kwabena E. Bennin
kwabena.bennin@wur.nl

Xun Xiao
x.xiao@massey.ac.nz

Stephen G. MacDonell
stephen.macdonell@otago.ac.nz

[1]   Massey University, Palmerston North, New Zealand

[2]   Wageningen University and Research, Wageningen, The Netherlands

[3]   Auckland University of Technology, Auckland, New Zealand

[4]   University of Otago, Dunedin, New Zealand