Empir Software Eng (2017) 22:716–767
DOI 10.1007/s10664-016-9446-4

Investigating the use of moving windows to improve software effort prediction: a replicated study

Chris Lokan[1] · **Emilia Mendes**[2,3]

Published online: 26 August 2016
© Springer Science+Business Media New York 2016

Abstract To date most research in software effort estimation has not taken chronology into account when selecting projects for training and validation sets. A chronological split represents the use of a project's starting and completion dates, such that any model that estimates effort for a new project *p* only uses as its training set projects that have been completed prior to *p*'s starting date. A study in 2009 ("S3") investigated the use of chronological split taking into account a project's age. The research question investigated was whether the use of a training set containing only the most recent past projects (a "moving window" of recent projects) would lead to more accurate estimates when compared to using the entire history of past projects completed prior to the starting date of a new project. S3 found that moving windows could improve the accuracy of estimates. The study described herein replicates S3 using three different and independent data sets. Estimation models were built using regression, and accuracy was measured using absolute residuals. The results contradict S3, as they do not show any gain in estimation accuracy when using windows for effort estimation. This is a surprising result: the intuition that recent data should be more helpful than old data for effort estimation is not supported. Several factors, which are discussed in this paper, might have contributed to such contradicting results. Some of our future work entails replicating this work using other datasets, to understand better when using windows is a suitable choice for software companies.

Communicated by: Martin Shepperd

✉ Chris Lokan
c.lokan@adfa.edu.au

Emilia Mendes
emilia.mendes@bth.se; emilia.mendes@oulu.fi

[1] School of Engineering & Information Technology, UNSW Canberra, Canberra, Australia

[2] Faculty of Computing, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden

[3] Faculty of Information Technology and Electrical Engineering, University of Oulu, PO Box 3000, 90014 Oulu, Finland

Keywords Chronological splitting · Effort estimation · Regression-based estimation models · Moving window

1 Introduction

Models for estimating software development effort are commonly built and evaluated using historical data. The usual approach involves separating the data into a training set (from which a model is built) and a validation set (with which the model's accuracy is assessed). An important question is which projects to include in the training set: should it be all available data, or a subset that seems particularly relevant?

Learning from past data is one form of "transfer learning" (Kocaguneli et al. 2014). Using a clustering technique called TEAK, Kocaguneli et al. found out that older project data might sometimes also be applicable to estimate effort for new projects (Kocaguneli et al. 2014). However, despite their results, there are other studies arguing that data set characteristics may change over time; this is a form of "dataset shift", whereby training data may differ from testing data (Turhan 2012), or "concept drift" as it is known in the machine learning literature (Minku and Yao 2012a). Such studies support the argument for disregarding "old" data, and that estimates should be based on how things are done now, not as they used to be done; in other words, data that is not sufficiently "recent" is no longer useful as training data for the purpose of effort estimation.

A study in 2009 (referred to here as "S3") (Lokan and Mendes 2009b) examined this issue by investigating the use of chronological split taking into account a project's age. A chronological split represents the use of a project's starting and completion dates, such that any model that estimates effort for a new project *p* only uses as its training set projects that have been completed prior to *p*'s starting date. This reflects what really occurs in practice.

S3's research question was whether the use of a training set containing only the most recent past projects (i.e. a "moving window" of the *N* most recently-completed projects) would lead to more accurate estimates, compared to using the entire history of past projects completed prior to the starting date of a new project. S3 investigated this issue using several window sizes, with estimates based on models built with stepwise regression, using a data set of 228 single-company projects from the ISBSG repository. The results showed that using a window could be advantageous with some window sizes. While this seems no great surprise, as intuitively it makes sense that "old" data may not be relevant to modern projects and development practices and should therefore be discounted, it was based on only one data set.

Several subsequent studies aimed to extend in different ways the knowledge gained from S3. These studies are described in Section 2. They all investigated the value of windows of different sizes, but they used different estimation methods, different data sets, and/or different windowing policies. Some results are contradictory, so research is needed to understand why these differences arise.

The study detailed herein replicates S3, using moving windows and stepwise regression to build prediction models, with three new data sets. The contribution of this paper is to replicate previous work on the application of moving windows in software effort estimation, in particular with data sets that are more homogeneous than previously studied.

Does this matter, given that much software development now uses agile methods, and most effort estimation is based on expert judgment (Jørgensen 2004)? This replication focuses on the issue of using past data, and algorithmic estimation (using regression analysis of past data) to predict project effort. It may be questioned as to whether: i) effort estimation is still important in practice; and ii) the use of an algorithmic technique for effort forecasting is useful in current practice.

Addressing the first point, despite existing evidence showing that judgment-based effort estimation is the technique most used in practice (Jørgensen 2004), this does not rule out the importance that effort prediction has within the context of project management. There have been several systematic literature reviews on the topic of effort estimation (e.g. effort estimation in general (Jørgensen and Shepperd 2007), Web effort estimation (Azhar et al. 2012), effort estimation in agile software development (Usman et al. 2014), effort estimation in global software development (Britto et al. 2014)); and also other studies, such as a survey with agile practitioners (Britto et al. 2015), case studies with Web development companies (Mendes 2014), and papers discussing ways to improve subjective effort estimation in practice (Jørgensen 2004, 2005, 2013; Jørgensen and Grimstad 2008). Further, there are numerous books on the topic of estimation, including estimation within the context of agile projects (e.g. (Cohn 2005)). These bodies of evidence suggest quite clearly that this topic is still relevant to both research and practice.

With regard to the second point, there have been several studies providing evidence of the use of more formal approaches than expert judgment to effort estimation within organizations (e.g. COCOMO); some in the relatively recent fields of web development (Azhar et al. 2012; Mendes 2014) and agile development (Britto et al. 2015; Schmietendorf et al. 2008). None of these studies has employed the same approach detailed herein, i.e., regression analysis and windows; however this does not mean that it could not be employed in practice, in particular if we also add tool support.

1.1 Research Questions

We address the same research questions investigated in S3, as follows:

RQ1: Assuming a chronological split approach to effort estimation, which takes into account a project's age, is there a difference between the accuracy of estimates using prediction models that are built using all available data in a training set, and the accuracy of estimates using prediction models that are built using only the *N* most recently-completed projects in the training set?

This research question is characterized by the following null hypothesis:

H0: There are no differences in the accuracy of effort estimates obtained using models built using a window of recent training data and effort estimates obtained using models built using all the available data.

Herein the treatment is the use of a window, and the control is the use of all available data in a training set completed prior to the new project's starting date.

RQ2: Can insights be gained by observing trends in estimation accuracy as *N* varies?

The first research question is addressed quantitatively, using a non-parametric paired-samples statistical hypothesis test – the Wilcoxon signed-rank test – where absolute residuals for both treatment and control are checked for statistically significant differences.

The second question is addressed observationally, by noting trends in average estimation accuracy as window sizes vary, and statistically, using correlations between average estimation accuracy and window size.

1.2 Relevance in Practice

Whether or not to consider using windows is only relevant if the software industry is actually developing and using predictive models based on historical data.

The original paper that is replicated herein used company-specific data from the ISBSG database[1]. This large database contains project data contributed by organizations worldwide, after they had gathered the data for their own use. We cannot state certainly that they collected effort data to use with algorithmic/other models for estimation purposes; however it is extremely likely. The ISBSG data is used by commercial companies that provide project management tools for estimating effort[2], and we know of organizations that use the ISBSG data as a resource for effort estimation. The Finnish dataset used in this paper also underpins commercial software estimation tools[3].

What these suggest is that some organizations want to make their estimates based on a more formal approach, using historical data, rather than relying only on subjective expertise. Also, there are several commercial tools that provide effort estimates. Again we cannot tell what sort of data they employ; however they all provide mechanisms that focus on formalizing the estimation process.

Regarding whether windows are used in industry: one of the authors of this paper has first-hand experience collaborating with companies in New Zealand and Brazil, building hybrid Web effort estimation models for these companies (Mendes 2014). These companies provided data on past projects as well as expert knowledge to build estimation models using Bayesian Networks. Some of these companies explicitly did not want to use past data beyond a certain number of years (a window approach) to build their model.

1.3 Organization of the Paper

The remainder of the paper is organized as follows: Section 2 summarizes related work; Section 3 presents background information about this replication by first summarizing the original study (S3) and then describing how this replication relates to S3. Section 4 describes the research method employed herein, followed by the presentation of our results in Section 5. Section 6 discusses the results from two perspectives: the first relates to the particular research questions within the context of the data employed herein; the second relates to the nature of this study as a replication of S3. Finally, Section 7 presents threats to validity, and our conclusions and directions for future work are given in Section 8.

[1] http://www.isbsg.org

[2] http://isbsg.org/project-estimation-tools/

[3] http://www.4sumpartners.com/

2 Related Work

To date about 30 studies relating to software effort estimation have considered the chronological sequence of projects in their research. Around half of these studies (Group 1) did not have research questions relating to chronology, but chose chronological order as the basis for forming training and validation sets; the other half (Group 2) investigated chronology as a research question in its own right. Studies in Group 1 are briefly presented next, followed by a more detailed description of Group 2 studies.

Group 1:

The first research we are aware of that considered the use of moving windows was by Kitchenham et al. (2002). They found that when they divided their data into subsets by start date, the regression models changed between the subsets. As a result they argued that old projects should be removed from the data set when new ones were added, so that the size of the data set remained constant. They recommended that the estimate for project *n* should be based on projects *n*–30 to *n*–1: a moving window of 30 projects.

Some studies considered projects up to a given dividing point as the training set, and projects after that point as the validation set (Bibi et al. 2008, 2010; Lefley and Shepperd 2003; Li et al. 2009; Lopez-Martin et al. 2012; Schmietendorf et al. 2008). Others treated projects as a data stream, arriving one by one in chronological order (Minku and Yao 2012b; Song et al. 2013). Chronology is inherent in studies relating to changing productivity over the years in software development projects (Fernández-Diego et al. 2010; Premraj et al. 2005), and studies into the use of recorded effort from earlier stages in a software project to estimate effort in later stages of the same project (Azzeh et al. 2010; MacDonell and Shepperd 2003, 2010). The apparent accuracy of effort estimates when evaluated using cross-validation, instead of treating projects as a data stream, has also been investigated (Lokan and Mendes 2008, 2009a; Sigweni et al. 2016).

Group 2:

We know of 15 studies to date that investigated the use of chronology in its own right. They are summarized below; details are in Table 5 in the Appendix.

These studies used a range of approaches to represent timing information:

- Fixed size moving window (Amasaki and Lokan 2012, 2013, 2014b, 2014c, 2016a, b, 2015; Amasaki et al. 2011; Lokan and Mendes 2009b, 2014, 2012; Tsunoda et al. 2013): given a project *p* for which effort is to be estimated, all the projects in the training set must have been completed prior to *p*'s starting date AND out of those, only the *n* most recently completed projects are used.
- Fixed duration moving window (Amasaki and Lokan 2014c, 2015, 2016a; Lokan and Mendes 2012, 2014): given a project *p* for which effort is to be estimated, all the projects in the training set must have been initiated and completed within the last *m* months from *p*'s starting date.
- Project-by-project split (growing portfolio) (Amasaki and Lokan 2012, 2014a, b, c, 2015, 2016a, b; Amasaki et al. 2011; Lokan and Mendes 2008, 2009b, 2012, 2014, 2009): given a project *p* for which effort is to be estimated, all the projects in the training set must have been completed prior to *p*'s starting date, AND all of those projects are considered. This procedure is repeated individually for all projects in the dataset.

- Date-based selection (Lokan and Mendes 2009a): a date *d* is chosen and used to reduce the training set to only include projects that were completed prior to *d*, and to reduce the validation set to only include projects that were initiated on or after *d*.
- Dummy variable of moving windows (Tsunoda et al. 2013): a dummy variable is also created, and is assigned the value one if a project finished recently; otherwise it is assigned zero. Within such a scenario, each effort estimation model built can be trained using all the data points in a dataset.
- Dummy variables of Year (Tsunoda et al. 2013): Dummy variables are created for each of the starting years for the projects in a dataset. Then all the projects that are starting in a given year have the value of their corresponding dummy variable set to one, and so forth. For example, the dummy variable named 2011 will be set to one for all the projects that have started in 2011.
- Dummy variables of Equal Bins (Tsunoda et al. 2013): Dummy variables are created representing a time span, where each time span contains the same number of projects. For example, a bin of size 3 means that each time span associated with a dummy variable will contain three data points.
- Year predictor (Tsunoda et al. 2013): Uses the starting year of projects as an independent variable that represents the timing information. The assumption here is that there is a relationship between effort and project starting year.
- Serial number predictor (Tsunoda et al. 2013): Creates an independent variable that represents the difference between a project's starting date and a base date. For example, if the base date is April 1, 2010, and the start date of a project is April 2, 2010, the independent variable holds 1.
- Weighting projects by age (Amasaki and Lokan 2013, 2014a, c, 2016b): varies the original fixed size moving windows, by giving recent projects more importance than older projects. Four strategies are considered:

- Unweighted growing: all past projects are retained, all projects have the same non-zero weight.
- Unweighted window: old projects that no longer fit within the window have a weight of zero, and all projects in the window have the same non-zero weight.
- Weighted growing: all past projects are retained, no project has a zero weight, and projects have different weight according to their age relative to the target project.
- Weighted window: projects outside the window have zero weight, projects within the window have non-zero weight, and projects are weighted differently within the window according to their age relative to the target project.

Related studies investigating chronology in its own right:

Lokan and Mendes (2008) ("S1") and Mendes and Lokan (2009) ("S2") were the first studies to investigate chronology in its own right. They used as a benchmark leave-one-out cross-validation, and used as chronology choices a project-by-project split, and a date-based selection, respectively. Both studies used a single-company dataset from the ISBSG repository. In both studies, cross-validation estimates showed significantly superior accuracy. Sigweni et al. recently found the same using a different data set (Sigweni et al. 2016).

Lokan and Mendes (2009b) (“S3”) (described in detail in the next section) was the first study to investigate moving windows directly. S3 studied windows containing fixed numbers of projects, as suggested by Kitchenham et al. (2002). The results showed that: i) windows containing up to 23 projects were detrimental to estimation accuracy, compared to retaining all training data, although there were few window sizes at which the difference was statistically significant; ii) windows containing 85 or more projects showed significantly better accuracy, in terms of mean magnitude of relative error (MMRE), though not in terms of mean absolute error (MAE). The reduction in MMRE was approximately 15 %.

Amasaki and Lokan (2015) (“S4”) also investigated different-sized moving windows. They used a different estimation technique (estimation by analogy, instead of regression), and studied two different data sets (both sourced from the PROMISE repository (Menzies et al. 2016): one was that used by Kitchenham et al. (2002); the other from Maxwell (Maxwell 2002)). They found that using windows improved the average values of accuracy statistics, though the improvements were not statistically significant.

Amasaki and Lokan (2012) (“S5”) investigated moving windows using both regression and estimation by analogy, on the data set used in S3. They found ranges of window sizes for which it was significantly better to use a window, with both regression and estimation by analogy. The effect of using a window was stronger with regression. Some differences in research method meant that the results could not be compared directly with S3 (because an extra independent variable was considered in S5) or S4 (because more neighbors and more combinations of potential independent variables were considered in S5).

Lokan and Mendes (2012) (“S6”) redefined windows to represent fixed time spans (e.g. projects that are up to 1 year old, up to 2 years old, etc.) rather than fixed numbers of past projects. They used the same data set as S3 but (as in S5) added an extra independent variable. They found that windows covering short time spans were detrimental to accuracy, but windows of two to three years improved accuracy significantly in terms of MMRE, and windows of three to four years improved accuracy significantly in terms of both MMRE (which was reduced by about 7 %) and MAE (which was reduced by about 4 %). Later they replicated S4 using a different data set (one of those analyzed herein), obtaining contradictory results (Lokan and Mendes 2014).

Tsunoda et al. (2013) (“S7”) compared the prediction accuracy between six different methods for treating timing information, based on linear regression and data from three different datasets (ISBSG, Maxwell and Kitchenham). None of the six different methods presented superior accuracy when based on both Maxwell and Kitchenham datasets, however the dummy variable of moving windows and the moving windows presented superior accuracy when using the ISBSG dataset, for a small and large datasets, respectively.

Further, Amasaki and Lokan (2013, 2014a b, 2016a) (“S8”, “S9”,”S10”, “S14”) used the ISBSG dataset to investigate several types of moving windows (weighted fixed size, fixed size, weighted fixed duration, fixed duration), using either linear regression or classification and regression trees. Results were overall quite promising when using windows (S8: superior accuracy for all window sizes; S9: superior results for windows containing 40 to 60 projects; S10: superior results for durations of 30 months, and 49 to 84 months; S14: superior results for windows containing around 40 to 60 projects).

Lokan and Mendes (2014) ("S11") extended S6 using an additional dataset, and also investigated the effect on estimation accuracy when using moving windows of various durations to form training sets on which to base effort estimates. Results showed that neither fixed size nor fixed duration windows provided superior estimation accuracy in the new dataset, thus suggesting that it is not always beneficial to exclude old data when estimating effort for new projects. When windows are helpful, windows based on duration are effective.

Finally, Amasaki and Lokan (2014c, 2015, 2016b) ("S12","S13", "S15") have recently investigated the use of gradual weighting, in which moving windows were used and the projects within the windows had different weights: more weight was given to recent projects in the window and less to older projects in the window. They found that different weighting functions affect estimation accuracy differently, weighted moving windows are significantly advantageous in larger windows, and non-weighted moving windows are significantly advantageous with smaller windows.

To summarize the data sets and estimation methods used:

- Data sets have been drawn from ISBSG (Amasaki and Lokan 2012, 2013, 2014a, b, c, 2016a; Lokan and Mendes 2008, 2009b, 2012, 2014; Mendes and Lokan 2009; Tsunoda et al. 2013); the Finnish data set (Amasaki and Lokan 2015, 2016b; Lokan and Mendes 2014), Kitchenham (Amasaki and Lokan 2015; Tsunoda et al. 2013), and Maxwell (Amasaki and Lokan 2015; Tsunoda et al. 2013).
- Chronology has been investigated in conjunction with linear regression (Amasaki and Lokan 2012, 2013, 2014b, c, 2016b; Lokan and Mendes 2008, 2009b, 2012, 2014; MacDonell and Shepperd 2010; Mendes and Lokan 2009; Tsunoda et al. 2013), Lasso (Amasaki and Lokan 2013, 2014a, b), estimation by analogy (Amasaki and Lokan 2012, 2015), and CART (Amasaki and Lokan 2014b, 2016a). Benefits from using windows as the chronology approach seem slightly stronger with regression and Lasso.

In most instances, research so far has shown that using windows to eliminate "old" data can improve the accuracy of effort estimates. This has been true with some different data sets, estimation methods, and windowing policies, and we believe it is what many practitioners and researchers take for granted. However, there are some studies that found otherwise. This paper replicates a previous study (S3), using one data set that has previously been studied (though using weighted windows and fixed duration windows, rather than the approach of S3) and two that have not previously been studied, in order to investigate this issue further. It builds on our previous work in the area by extending the range of data sets that have been studied, in particular considering data sets that are homogeneous in terms of business sector.

3 Background to this Replication

Following the guidelines proposed by Carver for reporting replications (Carver 2010), this section describes the original study (S3) and its results, the aims of this replication, and the changes between the original study and this replication.

3.1 The Original Study

As stated above, S3's goal was to investigate whether the use of a training set containing only the *N* most recently-completed past projects would lead to more accurate predictions than using the entire history of past projects completed prior to the starting date of a new project.

S3 used a data set of 228 projects from a single organization (sourced from the International Software Benchmarking Standards Group (ISBSG) Database Release 10). The time span of the projects varied from June 1994 (earliest start) to March 2003 (latest finish). Projects varied in type (new development, enhancement), language type (3GL, 4GL), platform (mainframe, midrange, PC, multi-platform), and industry sector. However, S3 did not consider industry sector as an independent variable. Over time, projects shifted from mainly new developments to mainly enhancements. No other noticeable shifts were identified.

A chronological split approach, taking into account a project's age, was used to estimate the effort for each project. Projects were considered in chronological order; a separate estimation model was built for each project; and only projects that had already finished were used as training data. All models were built using an automated process (backward stepwise multiple regression), programmed in the statistical language R.

The accuracy measures that were used in S3 to compare the effort models were the mean magnitude of relative error and mean absolute residuals. Differences in accuracy were assessed by considering the set of projects whose estimate could be influenced by the use of a window. As the window size increases, the set of evaluation projects becomes smaller.

Note that details on the research method employed in S3 and herein are given in Section 4.

The main results from S3 were the following:

- For each window size (*N*) from 20 projects to 120 projects, accuracy was compared between estimates that were based on a window of the last *N* completed projects, and estimates that were based on the entire set of projects completed so far.
- Average accuracy statistics were significantly worse when using small window sizes (up to 23 projects), compared to retaining all training data.
- Average accuracy statistics were superior (however not statistically significant) when using a window size of 53 to 66 projects, compared to retaining all training data.
- Whenever employing window sizes ranging from 67 to 85 projects, some sizes presented significant improvement in accuracy in favor of using a window. With a window of 85 projects or more, MMRE was significantly better than when the window was used.
- Overall, accuracy was best across the entire data set (including the projects for which the window did not make a difference) when a window of about 75 projects was used.
- Based on the data set employed, the overall trend was that small windows clearly reduce estimation accuracy and larger windows help accuracy.
- The advantage of using a window was not significant in terms of absolute residuals; however it became significant when based on MMRE and large windows.

3.2 This Replication

The motivation to carry out this replication is to investigate further whether the use of a training set containing only recent projects would lead to more accurate predictions, when compared to using the entire history of past projects. We believe this is an important issue to investigate, as we believe that the use of a window represents more closely what occurs in practice.

In particular, the aim is to broaden the results obtained in S3 by investigating the same research question and method with other independent data sets. If the results are consistent, this may help in generalizing findings to a wider population of software organizations. If the results are inconsistent, insight may be gained by considering how differences in the nature of the data may relate to differences in the results.

In regard to the level of interaction with the original experimenters, the same researchers who conducted S3 are the ones carrying out this replication. This is therefore not an independent replication, in the sense of someone else replicating researchers' work in order to confirm the results of the original study. Instead, it involves the same researchers replicating the original research question and method as closely as possible but with different data, in order to gain more insight into the original results; another form of "replication" (Mäntylä et al. 2010; Shull et al. 2008).

This study retains the definition of a window as containing a fixed number of projects, rather than covering a fixed time span, in order to keep the design as close as possible to S3.

The changes made between the original experiment S3 and this experiment are the following:

- There are some different characteristics between the data set used in S3 and the three data sets employed in this study, which are:

- The number of projects, which is similar to or smaller than the number of projects in S3;
- Homogeneity with respect to industry sector. The datasets used in this replication were each dominated by a single sector (two from insurance, and one from public administration), whereas the data set used in S3 contained projects from several sectors (insurance, manufacturing, banking, and service industries were most common);
- The projects' age span. The three data sets employed herein cover longer time spans, starting earlier and finishing later (1988 to 2007, 1991 to 2007, and 1982 to 2007), than the data set used in S3 (1994 to 2003);
- The three organizations studied here are from Finland; the organization studied in S3 is not.

- S3 used the pairwise *t*-test and pairwise Wilcoxon signed-rank test to assess the statistical significance between predictions; however we only employ the pairwise Wilcoxon signed-rank test herein, as our samples are at times quite small.
- Comparisons are added with baseline models: the mean model, whereby the effort estimate is the mean of the effort values of the projects in the training set; and the median model, whereby the effort estimate is the median of the effort values of the projects in the training set.
- Measures such as MMRE, median MRE (MdMRE) and Pred(*l*) have been often used in previous studies (including S3) to evaluate the accuracy of effort estimation models. However, these measures have been criticized for the likelihood of introducing bias. For example, Kitchenham et al. (2001) showed that MMRE and Pred(*l*) are respectively measures of the

spread and kurtosis of z, where ($z=\hat{e} / e$), $\hat{e}$ is the estimate and e is the actual effort. They suggested the use of boxplots of z and boxplots of the residuals ($\hat{e} - e$) as useful alternatives to simple summary measures, since those can give a good indication of the distribution of residuals, and z and can help explain summary statistics such as MMRE and Pred(25). Later, Shepperd and MacDonell (Shepperd and MacDonell 2012) argued for comparisons based on MRE to be deprecated and to use instead the Mean Absolute Error (MAE) for the evaluation and comparison of prediction techniques, because it is unbiased towards over- or under-estimations. Therefore, herein we use MAE alone and no longer use MMRE; by doing so we change slightly the experimental method between S3 and this replication.
- Holm-Bonferroni corrections are used herein, when multiple tests for statistical significance are made for each organization with different window sizes.

To summarize: this study aims to replicate S3, aiming to gain broader insight into the effect on estimation accuracy of using windows of recent projects as training data. It does so by investigating the same research questions and using the same underlying experimental design and method as in S3, employing three independent single-company data sets (smaller and more homogeneous than the one used in S3). To strengthen the experimental method, comparisons with baseline models are added, MMRE is no longer used, and the Holm-Bonferroni correction is applied to families of statistical significance tests.

4 Research Method

4.1 Formation of Data Sets

4.1.1 Selection of Projects

The data sets used herein were sourced from the Finnish data set (as of May 2008). This data set contains data for 846 projects, all from Finnish IT companies. They were completed between 1978 and 2007, and represent a wide range of the IT industry.

To form a data set suitable for our analysis (high quality data, with comparable definitions for size and effort), we removed projects according to the following criteria:

- Remove projects if they were assigned a low data quality rating.
- Remove duplicate projects.
- Remove projects if their size is measured in COSMIC, rather than FiSMA FPs. (Most projects had their size measured using FiSMA FPs, hence we removed those measured using COSMIC. Note that it is incorrect to include in the same analysis projects that were measured using different function point sizing methods, as the way they measure function points differ.)

This left 794 software projects. Within these 794 projects, there were three substantial single-company data sets. These three data sets are analyzed in this paper.

We label them Organizations A, B, and C. Rules of confidentiality mean that no company's identity is known to us.

Organization A had data for 201 projects. All but one were from the same industry sector (Insurance); we removed the one project that was not from the Insurance sector, leaving 200 projects. Organization B had data for 103 projects; 95 were from the Public Administration sector, and 8 from other sectors. We decided to keep all 103 projects, as the minority sectors still represent 7 % of the dataset size. Organization C had data for 95 projects, entirely from one sector (Insurance).

Each data set is described in detail later in this section. Summary statistics for size (measured in FiSMA function points (FP)), effort (staff-hours), and project delivery rate ("PDR", hours per FP, calculated as effort divided by size: high PDR values indicate low productivity) are presented, and the nominal variables are characterized.

To look broadly at how project characteristics changed over time, we divided them chronologically by start date into groups containing equal numbers of projects. S3 did the same, using three groups split by start date. Here we investigated varying numbers of groups, from two to ten.

With few groups, the time span covered by each group can be too long to identify changes in the data. As the number of groups increases, the number of projects per group decreases, and the time spanned by the groups decreases. A point is reached at which time spans are probably too short to represent the rate of change in an organization's projects and practices (for example, with 10 groups the time spanned by most groups is well under one year).

In each organization, data was sparse up to 1998, but accumulated steadily from 1999 to 2007.

For each number of groups, we summarized the nominal variables (how often each value occurred) and ratio-scaled variables (range, quartiles, mean, standard deviation) for the projects in each group, and identified trends in the results. From our investigation we judged that 2 to 4 groups was too few, 5 to 7 was reasonable, and 8 or more was too many. Dividing the projects chronologically into five groups provided the best balance of group size, time span, and exposure of trends. Each group contained at least 19 projects, and all (except for the first group from each organization) spanned a period of one to three years.

4.1.2 Selection of Variables

The Finnish repository provides data on many variables. The fundamental variables are size, effort, and four basic project classifiers: development type, hardware platform, development language, and business sector. Other variables include several situation analysis variables, analogous to general system characteristics in IFPUG function points, or to COCOMO cost drivers.

In this study we restrict our attention to five variables (see Table 1), which are the same five variables used in S3 despite their ISBSG definitions and the Finnish definitions not being exactly the same:

- S3 measured size in IFPUG function points, while the Finnish data set measures size using the FiSMA definition of function points.

Table 1 Variables used from the Finnish repository

Variable	Scale	Description
Effort	Ratio	Project effort in person-hours
Size	Ratio	Application size in FiSMA function points
LangType	Nominal	Language type: 3GL, 4GL, application generator
DevType	Nominal	Development type: New development, enhancement (including integration and conversion), maintenance, other
Platform	Nominal	The type of hardware the system was developed for: mainframe, midrange, multi-platform, PC-network, standalone PC, other

- ISBSG defines development type as new development, enhancement, or re-development; the Finnish data set defines it as new development, enhancement, maintenance or other.
- ISBSG defines the development platform as mainframe, midrange, multi-platform, or PC; the Finnish data set defines it as mainframe, midrange, multi-platform, PC-network, PC-standalone, or other.

Possible threats to validity due to these differences are discussed in Section 6.2.

The definition of Effort is identical for both ISBSG and FiSMA. Further, we also ensured that the definition of development language type is identical for both ISBSG and FiSMA (S3 used ISBSG's Language type variable to represent the programming language; we used ISBSG's classification of languages to define the Language Type variable that is used herein).

We did not consider business sector as an independent variable, because it was not considered in S3, and it scarcely varies in any of the three Finnish data sets.

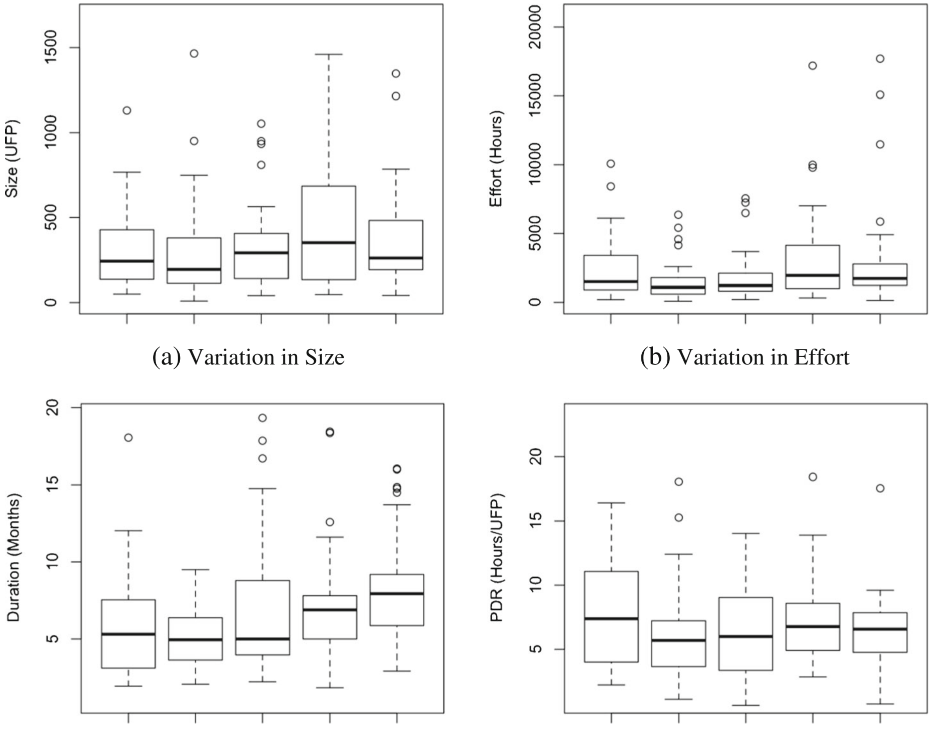
4.2 Description of Data Sets

4.2.1 Organization A

From the earliest start date to the latest finish date, Organization A's projects span almost 20 years, from January 1988 to November 2007. Data is sparse to begin with (there are only 8 projects from the first 10 years), but thereafter it arrives steadily. By start date, the first 20 % of projects span 12 years; the next four groups of 20 % of projects in chronological order span about 1.5, 1.5, 1.5, and 3 years respectively.

Table 2 Organization A: Summary statistics for ratio-scaled variables

Variable	Mean	Median	StDev	Min	Max
Size (FP)	380	264	402	9	3375
Effort (Hours)	2533	1471	3858	86	41,640
Duration (Months)	6.8	5.9	4.5	1.9	39.0
PDR (Hours/FP)	7.01	6.33	4.61	0.72	46.47



(a) Variation in Size

(b) Variation in Effort

(c) Variation in Duration

(d) Variation in PDR

Fig. 1 Variation over time in ratio-scaled variables: Organization A

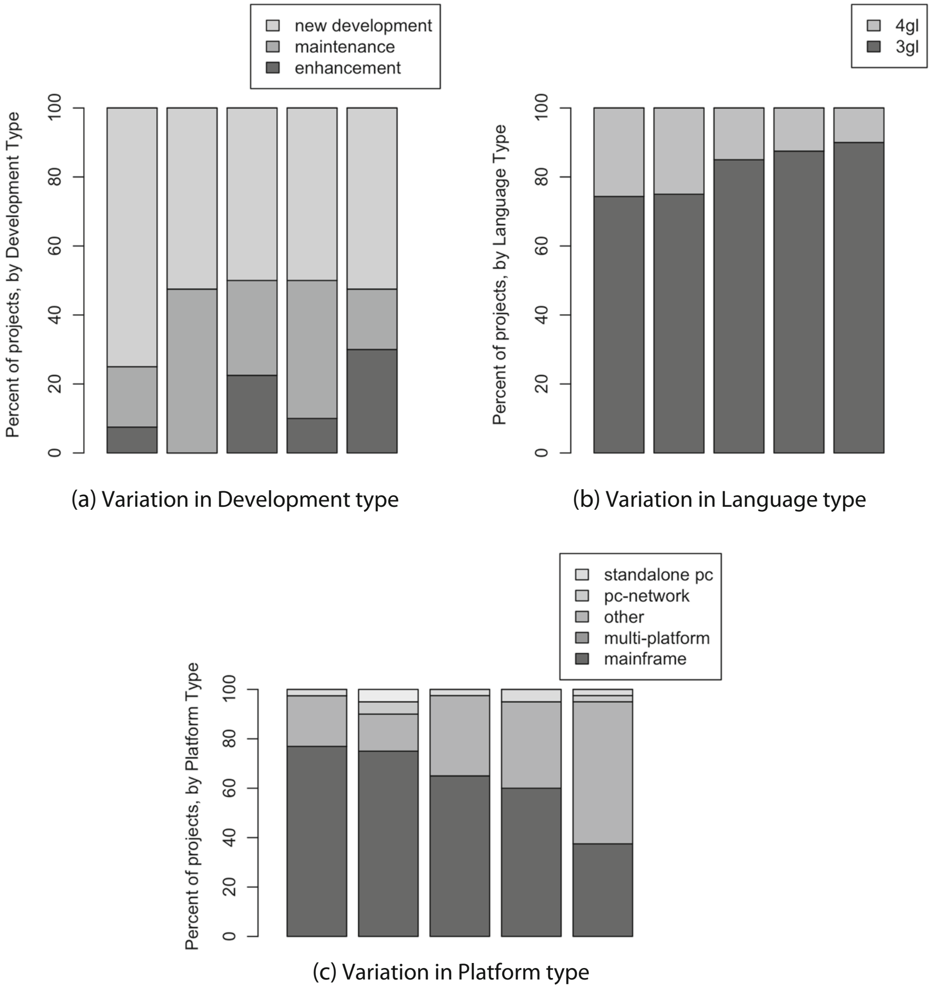
Table 2 summarizes the ratio-scaled variables for the 200 projects from Organization A. Figure 1 shows how they vary broadly over time (the five boxplots in each sub-figure represent the first, second, third, fourth, and fifth sets respectively, when the 200 projects are divided in order of start date into five groups of 40 projects each). Figure 2 shows how the development type, language type, and platform type vary broadly over time.

Summarizing the nominal variables for Organization A:

- Language type: 82 % are 3rd generation language (3GL) projects, 18 % are 4th generation language (4GL) projects.
- Development type: 56 % are new developments, 44 % are maintenance or enhancement projects.
- Platform: 62 % are mainframe projects, 32 % are multi-platform projects.
- Sector: all are from the insurance sector.

Figures 1 and 2 show some variation over time in project characteristics:

- Although there is no consistent trend in Size (Fig. 1a) or Effort (Fig. 1b), both are generally highest in the fourth group of projects.
- Duration is generally lower in the earlier projects than in the later projects (Fig. 1c).
- PDR is generally higher in the first fifth of projects; thereafter there is no consistent trend (Fig. 1d).



(a) Variation in Development type

(b) Variation in Language type

(c) Variation in Platform type

Fig. 2 Variation over time in nominal variables: Organization A

- New developments constitute 75 % of the first fifth of projects; thereafter they are steady at about 50 % (Fig. 2a).
- There is a continual trend away from 4GLs towards 3GLs (Fig. 2b).
- There is a continual trend away from Mainframe platforms towards multi-platform environments (Fig. 2c).

4.2.2 Organization B

From the earliest start date to the latest finish date, Organization B's projects span 16.5 years, from June 1991 to December 2007. As with Organization A, data is sparse to begin with. By start date, the first 20 % of projects span 7 years; the next four groups of 20 % of projects in chronological order span about 2.5, 1, 2, and 3 years respectively.

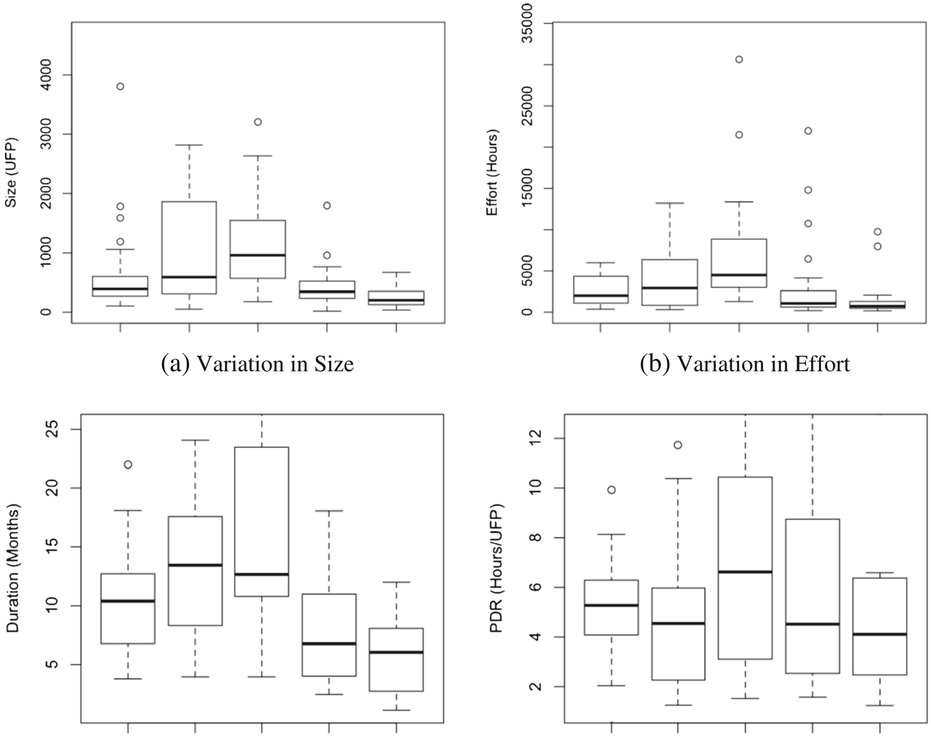
Table 3 Organization B: Summary statistics for ratio-scaled variables

Variable	Mean	Median	StDev	Min	Max
Size (FP)	859	433	1314	18	9390
Effort (Hours)	4773	2028	8700	179	67,580
Duration (Months)	12.0	10.4	9.6	1.1	50.6
PDR (Hours/FP)	5.99	4.81	4.46	0.42	25.00

Table 3 summarizes the ratio-scaled variables for the 103 projects from Organization B. Figures 3 and 4 show how the project characteristics vary broadly over time (projects are divided chronologically by start date into five groups, four of 20 projects and the last of 23 projects).

Summarizing the nominal variables for Organization B:

- Language type: 66 % are 3GL projects, 34 % are 4GL projects.
- Development type: 76 % are new developments, 19 % are maintenance or enhancement projects.
- Platform: 14 % are mainframe projects, 76 % are multi-platform projects.
- Sector: 93 % are Public Administration projects.



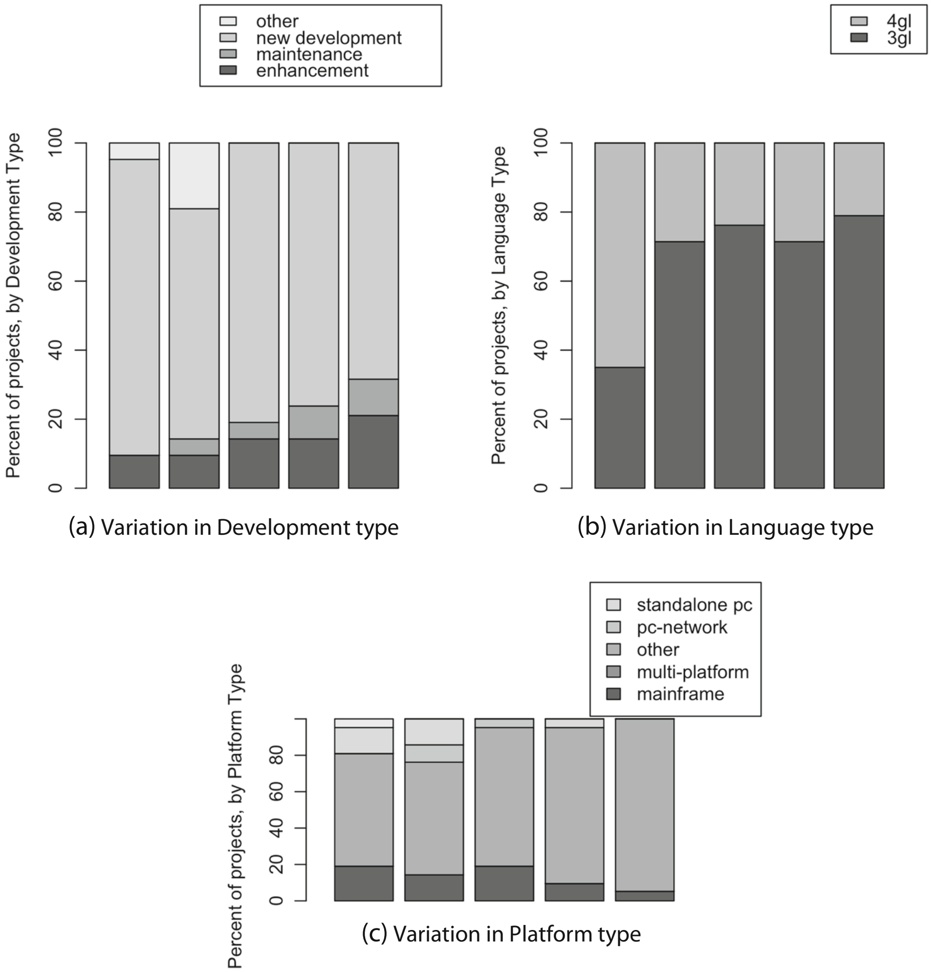
(a) Variation in Size

(b) Variation in Effort

(c) Variation in Duration

(d) Variation in PDR

Fig. 3 Variation over time in ratio-scaled variables: Organization B



(a) Variation in Development type

(b) Variation in Language type

(c) Variation in Platform type

Fig. 4 Variation over time in nominal variables: Organization B

Figures 3 and 4 again show some variation over time in project characteristics:

- Size (Fig. 3a), Effort (Fig. 3b) and Duration (Fig. 3c) all increase through the first 60 % of the projects, then drop to their lowest values in the last 40 % of the projects, which are the most recent in this data set. This could be explained by the increase in the number of enhancement and maintenance-type of development projects.
- There is a general decline in PDR, except for a spike in the middle 20 % of projects.
- PDR is generally higher in the first fifth of projects; thereafter there is no consistent trend (Fig. 3d).
- There is a steady increase in maintenance and enhancement projects, at the expense of new developments (Fig. 4a).
- 3GLs take over from 4GLs after the first 20 % of projects (Fig. 4b).
- There is a steady trend away from both PC and Mainframe platforms towards multi-platform environments (Fig. 4c).

Table 4 Organization C: Summary statistics for ratio-scaled variables

Variable	Mean	Median	StDev	Min	Max
Size (FP)	220	94	417	6	2667
Effort (Hours)	2015	840	3627	42	21,800
Duration (Months)	9.5	5.5	15.1	0.8	105.1
PDR (Hours/FP)	9.93	8.95	5.61	0.38	47.85

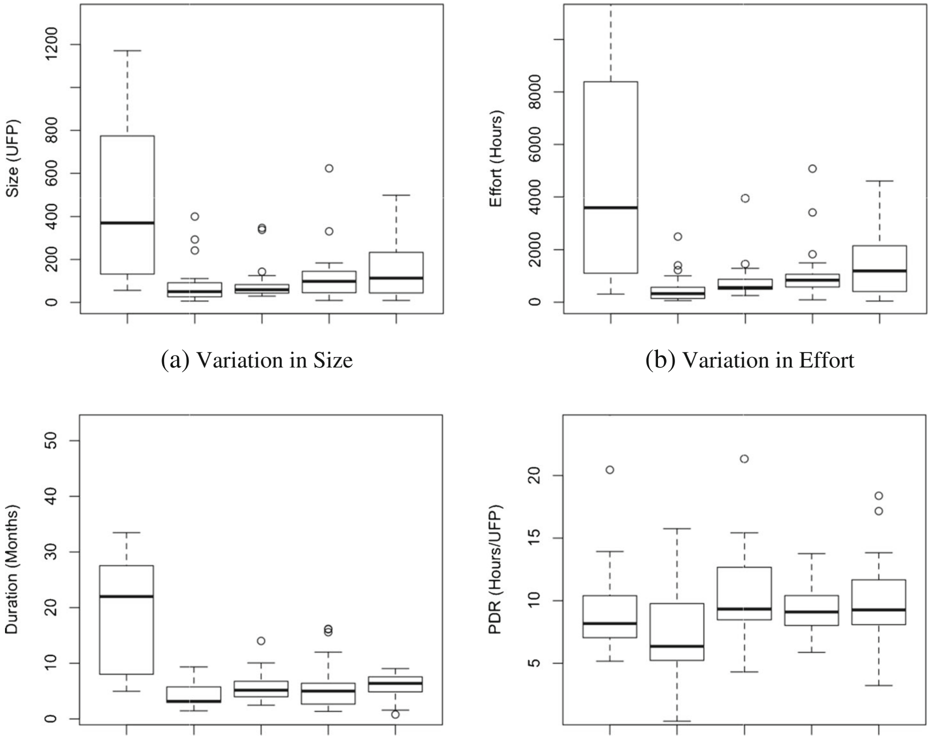
4.2.3 Organization C

From the earliest start date to the last finish date, Organization C's projects span 25.5 years, from January 1982 to June 2007. Again the data is sparse to begin with (there are only 10 projects from the first 18 years). By start date, the first 20 % of projects span nearly 20 years; the next four groups of 20 % of projects in chronological order span about 2.5, 1, 1, and 1 years respectively.

Table 4 summarizes the ratio-scaled variables for the 95 projects from Organization C. Figures 5 and 6 show how the project characteristics vary broadly over time (projects are divided chronologically by start date into five groups of 19 projects each).

Summarizing the nominal variables for Organization C:

- Language type: 81 % are 3GL projects, 17 % are application generator projects.



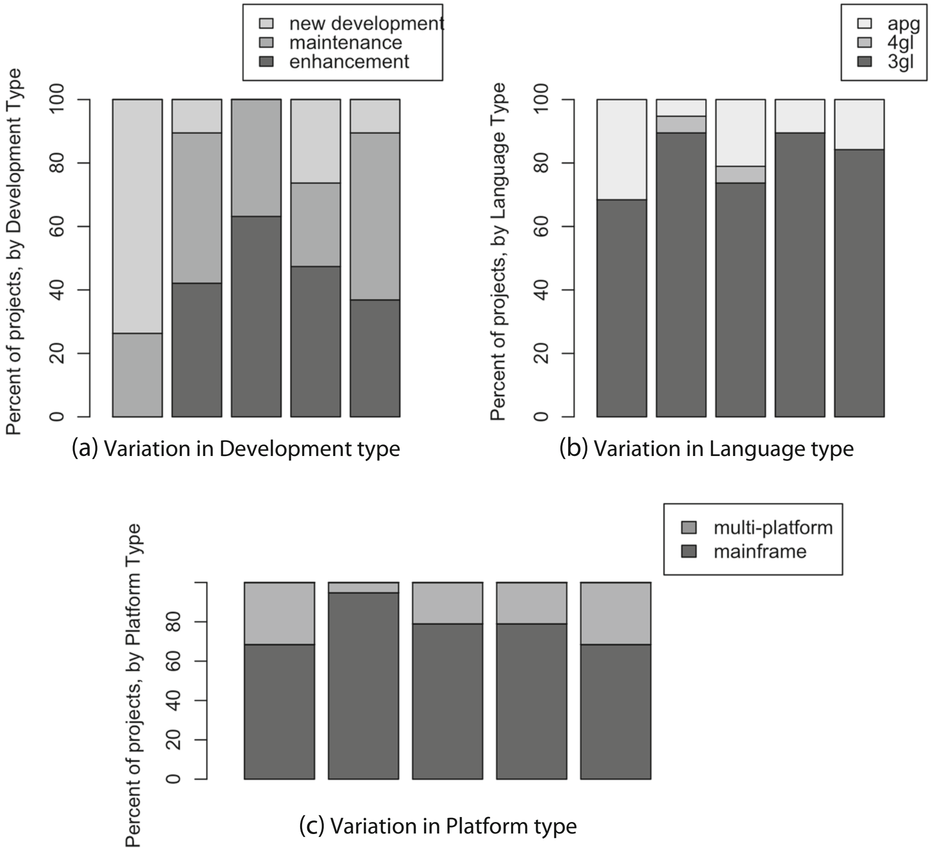
(a) Variation in Size

(b) Variation in Effort

(c) Variation in Duration

(d) Variation in PDR

Fig. 5 Variation over time in ratio-scaled variables: Organization C



(a) Variation in Development type

(b) Variation in Language type

(c) Variation in Platform type

Fig. 6 Variation over time in nominal variables: Organization C

- Development type: 24 % are new developments, 76 % are maintenance or enhancement projects.
- Platform: 78 % are mainframe projects, 22 % are multi-platform projects.
- Sector: all are from the insurance sector.

Figures 5 and 6 again show some variation over time in project characteristics:

- Size (Fig. 5a), Effort (Fig. 5b) and Duration (Fig. 5c) are all much higher in the first 20 % of projects. They then drop to much lower values, before gradually (but only slightly) increasing.
- PDR is fairly stable, except for a drop in the second 20 % of projects (Fig. 5d).
- New developments dominate the first 20 % of projects, but barely feature thereafter (Fig. 6a).
- Language type (Fig. 6b) and Platform type (Fig. 6c) fluctuate, but not by much and with no particular pattern.

4.3 Logarithmic Transformation of Size and Effort

An important check when building a regression model is that residuals are normally distributed.

The very large number of models built in this research made it impractical to check the distribution of residuals manually for every model. Instead, we automated the use of the Shapiro-Wilk test (setting statistical significance at $\alpha = 0.05$) to check whether the residuals were normally distributed, after each model was built; and we ran the whole experiment twice: once with Effort as the dependent variable and Size as an independent variable, and again with log(Effort) as the dependent variable and log(Effort) as an independent variable.

Without the log transform, in every data set Size and Effort were not normally distributed; residuals were normally distributed in 61 % of models overall, varying from 52 to 68 % among the three organizations. With the log transform, in every data set log(Size) and log(Effort) were normally distributed; residuals were normally distributed in 72 % of models overall, varying from 58 to 91 % among the three organizations.

A different question is whether using the log transform affects the accuracy of the estimates. We found no window size, for any of the organizations, at which there was a statistically significant difference in MAE between using the log transform or not (two-sided pairwise Wilcoxon signed-rank test, overall significance level set at 0.05, Holm-Bonferroni correction applied).

We concluded that accuracy of estimates does not affect the decision of whether or not to use the log transform; that there is an advantage in using the log transform by making it more likely that residuals are normally distributed; and hence the log transform is worthwhile.

We note that S3 applied the log transform to Size and Effort, so doing so here supports direct comparison. Moreover, we note that applying the log transform to Size and Effort is a common choice by researchers and also statisticians working in this research field (Kitchenham and Mendes 2009; Mendes 2014; Mendes and Mosley 2008).

The rest of this paper is based on models that use log(Effort) as the dependent variable and log(Size) as an independent variable.

4.4 Choosing Between Estimation Models

As we used multiple regression, we assumed that 10 projects per independent variable is desirable (Tabachnick and Fidell 1996).

If the training set was large enough, all independent variables could be included in a single stepwise regression process, leading to a single best model.

However, if a training set contained few projects, it could be possible to investigate models containing different sets of variables (e.g. size and language type, or size and platform, but not all of size, language type, and platform together because the data set was too small to support that many independent variables). In this case, every possible model, considering every combination of independent variables that could be supported by the amount of data, was investigated. If it turned out that there was more than one possible model in which all independent variables were significant, some criterion was needed to decide which model to prefer. We considered two: highest adjusted R^2 (since that explains the greatest amount of variation in effort); and lowest MAE (since MAE is our accuracy criterion).

We ran the entire experiment twice, using each of these decision criteria. Results showed no significant difference when employing MAE instead of adjusted R^2, across all organizations and window sizes. Hence we chose to use highest adjusted R^2 as the decision criterion, as in S3.

4.5 Influential Data Points

To prevent models from being unduly influenced by large residuals and highly influential data points, we used Cook's Distance statistic (Cook 1977) to identify projects that exhibited jointly a large influence and large residual. When using this statistic, any projects with Distance greater than 4/N, where N represents the total number of projects, are considered to have high influence on the results. When there are highly influential projects the stability of the model needs to be tested by removing these projects, and observing the effect of their removal on the model. If the coefficients remain stable and the adjusted R-squared increases, this indicates that the highly influential projects are not destabilizing the model and therefore do not need to be removed.

In detail, the following approach was used. Calculate Cook's distance values for all projects. Following (Maxwell 2002), projects with distances higher than 3 = (4/N) were immediately removed from the training set. Those with distances higher than 4/N but smaller than (3 = (4/N)) were removed temporarily in order to test the model stability, by observing the effect of their removal on the model. If the model coefficients remained stable (no change of sign in the value of any coefficient, and no change by more than 25 % in the value of any coefficient) and adjusted R^2 was the same or better (at least 99 % of the original value), the influential projects were retained in the data analysis. Otherwise the influential points were removed.

Few projects were removed. Those that were removed tended to be the same ones repeatedly. Inspection of the data showed that these projects were notably different to the other projects in the data sets. For Organization A, three projects were frequently removed: one had an unusually small effort for its size, one had an unusually large effort for its size, and one was an outlier in both size and effort (triple the size of any other project). For Organization B, two projects were frequently removed: one was an outlier in effort (much larger than any other project) and the other had an unusually low effort for its size. For Organization C, two projects were frequently removed: one had a very low effort for its size, the other had an unusually small size and an unusually high effort for its size.

4.6 Estimating Effort for a Single Project

As in S3, a chronological split approach, taking into account a project's age, was used to estimate the effort for each project. The following steps were repeated for each project *p* in turn, until effort estimates were obtained for all projects:

1. The starting date (*sd*) for *p* was used to split the data set into two groups: *completed* projects that had finished prior to *sd*, and *active/future* projects that were active or had not yet started at *sd*.
2. If a window is being considered, any *completed* projects that did not fall within the window of "most recent" projects were removed from the set. In detail: the R program selects as candidate training projects those whose finish date is prior to p's start date; sorts the candidate projects in increasing order of finish date; and selects the *N* projects with the greatest finish date (the most recent completions). If multiple projects finished on the earliest of those finally-selected completion dates,

all are selected - so occasionally there could be a small number of extra projects more than N in the training set.

3. The remaining set of *completed* projects was used as the training set in order to build a regression model M.
4. Cook's distance (Cook 1977) was used to determine whether any highly influential *completed* projects should be removed (described below); if any were removed, M was then refitted using the reduced data set.
5. M was applied to p's data in order to obtain an effort estimate for p.

All of the models herein were built using an automated process, programmed in the statistical programming language R[4]. The procedure was as follows:

- Size and Effort were both transformed to a natural logarithmic scale.
- Independent variables whose value was missing for the project to be estimated were not considered for inclusion in the estimation model.
- Every model had log(Effort) as the dependent variable, and included log(Size) as an independent variable. Beyond that, given a training set of N projects, no model was investigated if it involved more than $N/10$ independent variables (rounded to the nearest integer).
- Models were built using backward stepwise multivariate regression[5]. The regression model produced an estimate of log(Effort); this was transformed to an estimate of actual effort before evaluating accuracy.
- To prevent models from being unduly influenced by extreme data points, Cook's distance was used as described in Section 4.5.
- If it turned out that there was more than one possible model in which all independent variables were significant, the model with highest adjusted R^2 was preferred.

4.7 Moving Windows

4.7.1 Evaluation Data Sets

When evaluating the difference between two approaches (using a window or not), it makes no sense to include projects where there could be no difference. This is the case during the initial stages, while the window fills up. If a window of N projects is used, no projects can be excluded from the window until at least $N+1$ are complete.

For example, at the start date of the 28th project from Organization A, 18 projects had finished and 9 were still active. By the time the 29th project started, 24 had now finished and 4 were still active. A window of 20 projects (i.e. only the 20 most-recently-completed projects are retained) could make no difference for the first 28 projects in the sequence, because the set of all completed projects still fits within the window. However, at the start of the 29th project, the first 4 to finish out of the 24 completed projects are excluded from the window, and from here on the use of a window can make

[4] Using R version 3.2.2 and relevant packages as current at January 2016.

[5] Using the "fastbw()" function from Harrell's "rms" package for R.

a difference compared to retaining all completed projects. Hence we evaluated the impact of using a window of 20 projects by comparing the two estimates (with and without the window) for projects 29 onwards in the sequence.

As the window size increases, the set of evaluation projects decreases.

We applied the same approach for every window size both in S3 and in this study.

4.7.2 Range of Window Sizes

In studies where a data set is divided into training and validation sets, it is common to use a split of about 2:1, respectively (Han and Kamber 2006). Following this guideline, the maximum window size that can be considered is that for which at least one third of the projects could be affected by the use of a window. To determine this window size, the latest start date must be found at which one third of the projects have yet to start; the maximum window size is then one less than the number of projects that have finished by that date. In S3 this was 120 projects; for Organization A it was 110 projects; for both Organizations B and C it was 50 projects.

To determine the smallest useful data set size, we inspected the regression models produced for each project and each window size. Many regression models, usually based on few data points, were not significant at the 5 % level. The question was therefore what was the minimum window size at which the regression model was always significant at the 5 % level? In S3 and for Organization A this was 20 projects; for both Organizations B and C it was 15 projects.

Therefore, we investigated the data sets with these ranges of window sizes:

- Organization A: 20 to 110 projects
- Organization B: 15 to 50 projects
- Organization C: 15 to 50 projects

4.8 Prediction Accuracy Measures

The most common measures used in software engineering to compare different effort estimation techniques are the mean magnitude of relative error (MMRE), and prediction at level *l* (Pred(*l*): the fraction of estimates that are within *l* % of the actual values). It is suggested (Conte et al. 1986) that *l* should be set at 25 % and that a good prediction system should offer this accuracy level 75 % of the time. However, MMRE has been criticized as a biased accuracy measure (Foss et al. 2003; Shepperd and MacDonell 2012).

Mean absolute error (MAE; also referred to as mean absolute residual (MAR)) is also commonly used. It is not biased towards over- or under-estimates.

Shepperd and MacDonell proposed a standardized accuracy measure SA to avoid the problems of MMRE (Shepperd and MacDonell 2012). Its calculation involves repeated sampling with replacement of effort values from the training set. They note that this converges to the mean of the effort values in the training set. As we use the mean model as a baseline (see section 4.9), we do not use the SA measure as well.

S3 used MMRE and MAE to measure the accuracy of effort models. In this paper we only use MAE.

To test for statistically significant differences between the accuracy of predictions with and without a window, we used the paired-samples two-sided Wilcoxon signed-rank test, setting the overall statistical significance level at 5 %. Applying the Holm-Bonferroni correction, for Organization A the significance level for a single test was 0.05/91 = 0.00055; for Organizations B and C it was 0.05/36 = 0.00139. All calculations were carried out using the statistical language R.

4.9 Baseline Models

When comparing different effort prediction models/techniques, it is important to also compare their accuracy to at least one benchmarking approach, so to be able to assess whether the model/technique-based predictions are significantly superior to the benchmark; if they are not, the use of a benchmark would suffice. Two measures are commonly used as benchmarks: the mean and median effort of the training set projects (Mendes 2014; Mendes and Mosley 2008; Minku et al. 2015). We used both of these measures herein. Note that they were not used in the original study we are replicating, so this is also a change to the original experimental set-up employed in S3.

4.10 Presentation of Results

The accuracy of estimates is presented in the following ways:

- By tabulating MAE, for estimates produced with and without the use of a window, and graphing differences between them. Whether it makes a difference to use the window is assessed by comparing the accuracy of estimates with and without the window, on only those projects where a difference is possible.
- By pair-wise two-sided Wilcoxon signed-rank tests for statistically significant differences in the values of MAE. Statistically significant differences are noted, and the window sizes at which they arise are marked on the figures.

5 Results

5.1 Comparison with Baseline Models

For all three organizations, with every window size and also with the growing portfolio, MAE with the regression model was significantly different from (and lower than) MAE with either the baseline mean model or the baseline median model (two-sided Wilcoxon signed-rank test, overall significance level set at 0.05, Holm-Bonferroni correction applied to each individual comparison).

What these results show is that all the regression models that were built provide superior accuracy to both of the two benchmarking measures, and thus would be a preferred choice to be used by a company, rather than to rely on either the median or mean effort of their past projects (training set).

5.2 Organization A

The estimation models generated for the projects from Organization A varied from project to project, and from window size to window size. All included log(Size) as an independent variable, by definition; most of the models also included Development type or Language type, but rarely both; some included both Development platform and Language type.

Table 6 (in the Appendix; the same information is presented graphically in Fig. 7) shows the effect on MAE of windows of different sizes for Organization A.

Figure 7a shows the MAE values with windows and with the growing portfolio. When the line is above zero, MAE is better with the growing portfolio than with windows. Figure 7b plots the percent difference in MAE against window size. Although the difference is always a positive number, no difference is statistically significant. Therefore, for Organization A, using a window makes no difference to the accuracy of estimates, for any window size.

In summary, for Organization A there is no reason to use a window, instead of retaining all training data.

5.3 Organization B

The estimation models generated for the projects from Organization B varied in structure and coefficients from project to project, and from window size to window size, but much less than for Organization A. All included log(Size) as an independent variable, by definition; some (generally the most accurate) also included Language type.

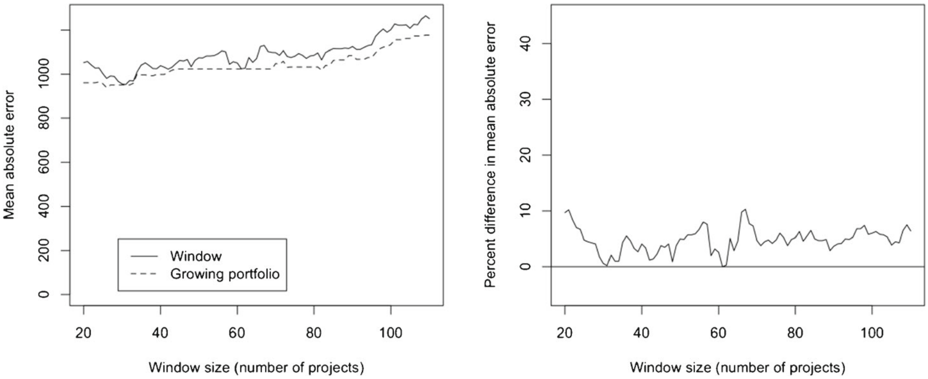
Table 7 (in the appendix) and Fig. 8 show the effect on MAE of windows of different sizes for Organization B. Figure 8b presents a pattern that is superficially different to Organization A: the line generally slopes downwards as window size increases, and it is slightly below zero for windows of 36 or more projects. However, no difference is statistically significant. Therefore, for Organization B, using a window makes no difference to the accuracy of estimates, for any window size.

In summary, for Organization B there is no reason to use a window instead of retaining all training data.

5.4 Organization C

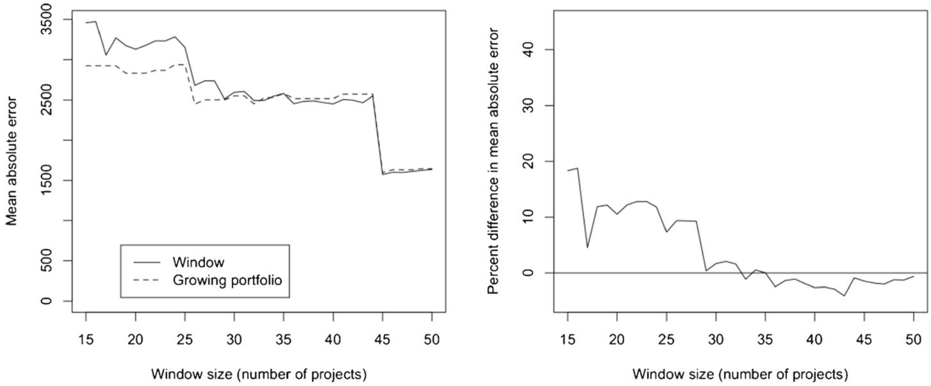
The estimation models generated for the projects from Organization C varied little in structure: most included both log(Size) and Language type as independent variables. However, the coefficients varied from project to project, and from window size to window size.

Table 8 (in the appendix) and Fig. 9 show the effect on MAE of windows of different sizes for Organization C. Figure 9b shows that there are several window sizes at which MAE is



(a) MAE with and without windows (b) Difference in MAE

Fig. 7 Accuracy statistics: Organization A (no differences are statistically significant, between using a window instead of retaining all training data)



(a) MAE with and without windows

(b) Difference in MAE

Fig. 8 Accuracy statistics: Organization B (no differences are statistically significant, between using a window instead of retaining all training data)

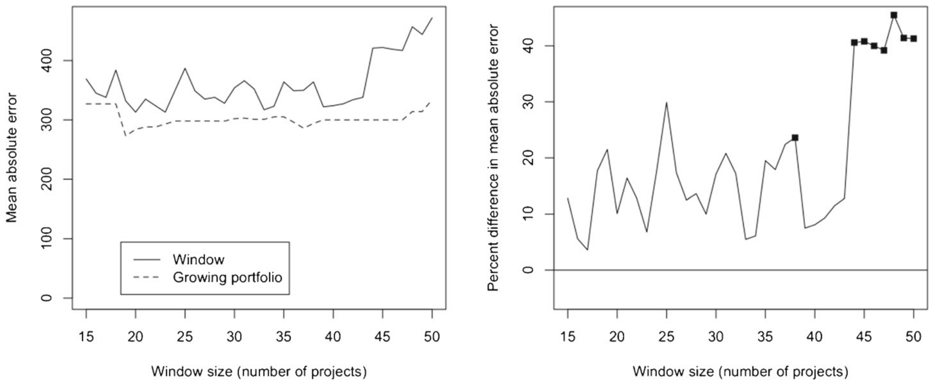
significantly worse when a window is used, instead of retaining all training data. The greatest increase in MAE is 45 %, with a window of 48 projects, but it exceeds 40 % for all windows sizes of 44 or more projects. The smallest increase in MAE is 4 %, with a window of 17 projects.

Table 8 and Fig. 10 show that even for the window sizes where the effect on MAE from using a window is statistically significant, the effect size is small (for every window size, the value of Cohen's *d* statistic, used here to evaluate effect size, is at most 0.14, well below the value of 0.2 which is considered to represent a small effect size (Cohen 1992; Shepperd and MacDonell 2012)).

In summary, for Organization C there is no evidence in favor of using a window instead of retaining all training data; on the contrary, there is weak evidence that it is harmful to do so.

5.5 Absolute Model Performance

Even though the regression models built in this study are significantly more accurate than the mean and median baseline models, they are still not very accurate. For Organization A the



(a) MAE with and without windows

(b) Difference in MAE

Fig. 9 Accuracy statistics: Organization C (statistically significant differences, in favor of retaining all training data, are marked as square points in Fig. 9b)

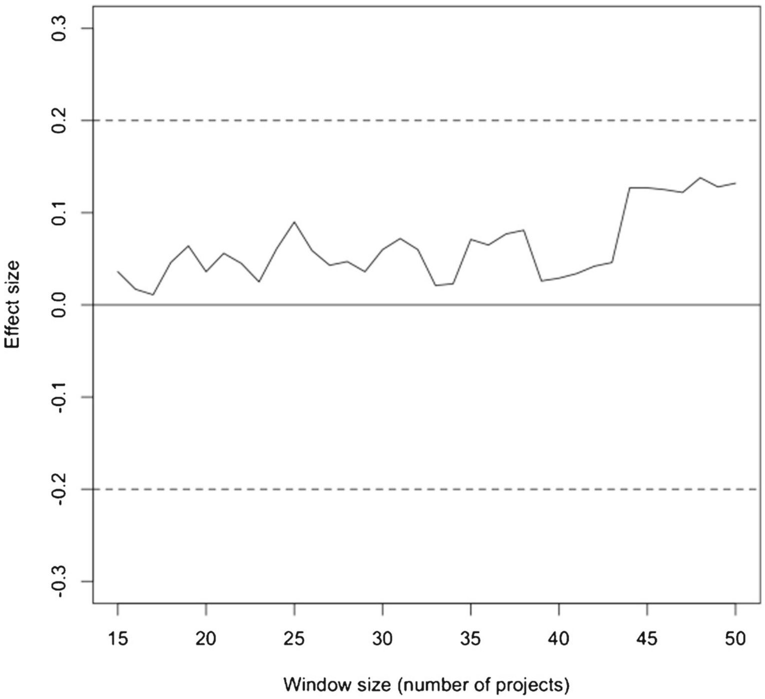


Fig. 10 Effect size: Organization C (positive values indicate that the errors are larger when using a window; effect sizes in the range -0.2 to 0.2 are considered small)

mean effort is 2533 hours, and MAE is around 1000: in relative terms, the mean error is about 40 % of the mean actual value. For Organization B, MAE varies from 1600 to about 3000 hours: 33 to 63 % of the mean effort of 4773 hours. For Organization C the MAE is around 300 hours, 15 % of the mean effort of 2015 hours.

These organizations might do better to seek other ways to estimate effort. That does not change the point of this paper, which studies the question that given regression is to be used, does it help to use windows to exclude older data?

6 Discussion

6.1 Discussion Relating to This Study

6.1.1 Research Question 1

The first research question is whether the use of a window of recent projects makes a difference to estimation accuracy.

With regard to Organization A, our results show that using a window of the *N* most recent projects, rather than retaining all training data, does not improve estimation accuracy with windows of any size that could be investigated with this data set. The difference is never statistically significant, however, and the effect size is always small.

With regard to Organization B, our results show that using a window of the *N* most recent projects also never improves estimation accuracy significantly. Some window sizes improve the MAE slightly; others make MAE worse. The difference is never statistically significant.

Finally, with regard to Organization C, for all window sizes that could be investigated with this data set, MAE is always worse when using a window, sometimes significantly worse. The effect size is in the range that is considered small.

Thus, for all three organizations, the answer to the first research question is that the use of a window of recent projects makes no helpful difference to estimation accuracy (no difference for two organizations, a harmful difference for the third).

6.1.2 Research Question 2

The second research question is whether insights can be gained by observing trends in estimation accuracy as window sizes change.

Even though few statistically significant improvements in accuracy were observed between using and not using windows, some trends in the average values of the accuracy statistics can be noted, for some of these organizations, as the window size changes. These observations help to inform the next section, which discusses similarities and differences between the original study S3 and the study detailed herein.

In regard to Organization A, Fig. 7b shows no pattern between window size and the percent change in MAE that results from using a window. The Pearson coefficient of correlation between the percent change in MAE and window size is 0.22, suggesting a very weak increase in percent change in MAE as the window size increases. It really does not matter much if windows are used or not, or which window size is used if they are adopted.

Results for Organization C are more erratic: the fluctuation in Fig. 9b is much greater than in Fig. 7b. There is little pattern between window size and the percent change in MAE, for windows of up to 43 projects. Thereafter, the increase in MAE is notably worse. The biggest increases come at the largest window sizes, at which some differences are statistically significant; this is reflected in the coefficient of correlation between the percent change in MAE and window size being 0.62.

Why the jump in MAE at a window size of 44 projects? Compared to using smaller window sizes, a possible answer comes from consideration of which projects are excluded by the window. Column 2 in Table 7 shows that the same set of 44 projects was used to evaluate accuracy for all window sizes from 39 to 47 projects. It was noted in section 4.2.3 that only 10 of this organization's projects came from the first 18 years, while 85 came from the last 7.5 years; also that the early projects were dominated by new developments, and were much larger in size, effort and duration. With windows of 39 to 41 projects, none of the first 10 projects are included in the training set for the 44 evaluation projects. With windows of 42 and 43 projects, 1 and 2 of the first ten projects are included, respectively. As the window size increases, more of the first ten projects are retained in the window. The different nature of those projects affects the estimation models detrimentally. Under this interpretation, for Organization C it is not using a fixed-sized window that helps; rather, it appears best to exclude the scattered projects from the first 18 years, and then to treat the remaining projects as a single group: they vary, but with no obvious trends, so there is no clear argument that older projects within this group are less representative.

Organization B shows a different trend. The line in Fig. 8b is not essentially flat, as for Organization A in Fig. 7b; nor is it as erratic as for Organization C in Fig. 9b. Instead, it slopes downwards as the window size increases, suggesting that the average accuracy improves as the window size increases. The coefficient of correlation between percent change in MAE and window size is -0.87, suggesting that MAE improves as the window size increases. This is the archetypal pattern seen in previous research into windows: intuition is that windows may help by excluding projects that are no longer relevant; small windows do not contain enough training data to form accurate estimation models; larger windows have a trade-off, in having more data from which to learn, but perhaps including more older data that is less relevant to current practice. However, no difference is statistically significant, at any window size: for this organization too, using windows has no effect on estimation accuracy.

In summary, for all three organizations there is no evidence in favor of using windows; with these data sets, it is as good or better to retain all past data.

6.2 Discussion Relating to Comparison Between This Study and S3

6.2.1 Differences in Method

Section 3.2 noted some changes in method, and section 4.1.2 noted some differences in data definitions, between S3 and this study.

The changes in method relate to the testing of statistical significance of differences in accuracy. S3 used both the parametric *t*-test and the non-parametric Wilcoxon test to determine statistical significance. It made little difference to the results: differences were significant for some window sizes (in favor of not using a window) with the *t*-test but not with the Wilcoxon test. In this study, results in Section 5 only used the Wilcoxon test for statistical significance. To see if the different test made a difference, we evaluated differences using the *t*-test as well. There were only a few isolated window sizes at which the tests gave different results, always in favor of not using a window. Hence the use of a different test for statistical significance did not change the finding that in these data sets windows are not helpful. Also, in this paper the Holm-Bonferroni correction was applied to significance tests. It made essentially no difference: very few p-values were 0.05 or less, so use of the Holm-Bonferroni correction did not cause the finding that in these data sets windows are not helpful.

When deciding upon which variables to use from each of the three datasets, we selected those that were the closest possible semantically to the ones used in S3—size in function points, development type and development platform. Although there are some differences in how these were measured within the context of the ISBSG dataset (used in S3) and the Finnish dataset (used herein), we believe that this was the most suitable choice, and also the most suitable compromise to enable the comparison between the results obtained from both studies.

6.2.2 Comparison Between Data Sets

Comparison Between Organization A and S3 The data set from Organization A is similar in size to the data set that was analyzed in S3: about 200 projects in each case. Both data sets are large enough to support experimentation with a wide range of window sizes, and to permit the effect of several independent variables to be determined.

The projects in S3's data set cover a narrower time span than the projects in Organization A's data set (8 years, rather than 20 years for Organization A). Therefore intuitively one might expect windows to have more effect for Organization A; however, that is not the case.

S3’s data set included projects from a range of industry sectors, while the data analyzed from Organization A was entirely from one sector (insurance). Homogeneity of industry sector is the most striking difference between these two data sets.

As for all three of the Finnish organizations, projects in S3’s data set shifted over time, with fewer new developments and more enhancement/maintenance projects. For S3, this was accompanied by a decline over time in average project size, effort and duration. Organization A had a similar pattern in development type, but not in size, effort or duration. Unlike Organization A, S3’s data set showed no trends in language type or platform type.

S3’s data set showed a slight trend of improving productivity over time. Organization A’s data showed worse productivity (higher PDR) in the earliest projects (Fig. 1d) but no trend thereafter. When productivity remains similar across most projects, the hypothesis that older projects are less representative may not apply, weakening the potential value of using windows.

Comparison Between Organization B and S3 The data set from Organization B is about half the size of S3’s data set. This means that a narrower range of window sizes can be investigated, and also fewer independent variables can be selected for each model.

S3’s data set contained projects that presented a narrower time span than the projects in Organization B’s data set (8 years rather than 16.5 years). S3’s projects roughly span the first half of Organization B’s projects, by the calendar.

S3’s data included projects from a range of industry sectors, while Organization B’s data is almost entirely from one sector (Public Administration). As with Organization A, homogeneity of industry sector is a striking difference between the two data sets.

Organization B’s projects showed shifts over time in language type, development type, and platform type. They also trended upwards in size, effort and duration for the first 60 % of projects, before dropping to much smaller values for those metrics. The only characteristics in common with S3’s data are a shift from new developments towards enhancement/maintenance projects, and (apart from a spike in the middle 20 % of projects) a general decline in PDR. Yet the shape of the graph in Fig. 8b is more like that for S3 (Lokan and Mendes 2009b) than for either Organization A or C. Perhaps the trend in PDR is the most important factor.

Comparison Between Organization C and S3 The data set from Organization C is about half the size of the data set that was analyzed in S3. This means that a narrower range of window sizes can be investigated, and also fewer independent variables can be properly analyzed.

The projects in S3’s data set presented a narrow time span (8 years rather than 25 years). However, most of the projects Organization C’s data set came also from an 8-year span, matching the duration of S3’s projects. However, they did not overlap very much in time: S3’s projects’ duration ended in 2001, at which time most of Organization C’s projects were yet to start.

S3’s data set included projects from a range of industry sectors, while Organization C’s data set comprised projects entirely from one sector (insurance). As with the other two Organizations investigated herein, homogeneity of industry sector is a striking difference between the two data sets.

Organization C's projects differ from S3's in how they evolve over time. Both organizations have more new developments to start with and more maintenance/enhancement projects later, but the shift is much stronger in Organization C. S3's data set showed a slight trend of improving PDR over time, but Organization C has no monotonic trend in PDR. This weakens the scope for windows to make a difference. The key to Organization C's data is that the later projects are very different to the earlier ones.

A major difference between the datasets investigated here and that studied in S3 is the time span that they cover. S3's data spans 8 years, while the three datasets studied here span much longer periods. This can mean that even windows containing few projects may cover several years—this is certainly true in Organization C, where the first ten projects came in intermittently over a period of 18 years. As a result, there may be several shifts in the nature of the projects contained in a window. In that case, the window does not capture "recency", so it is not likely to help. This could contribute to the conclusion that windows are not helpful for these organizations.

7 Threats to Validity

This study has some limitations and threats to validity.

7.1 External Validity

First, the Finnish repository is a convenience sample, and therefore does not represent a random sample of projects from a defined population. Project data was volunteered by Finnish ICT companies, and it is likely that these companies are interested in applying metrics to software process improvement. Each data set used herein came almost entirely from a single industry sector (two from insurance, one from public administration), thus we believe that our results may be generalizable to other companies in these same industry sectors, and also to companies that develop similar projects to those that were used herein. However, note that such generalization cannot go beyond the two groups that have just been mentioned, given that the data employed is solely a convenience sample (as abovementioned).

We believe that the increase in studies like ours, using other industry-based data sets will lead to two important outcomes: first, we will obtain a wider picture relating to the use of a chronological split when estimating effort; second, we will be able to increase our trust in the findings.

Note that by studying single-company data, numerous potential sources of variation are likely to have been removed.

It is vital for the validity of this research that project start and end dates are correctly recorded. We note that considerable effort is made to assess the quality of the data when it is added to the Finnish data set, and that start and end dates are mandatory in this process (Forselius 2006). Therefore we trust that the dates are accurate.

7.2 Conclusion Validity

All the models employed in this study were built automatically. Automating the process necessarily involved making some assumptions, and the validity of our results depends on those assumptions being reasonable. These are: residuals are checked for being normally distributed by automating the use of the Shapiro-Wilk test (setting statistical significance at $\alpha = 0.05$)—usually they are normally distributed, but this is not checked manually for each model; when choosing between two models in which all independent variables were significant, the one with higher adjusted R^2 was chosen as the preferred model (details were given in Section 4.4); multi-collinearity between independent variables is assumed to be handled automatically by the nature of the stepwise procedure. Based on our past experience building models manually, we believe that these assumptions are acceptable. One would not want to base important decisions on a single model built automatically, without at least doing some serious manual checking, but for calculations such as chronological split estimation across a substantial data set we believe that the process here is reasonable.

In summary, with respect to the conclusion validity we carefully applied the statistical tests that were the suitable choice given the data we had, verifying all the required assumptions. In addition, we also reported effect size to provide additional evidence to assess the relevance of our results.

8 Conclusions

This paper's contribution is to simulate a scenario in which only a company's most recent projects are used to estimate effort for a new project. This simulation is representative of the existing practice in numerous companies (one of the authors has first hand experience of it with companies in New Zealand and Brazil). The results are therefore relevant in practice.

With regard to its research contribution, the paper presents a replication of previous work with new data sets. A detailed description has been presented of the data sets, and how they compare to the data set previously studied. As the results contradict previous results, this can help in developing an understanding of when windows may or may not be helpful. The paper also presents a detailed description and comparison of all the previous work to date on the use of chronology in software effort estimation.

This paper replicated a previous study (S3), investigating the same issue examined in S3: whether using a chronological split that takes into account a project's age (moving window) had a significant effect on the accuracy of effort estimation models, when compared to models built using all available projects as training data. Three different single-company data sets were employed, and our results showed that the use of moving windows did not present a statistically significant improvement on accuracy, when based on MAE; however windows showed significantly superior accuracy than the two baseline measures employed—mean and median effort. Overall, for the three data sets employed herein, a window of the *N* most recent projects never improves estimation accuracy significantly, when compared to retaining all the training data. These results contradict those for S3, when based on window sizes greater than 85, and MMRE.

Previous research (S3, S11) suggested that the disadvantage of small windows lies with not having enough data to investigate the effect of many independent variables, but that this effect goes away with windows that contain a large amount of data. Here, even with large windows it is better to retain all training data.

Further investigation is clearly needed to understand this.

There are several possible reasons as to why results diverged, which have been detailed in Section 6.

Based on the results from previous studies and this study, we propose the following interpretation for relationships between the numbers of independent variables considered, the amount of variation in effort explained by the independent variables, and window sizes:

- With heterogeneous data, at least three or four independent variables are required in order to explain at least 60 % of the variation in effort; this means at least 40 or so projects are needed for statistical credibility.
- Small windows are clearly bad: they do not hold enough data for regression models to identify the effect of different independent variables, or for estimation by analogy to have sufficient good analogies to choose from. Therefore, in such circumstances, using a window is harmful to estimation accuracy. This must be kept in mind when considering using windows: with small windows, the advantage of only having recent data is outweighed by the disadvantage of not having enough data.
- Large windows mean that few projects are rejected as being too old to stay within the window: there is less difference between using the window and using the whole set of projects completed so far. Using the window is neither harmful nor helpful; it just becomes irrelevant.
- In between, we may hope to see a range of window sizes for which there can be enough data to be useful, but not so much data that the oldest projects reflect tasks or practices that are out of date. This would mean that using the window might be helpful.
- There might not be a helpful in-between range: if tasks and practices are stable, windows won't help.
- Windows also will not help if tasks and practices are not stable, but the rate of accumulating data is low. A window would have the effect of discounting older projects, but it would never contain enough data to be useful.
- Windows may help if they contain similar projects that reflect current practices, and exclude older projects that no longer reflect current practices. If a window covers a long time span, there may be several shifts in the nature of the projects it contains. In that case, the window does not capture "recency", so it is not likely to help. Matching the window size to the rate of change in an organization's projects is important.

This study, S3, and all the previous studies that investigated moving windows employed single-company data sets. However there are numerous situations, when companies do not have their own datasets of past projects, that lead those companies to employ datasets containing past projects volunteered by other companies (cross-company datasets) (Kitchenham et al. 2007). Therefore, a legitimate question that arises is whether the patterns we have observed herein, in S3 and other studies investigating moving windows, would remain the same if employing a cross-company data set. This question is an avenue for future investigation.

This study differs from S3 in that all of the data sets studied herein came from single industry sectors, whereas S3's data came from multiple sectors. This homogeneity in the data may partly explain why windows are not helpful for these organizations.

This study and S3 both used a single effort estimation technique; however, Amasaki and Lokan (S5) (2012) employed two different techniques and obtained good results with either technique, thus motivating further investigation of moving windows using different techniques. Note that we have employed herein a regression technique because this was also the technique used in S3; however, given the large number of nominal scale variables in all the data sets investigated herein, perhaps another technique may also be suitable (e.g. CART). Therefore the use of other estimation techniques with the same data sets employed herein and in S3 is also an avenue of future work.

In the end, fixed-size and fixed-duration windows are only approximate mechanisms for taking into account points in time where significant changes happen in an organization's projects, practices, or staff. Finding how to recognize those important points in time, and how to use that knowledge in effort estimation, are crucial future work.

Finally, this paper reports a negative result: windows have been studied in detail for these organizations, and found not to provide any benefit for effort estimation. The practical significance of the paper is to identify that it is not necessarily best to discard older data, contradicting the intuition that older data is less relevant for effort estimation. Organizations using historical data for effort estimation should understand their data and how it has evolved, rather than automatically discard older data.

Acknowledgments We thank Pekka Forselius for making the Finnish data set available to us for this research.

Appendix

Tables 6, 7, and 8 present in full numerical detail the information that is plotted in Figs. 7a and b, 8a and b, and 9a and b. In each table, the first column shows the window size. The second column shows the number of projects for which the use of a window of that size could make a difference to the estimate, compared to using the growing portfolio. The third column shows the value of MAE across all of those projects, when a window is used. The fourth column shows the value of MAE for the same set of projects, when a window is not used and instead the training set always contains all projects completed so far. The fifth column shows the difference between columns 3 and 4; a positive number means that MAE is worse when a window is used compared to retaining all data, and a negative number indicates that MAE is better when a window is used compared to retaining all data. The sixth column presents the difference in MAE (the fifth column) as a percentage of the MAE without a window (the fourth column). The seventh column shows the p-value when the paired-samples two- sided Wilcoxon test was used to test the hypothesis that MAE with a window differed from MAE with the growing portfolio; values below 0.00055 indicate a statistically significant difference for that test (applying the Holm-Bonferroni correction to the overall significance level of 0.05). The final column shows the effect size *r,* calculated from Cohen's *d* statistic [6] (Cohen 1992): $r = d / sqrt(d^2 = 4)$. Effect size is considered small if it is below about 0.2, medium at about 0.5, and large above about 0.8 (Cohen 1992; Shepperd and MacDonell 2012).

[6] Using the "cohen.d()" function from the "effsize" package in R.

Table 5 Summary of related work

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
S1 (Lokan and Mendes 2008)	ISBSG database Release 10	228/678	project-by-project split	1) Using a project-by-project chronological splitting for both cross- and single-company projects: a) How successful is a cross-company model at estimating effort for pro jects from a single company, when the model is built from a data set that does not include projects from that company? b) How successful is a cross-company model, compared to a single-company model? 2) Using a leave-one-out cross-validation approach: a) How successful is a cross-company model at estimating effort for pro jects from a single company, when the model is built from a data set that does not include projects from that company? b) How successful is a cross-company model, compared to a single-company model? 3) Using a leave-two-out cross-validation approach:	Automated stepwise regression procedure in R	Leave-one-out and leave-two-out cross-validation using all the projects in the CC and SC datasets.	Mean absolute residuals, Mean MRE (MMRE), Mean EMRE (MEMRE), Mean z, Pred (0.25), Median absolute residuals, Median MRE (MdMRE), Median EMRE (MdEMRE), Median z, Wilcoxon paired-samples test on absolute residuals and *z* (no effect size re ported)	No, when based on absolute residuals Yes, when based on z values

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
				a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company? b) How successful is a cross-company model, compared to a single-company model? 4) Does the use of a project-by-project chronological split, instead of a leave-one-out cross-validation, affect the accuracy of the models? 5. Does the use of a project-by-project chronological split, instead of a leave-two-out cross-validation, affect the accuracy of the models? 6. Does the use of a leave-two-out cross-validation, instead of a leave-one-out cross-validation, affect the accuracy of the models?				
S2 (Mendes and Lokan, 2009a)	ISBSG database Release 10	361/520 387/539	Date-based selection (two dates: 1st	1) Using a date-based selection of cross-company (CC) and	Manual stepwise regression+	Training and validation sets containing the	Magnitude of Relative Error Relative to the Estimate (EMRE),	No.

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
			January 2001, and 1st January 2002)	single-company (SC) projects: a) How successful is a CC model at estimating effort for projects from a single company, when the model is built from a data set that does not include that company's projects? b) How successful is a CC model, compared to a SC model? 2) Using a random selection of CC and SC projects: a) How successful is a CC model at estimating effort for projects from a single company, when the model is built from a data set that does not include that company's projects? b) How successful is a CC model, compared to a SC model? 3) Does the use of a chronological split, instead of a random split, affect the accuracy of the models? 4) Does the use of different chronological splits produce different results?	linear regression	same projects as in the data-driven selection; however random ly organized.	Pred(0.25), Wilcoxon paired-samples test on absolute residuals and *z* (no effect size reported)	
S3 (Lokan		ISBSG: 228	Fixed-size moving window		Automated stepwise	project-by-project split		

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
and Mendes 2009b)	ISBSG database Release 10			1) Assuming a project-by-project approach to effort estimation, is there a difference between the accuracy of estimates using prediction models that are built using all available data in a training set, and the accuracy of estimates using prediction models that are built using only the N most recent projects in the training set? The null hypothesis is that there is no difference, for all values of N. 2) If there is a difference, can insights be gained by observing trends in estimation accuracy as N varies?	regression procedure in R		Mean Magnitude of Relative Error (MMRE), Pred(0.25), paired-samples *t*-test on absolute residuals, (no effect sizes reported); also used Regression Error Characteristic Curves to visualize the distributions of accuracy measures.	Yes, for window sizes containing 75 projects
S4 (Amasaki et al. 2011)	Kitchenham dataset Maxwell dataset	Kitchenham: 105 Maxwell: 63	Fixed-size moving window	1) Whether any use of a window of recent projects made a difference to accuracy? 2) Whether insights can be gained by observing trends in accuracy as the window size varies?	Estimation by Analogy	project-by-project split	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Pred(0.25), Wilcoxon paired-samples test on absolute residuals (no effect size reported)	No.
		228						

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
S5 (Amasaki and Lokan 2012)	ISBSG database Release 10		Fixed-size moving window	1) With the single company dataset used in (Amasaki and Lokan 2012): • Is there a difference in the accuracy of estimates using EbA with and without windowing? • If there is a difference, are there any insights with regards to trends that may be drawn as window sizes vary? 2) In comparison to the results in (Amasaki and Lokan 2012): • Do the results with the same data set differ as the estimation technique changes? • What may be suggested by any difference in results?	Automated stepwise procedure in R, and Estimation by Analogy implemented using IBk in Weka	project-by-project split	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Pred(0.25), Wilcoxon paired-samples test on absolute residuals and z (no effect size reported)	Yes, for both techniques, when using large window sizes
S6 (Lokan and Mendes 2012)	ISBSG database Release 10	228	Fixed duration moving window, where durations range from 12 to 84 months	1) Assuming a project-by-project approach to effort estimation (meaning a separate training set is formed for each project to be estimated), is there a difference between the accuracy of estimates using prediction models that are built using all available data as the training set, and the accuracy of estimates using	Automated stepwise regression procedure in R	project-by-project split	Mean absolute residual, Mean MRE, Wilcoxon paired-samples test on absolute residuals (no effect size reported)	Yes, window durations of 12 or 13 months, based on absolute residuals; from 25 to 34 months, based on MRE; and from 35 to 48 months, based on both absolute residuals and MRE.

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
				prediction models that are built considering only those projects whose development occurred during the last N months? The null hypothesis is that there is no difference, for all values of N. 2) If there is a difference, can insights be gained by observing trends in estimation accuracy as N varies? 3) How do these results compare with results based on fixed-size windows (windows containing a fixed number of projects)?				
S7 (Tsunoda et al., 2013)	ISBSG database Release 10, Maxwell dataset, Kitchenham dataset	ISBSG:217 Maxwell: 37 Kitchenham: 135	Fixed-size moving windows, dummy variable of moving windows, dummy variables of equal bins, dummy variables of year, year predictor, and serial number	1) Do different methods for treating timing information lead to different estimation accuracy? 2) (If the anwer of 1) is "yes") Which methods are effective for constructing effort estimation models? 3) Is using timing information always effective for constructing effort estimation models?	Multivariate regression.	All data from each dataset	Average and Median of: Absolute error, Magnitude of Relative Error (MRE), Magnitude of Error Relative to the Estimate (MER), and Balanced Relative Error (BRE). Wilcoxon paired-samples test (no effect size reported)	No, for both Maxwell and Kitchenham datasets Yes, for ISBSG database (Fixed-size moving windows, dummy variable of moving windows, dummy variables of equal bins, year predictor, and serial number)

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
S8 (Amasaki and Lokan 2013)	ISBSG database Release 10	228	Weighted fixed size moving windows, fixed size moving windows	1) Is there a difference in the accuracy of estimates between moving windows and weighted moving windows? 2) If there is a difference, are there any insights with regards to trends with the use of different weighting functions?	Automated linear regression procedure in R	Fixed size moving windows	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Pred(0.25), Wilcoxon paired-samples test on absolute residuals (no effect size reported)	Yes, for all window sizes.
S9 (Amasaki and Lokan 2014a)	ISBSG database Release 10	228	fixed size moving windows	1) Is there a difference in the accuracy of estimates using CART with and without windowing? 2) Can insights be drawn from trends in accuracy as window sizes vary?	Classification and Regression Trees.	project-by-project split	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Wilcoxon paired-samples test on absolute residuals (no effect size reported)	Yes, for windows containing 40 to 60 projects
S10 (Amasaki and Lokan 2014b)	ISBSG database Release 10	228	Weighted fixed duration moving windows, fixed duration moving windows	1) Is there a difference in the accuracy of estimates between unweighted and weighted moving windows, when the definition of window size is based on duration? 2) Can insights be gained from difference of trends in accuracy among weighted and unweighted moving windows as the window size varies?	Automated linear regression procedure in R	project-by-project split	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Mean Magnitude of Absolute Error (MAE), Pred(0.25), Wilcoxon paired-samples test on absolute residuals (no effect size reported)	Yes, for durations of 30 months, and 49 to 84 months.

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
				3) How do these results compare with results based on fixed-size windows (windows containing a fixed number of projects)?				
S11 (Lokan and Mendes 2014)	ISBSG database Release 10; Finnish dataset	ISBSG: 228 Finnish: 198	Project-by-project split, fixed duration moving windows	1. Assuming a project-by-project approach to effort estimation (meaning a separate training set is formed for each project to be estimated), is there a difference between the accuracy of estimates using prediction models that are built using all available data as the training set, and the accuracy of estimates using prediction models that are built considering only those projects whose development occurred during the last N months? The null hypothesis is that there is no difference, for all values of N. 2. Can insights be gained by observing trends in estimation accuracy as N varies? 3. How do these results compare	Automated stepwise regression procedure in R	All data from each dataset	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Pred(0.25), Wilcoxon paired-samples test on absolute residuals (no effect size reported)	No.

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
				with results based on fixed size windows (windows containing a fixed number of projects)?				
S12 (Amasaki and Lokan 2014a)	ISBSG database Release 10	228	gradual weighting	1) Does gradual weighting affect estimation accuracy? 2) Is there a difference in the accuracy of estimates between gradual weighting and moving windows? 3) Is there a difference in the accuracy of estimates when combining gradual weighting with moving windows? 4) Are there any insights with regard to trends with the use of different weighting functions?	Automated linear regression procedure in R	project-by-project split; fixed size moving windows	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Wilcoxon paired-samples test on absolute residuals (no effect size reported)	Yes.
S13 (Amasaki and Lokan 2015)	Finnish dataset	198	fixed size moving windows, fixed duration moving windows	1) Is there a difference in the accuracy of estimates with and without windows, using EbA, and using fixed-size windows? 2) Is there a difference in the accuracy of estimates with and without windows, using EbA, and using fixed-duration windows? 3) What similarities and differences are there	Estimation by Analogy	project-by-project split	Absolute residual, Magnitude of Relative Error, Mean Magnitude of Relative Error (MMRE), Pred(0.25), Mean Absolute Error (MAE), Wilcoxon paired-samples test on ab solute residuals (no effect size reported)	Yes for fixed duration moving windows

Table 5 (continued)

Study	Dataset(s)	Size WC set(s)/ size CC set(s)	Chronology approach	Research questions	Estimation technique(s)	Benchmark(s)	Accuracy measures	Does the treatment provide significantly superior prediction accuracy?
				between the results using fixed-size windows and fixed-duration windows? 4) How do these results compare with results using LR instead of EbA?				
S14 (Amasaki and Lokan 2016a)	ISBSG database Release 10	228	fixed size moving windows, fixed duration moving windows	1) Is there a difference in the accuracy of estimates with and without windows, using CART and fixed- duration windows? 2) How do these results compare with results based on fixed-size windows?	Classification and Regression Trees	project-by-project split	Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), Wilcoxon paired-samples test on absolute residuals and relative errors (no effect size reported)	Rarely for fixed duration windows, yes for fixed size moving windows
S15 (Amasaki and Lokan 2016b)	Finnish dataset	198	gradual weighting	1) Is there a difference in the accuracy of estimates between unweighted and weighted moving windows, using fixed-size windows? 2) Can insights be gained from the difference of trends in accuracy among weighted and unweighted moving windows, as the window size varies? 3) How do these results compare with results based on the ISBSG dataset?	Automated linear regression procedure in R	project-by-project split; fixed size moving windows	Mean Absolute Error (MAE), Wilcoxon paired-samples test on absolute errors,	No.

Table 6 Organization A: Mean absolute residuals by window size

Window size (N)	Number of projects evaluated	MAE with a window	MAE without a window	Differ-ence in MAE	Percent difference in MAE	p-value (MAE is different)	Effect size
20	172	1053	961	93	9.7	0.608	0.030
21	172	1058	961	98	10.2	0.597	0.031
22	172	1042	961	81	8.4	0.748	0.026
23	172	1028	961	67	7.0	0.831	0.022
24	170	1029	965	64	6.7	0.713	0.022
25	165	1003	957	46	4.8	0.884	0.015
26	162	981	939	42	4.5	0.869	0.014
27	159	992	951	41	4.3	0.504	0.013
28	159	990	951	39	4.1	0.478	0.013
29	159	968	951	17	1.8	0.783	0.006
30	159	956	951	5	0.6	0.942	0.002
31	159	953	951	2	0.2	0.968	0.001
32	158	971	951	20	2.1	0.686	0.007
33	149	969	959	10	1.0	0.936	0.003
34	140	1012	1002	10	1.0	0.724	0.003
35	139	1041	997	44	4.4	0.393	0.014
36	139	1052	997	55	5.5	0.226	0.017
37	137	1040	993	47	4.7	0.397	0.015
38	137	1026	993	33	3.3	0.477	0.010
39	135	1025	999	26	2.7	0.408	0.008
40	135	1039	999	41	4.1	0.348	0.013
41	135	1032	999	34	3.4	0.430	0.011
42	133	1023	1010	12	1.2	0.324	0.004
43	132	1032	1018	14	1.4	0.504	0.005
44	131	1048	1024	24	2.3	0.417	0.008
45	130	1063	1024	39	3.8	0.409	0.012
46	130	1060	1024	36	3.5	0.443	0.011
47	130	1067	1024	42	4.1	0.420	0.013
48	130	1034	1024	9	0.9	0.611	0.003
49	130	1063	1024	39	3.8	0.604	0.012
50	130	1075	1024	51	5.0	0.465	0.016
51	130	1074	1024	50	4.9	0.611	0.016
52	130	1083	1024	58	5.7	0.429	0.018
53	130	1083	1024	58	5.7	0.476	0.018
54	130	1085	1024	61	5.9	0.496	0.019
55	130	1093	1024	68	6.7	0.479	0.021
56	130	1106	1024	81	8.0	0.343	0.025
57	130	1102	1024	77	7.6	0.351	0.024
58	130	1045	1024	21	2.0	0.590	0.007
59	130	1057	1024	33	3.2	0.500	0.010
60	130	1051	1024	27	2.6	0.537	0.008
61	130	1025	1024	0	0.0	0.556	0.000

Table 6 (continued)

Window size (N)	Number of projects evaluated	MAE with a window	MAE without a window	Differ-ence in MAE	Percent difference in MAE	p-value (MAE is different)	Effect size
62	130	1027	1024	3	0.3	0.740	0.001
63	130	1076	1024	52	5.1	0.333	0.016
64	130	1054	1024	30	2.9	0.410	0.009
65	130	1071	1024	47	4.6	0.419	0.014
66	130	1125	1024	100	9.8	0.125	0.031
67	130	1130	1024	106	10.3	0.096	0.032
68	130	1103	1024	78	7.7	0.222	0.024
69	130	1099	1024	74	7.3	0.330	0.023
70	125	1098	1049	49	4.7	0.316	0.015
71	119	1086	1046	40	3.8	0.529	0.012
72	117	1107	1059	48	4.5	0.378	0.014
73	115	1081	1032	49	4.8	0.495	0.014
74	114	1076	1033	43	4.2	0.501	0.012
75	114	1082	1033	50	4.8	0.989	0.014
76	114	1094	1033	61	6.0	0.518	0.018
77	113	1086	1033	53	5.2	0.366	0.015
78	113	1072	1033	39	3.8	0.352	0.011
79	113	1083	1033	50	4.9	0.294	0.015
80	113	1086	1033	54	5.2	0.206	0.016
81	113	1097	1033	65	6.3	0.147	0.019
82	111	1065	1018	47	4.6	0.227	0.014
83	106	1098	1040	58	5.5	0.117	0.017
84	106	1108	1040	67	6.5	0.116	0.019
85	103	1116	1063	53	5.0	0.244	0.015
86	100	1115	1065	50	4.7	0.185	0.014
87	100	1115	1065	50	4.7	0.231	0.014
88	100	1118	1065	52	4.9	0.193	0.015
89	98	1116	1085	32	2.9	0.521	0.009
90	98	1125	1085	40	3.7	0.267	0.011
91	92	1112	1068	44	4.1	0.338	0.012
92	92	1112	1068	44	4.2	0.388	0.012
93	92	1121	1068	53	5.0	0.228	0.015
94	91	1129	1076	53	4.9	0.285	0.015
95	91	1133	1076	57	5.3	0.234	0.016
96	86	1171	1097	75	6.8	0.069	0.020
97	83	1190	1114	76	6.8	0.080	0.020
98	82	1205	1122	82	7.4	0.035	0.022
99	81	1190	1125	65	5.8	0.074	0.017
100	80	1202	1134	68	6.0	0.098	0.017
101	78	1228	1156	72	6.3	0.041	0.018
102	78	1222	1156	67	5.8	0.047	0.017
103	78	1222	1156	66	5.7	0.064	0.017

Table 6 (continued)

Window size (N)	Number of projects evaluated	MAE with a window	MAE without a window	Differ-ence in MAE	Percent difference in MAE	p-value (MAE is different)	Effect size
104	77	1224	1162	62	5.3	0.058	0.016
105	77	1208	1162	46	3.9	0.086	0.011
106	76	1226	1173	53	4.5	0.124	0.013
107	76	1224	1173	51	4.3	0.169	0.012
108	76	1250	1173	77	6.5	0.096	0.019
109	72	1265	1177	89	7.5	0.042	0.021
110	72	1252	1177	75	6.4	0.055	0.018

Table 7 Organization B: Mean absolute residuals by window size

Window size (N)	Number of projects evaluated	MAE with a window	MAE without a window	Differ-ence in MAE	Percent difference in MAE	p-value (MAE is different)	Effect size
15	80	3459	2923	536	18.33	0.183	0.040
16	80	3472	2923	549	18.78	0.067	0.042
17	80	3056	2923	133	4.55	0.067	0.012
18	80	3271	2923	348	11.89	0.042	0.029
19	76	3175	2831	344	12.16	0.339	0.028
20	76	3129	2831	298	10.53	0.614	0.024
21	76	3176	2831	345	12.18	0.283	0.028
22	75	3233	2866	367	12.8	0.031	0.030
23	75	3233	2866	367	12.81	0.024	0.030
24	70	3283	2937	347	11.81	0.004	0.028
25	70	3152	2937	215	7.34	0.032	0.018
26	68	2679	2450	230	9.38	0.242	0.026
27	66	2735	2502	234	9.34	0.071	0.026
28	66	2734	2502	232	9.27	0.037	0.026
29	66	2511	2502	10	0.39	0.085	0.001
30	62	2592	2549	43	1.69	0.042	0.005
31	58	2603	2550	53	2.07	0.019	0.006
32	56	2494	2454	39	1.61	0.008	0.004
33	54	2495	2523	-28	-1.12	0.026	-0.003
34	51	2545	2532	14	0.53	0.039	0.001
35	49	2578	2577	1	0.04	0.214	0.000
36	46	2455	2517	-62	-2.45	0.600	-0.006
37	46	2483	2517	-34	-1.37	0.104	-0.004
38	46	2490	2517	-27	-1.09	0.175	-0.003
39	46	2469	2517	-48	-1.9	0.111	-0.005
40	46	2452	2517	-65	-2.6	0.022	-0.007
41	43	2507	2571	-64	-2.5	0.021	-0.006
42	43	2498	2571	-73	-2.85	0.472	-0.007
43	43	2467	2571	-104	-4.06	0.510	-0.010
44	43	2548	2571	-23	-0.89	0.976	-0.002
45	40	1573	1596	-23	-1.47	0.995	-0.004
46	39	1601	1631	-30	-1.81	0.775	-0.005
47	39	1598	1631	-32	-1.99	0.754	-0.005
48	39	1611	1631	-20	-1.22	0.399	-0.003
49	35	1625	1646	-21	-1.27	0.206	-0.003
50	35	1636	1646	-10	-0.62	0.501	-0.002

Table 8 Organization C: Mean absolute residuals by window size

Window size (N)	Number of projects evaluated	MAE with a window	MAE without a window	Differ-ence in MAE	Percent difference in MAE	p-value (MAE is different)	Effect size
15	76	369	327	42	12.8	0.65427	0.036
16	76	345	327	18	5.6	0.59564	0.017
17	75	338	327	12	3.6	0.72944	0.011
18	75	384	327	58	17.7	0.87205	0.046
19	72	332	273	59	21.5	0.69534	0.064
20	66	313	284	29	10.1	0.85804	0.036
21	65	335	288	47	16.4	0.64735	0.056
22	65	324	288	37	12.7	0.82534	0.045
23	63	313	293	20	6.8	0.95806	0.025
24	62	350	298	53	17.7	0.75771	0.061
25	62	387	298	89	29.9	0.64607	0.090
26	62	349	298	51	17.3	0.56535	0.059
27	62	335	298	37	12.5	0.75238	0.043
28	62	338	298	40	13.6	0.28657	0.047
29	62	328	298	30	10.0	0.19826	0.036
30	61	354	302	52	17.1	0.12301	0.060
31	60	366	303	63	20.8	0.14699	0.072
32	59	352	301	52	17.2	0.21022	0.060
33	59	317	301	16	5.5	0.81793	0.021
34	58	323	305	19	6.1	0.79535	0.023
35	58	364	305	59	19.5	0.00479	0.071
36	55	349	296	53	17.9	0.00420	0.065
37	49	350	286	64	22.4	0.00252	0.077
38	46	364	294	70	23.6	0.00101	0.081
39	44	322	300	22	7.5	0.38461	0.026
40	44	324	300	24	8.1	0.15451	0.029
41	44	327	300	28	9.3	0.05282	0.034
42	44	334	300	34	11.5	0.05583	0.042
43	44	338	300	38	12.8	0.01823	0.046
44	44	421	300	122	40.6	0.00064	0.127
45	44	422	300	122	40.8	0.00108	0.127
46	44	419	300	120	40.0	0.00019	0.125
47	44	417	300	117	39.2	0.00068	0.122
48	37	457	314	143	45.5	0.00020	0.138
49	37	444	314	130	41.4	0.00027	0.128
50	34	472	334	138	41.3	0.00090	0.132

References

Amasaki S, Lokan C (2012) The effects of moving windows to software estimation: comparative study on linear regression and estimation by analogy. IWSM/Mensura 2012, Assisi

Amasaki S, Lokan C (2013) The evaluation of weighted moving windows for software effort estimation. Product-Foc Software Process Improve, LNCS 7983:214–228, **Springer**

Amasaki S, Lokan C (2014a) On the effectiveness of weighted moving windows: experiment on linear regression based software effort estimation. J Software: Evol Process 27(7):488–507

Amasaki S, Lokan C (2014b) The effects of moving windows on software effort estimation: comparative study with CART. Proc 6th Int Workshop Empirical Software Eng Pract, Osaka, Japan

Amasaki S, Lokan C (2014c) The effects of gradual weighting on duration-based moving windows for software effort estimation. 15th Int Conf Product-Focused Software Eng Process Improve, Helsinki, Finland: 63–77

Amasaki S, Lokan C (2015) A replication of comparative study of moving windows on linear regression and estimation by analogy. Proc 11th Int Conf Predict Models Data Anal Software Eng, Beijing, China 1–6:6–10

Amasaki S, Lokan C (2016a) Evaluation of moving window policies with CART. Proc 7th Int Workshop Empirical Software Eng Pract, Osaka, Japan

Amasaki S, Lokan C (2016b) A replication study on the effects of weighted moving windows for software effort estimation. Proc 20th Int Conf Eval Assessment Software Eng, Limerick, Ireland

Amasaki S, Takahara Y, Yokogawa T (2011) Performance evaluation of windowing approach on effort estimation by analogy. IWSM/Mensura 2011, Nara, pp 188–195

Azhar D, Mendes E, Riddle P (2012) A systematic review of Web resource estimation. Proc 8th Int Conf Predict Models Software Eng, Lund, Sweden: 49–58

Azzeh M, Cowling PI, Neagu D (2010) Software stage-effort estimation based on association rule mining and fuzzy set theory. Proc 10th Int Conf Comput Inform Technol, Bradford, UK: 249–256

Bibi S, Stamelos I, Angelis L (2008) Combining probabilistic models for explanatory productivity estimation. Inf Softw Technol 50(7–8):656–669

Bibi S, Stamelos I, Gerolimos G, Kollias V (2010) BBN based approach for improving the software development process of an SME—a case study. J Softw Maint Evol Res Pract 22(2):121–140

Britto R, Freitas V, Mendes E, Usman M (2014) Effort estimation in global software development: a systematic literature review. Proc 9th Int Conf Global Software Eng, Shanghai, China: 135–144

Britto R, Mendes E, Börstler J (2015) An empirical investigation on effort estimation in agile global software development. Proc 10th Int Conf Global Software Eng, Ciudad Real, Spain: 38–45

Carver J (2010) Towards reporting guidelines for experimental replications: a proposal. Proc 1st Int Workshop Replic Empirical Software Eng Res. Cape Town, South Africa

Cohen J (1992) A power primer. Psychol Bull 112:155–159

Cohn M (2005) Agile estimating and planning. Prentice Hall

Conte SD, Dunsmore HE, Shen VY (1986) Software engineering metrics and models. Benjamin-Cummings

Cook RD (1977) Detection of influential observations in linear regression. Technometrics 19:15–18

Fernández-Diego M, Martínez-Gómez M, Torralba-Martínez J-M (2010) Sensitivity of results to different data quality meta-data criteria in the sample selection of projects from the ISBSG dataset. Proc 6th Int Conf Predict Models Software Eng, Timisoara, Romania: 13:1–13:9.

Forselius P (2006) Data quality criteria for Experience® data collection. STTF Oy

Foss T, Stensrud E, Kitchenham B, Myrtveit I (2003) A simulation study of the model evaluation criterion MMRE. IEEE Trans Softw Eng 29(11):985–995

Han J, Kamber M (2006) Data mining concepts and techniques. Morgan Kaufmann

Jørgensen M (2004) A review of studies on expert estimation of software development effort. J Syst Softw 70(1):37–60

Jørgensen M (2005) Practical guidelines for expert-judgment-based software effort estimation. IEEE Softw 22(3):57–63

Jørgensen M (2013) Relative estimation of software development effort: it matters with what and how you compare. IEEE Softw 30(2):74–79

Jørgensen M, Grimstad S (2008) Avoiding irrelevant and misleading information when estimating development effort. IEEE Software 25(3): 78–83

Jørgensen M, Shepperd M (2007) A systematic review of software development cost estimation studies. IEEE Trans Softw Eng 33(1):33–53

Kitchenham BA, Mendes E (2009) Why comparative effort prediction studies may be invalid. Proc 5th Int Conf Predict Models Software Eng, Vancouver, Canada: 4:1–4:5

Kitchenham BA, Pickard LM, MacDonell SG, Shepperd MJ (2001) What accuracy statistics really measure. IEE Proc - Software 148(3):81–85

Kitchenham B, Pfleeger SL, McColl B, Eagan S (2002) An empirical study of maintenance and development estimation accuracy. J Syst Softw 64(1):57–77

Kitchenham BA, Mendes E, Travassos G (2007) Cross versus within-company cost estimation studies: a systematic review. IEEE Trans Softw Eng 33(5):316–329

Kocaguneli E, Menzies T, Mendes E (2014) Transfer learning in effort estimation. Empir Softw Eng 19:1–31

Lefley M, Shepperd MJ (2003) Using genetic programming to improve software effort estimation based on general data sets. LNCS 2724. Springer, Verlag, pp 2477–2487

Li YF, Xie M, Goh TN (2009) A study of the non-linear adjustment for analogy based software cost estimation. Empir Softw Eng 14:603–643

Lokan C, Mendes E (2008) Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions. Proc 12th Int Conf Eval Assess Software Eng, Bari, Italy: 151–160

Lokan C, Mendes E (2009a) Using chronological splitting to compare cross- and single-company effort models: further investigation. Proc 32nd Austral Conf Comput Sci, Wellington, NZ: 47–54

Lokan C, Mendes E (2009b) Applying moving windows to software effort estimation. Proc 3rd Int Symp Empirical Software Eng Measure, Lake Buena Vista, Florida, USA: 111–122

Lokan C, Mendes E (2012) Investigating the use of duration-based moving windows to improve software effort estimation. Proc 19th Asia-Pacific Software Eng Conf, Hong Kong

Lokan C, Mendes E (2014) Investigating the use of duration-based moving windows to improve software effort prediction: a replicated study. Inf Softw Technol 56(9):1063–1075

Lopez-Martin C, Isaza C, Chavoya A (2012) Software development effort prediction of industrial projects applying a general regression neural network. Empir Softw Eng 17(6):738–756

MacDonell SG, Shepperd MG (2003) Using prior-phase effort records for re-estimation during software projects. Proc 9th IEEE Int Symp Software Metrics, Sydney, Australia

MacDonell SG, Shepperd MJ (2010) Data accumulation and software effort prediction. Proceedings of 4th International Symposium on Empirical Software Engineering and Measurement. Bolzano-Bozen, Italy

Mäntylä MV, Lassenius C, Vanhanen J (2010) Rethinking replication in software engineering: can we see the forest for the trees?. Proc 1st Int Workshop Replic Empirical Software Eng Res, Cape Town, South Africa

Maxwell K (2002) Applied statistics for software managers. Software Quality Institute Series, Prentice Hall

Mendes E (2014) Practitioner's Knowledge representation—a pathway to improve software effort estimation. Springer, ISBN 978-3-642-54156-8

Mendes, E. and C. Lokan. 2009. Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions: a replicated study. Proc 13th Int Conf Eval Assess Software Eng, Durham, UK

Mendes E, Mosley N (2008) Bayesian network models for web effort prediction: a comparative study. IEEE Trans Softw Eng 34(6):723–737

Menzies T, Krishna R, Pryor D (2016) The promise repository of empirical software engineering data; http://openscience.us/repo. North Carolina State University, Department of Computer Science

Minku LL, Yao X (2012a) Can cross-company data improve performance in software effort estimation?. Proc 8th Int Conf Predict Models Software Eng, Lund, Sweden: 69–78

Minku LL, Yao X (2012b) Using unreliable data for creating more reliable online learners. International Joint Conference on Neural Networks, Brisbane, pp 1–8

Minku LL, Sarro F, Mendes E, Ferrucci F (2015) How to make best use of cross-company data for Web effort estimation?. Proc 9th Int Symp Empirical Software Eng Measure, Beijing, China: 1–10

Premraj R, Shepperd MJ, Kitchenham BA, Forselius P (2005) An empirical analysis of software productivity over time. Proc 11th Int Symp Software Metrics, Como, Italy

Schmietendorf A, Kunz M, Dumke R (2008) Effort estimation for agile software development projects. Proceedings 5th Software Measurement European Forum, Milan, pp 113–126

Shepperd MJ, MacDonell SG (2012) Evaluating prediction systems in software project estimation. Inf Softw Technol 54(8):820–827

Shull FJ, Carver JC, Vegas S, Juristo N (2008) The role of replications in empirical software engineering. Empir Softw Eng 13:211–218

Sigweni B, Shepperd MJ, Turchi T (2016) Realistic assessment of software effort estimation models. Proc 20th Int Conf Assess Eval Software Eng, Limerick, Ireland

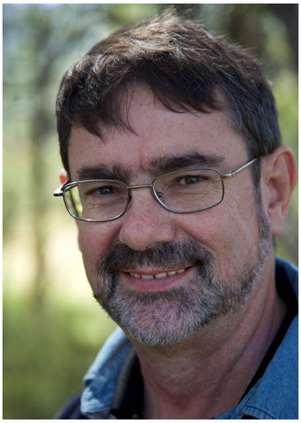
Song L, Minku LL, Yao X (2013) The impact of parameter tuning on software effort estimation using learning machines. Proc 9th Int Conf Predict Models Software Eng, Baltimore, USA: 9:1–9:10

Tabachnick BG, Fidell LS (1996) Using multivariate statistics. Harper-Collins

Tsunoda M, Amasaki S, Lokan C (2013) How to treat timing information for software effort estimation?. Proc 2013 Int Conf Software Syst Process, San Francisco, USA:10–19

Turhan B (2012) On the dataset shift problem in software engineering prediction models. Empir Softw Eng 17:62–74

Usman M, Mendes E, Weidt F, Britto R (2014) Effort estimation in agile software development: a systematic literature review. Proc 10th Int Conf Predict Models Software Eng, Turin, Italy: 82–91



Chris Lokan received the Ph.D. degree in Computer Science from the Australian National University. He is an Associate Professor at the University of New South Wales, Canberra campus. His teaching and research concentrate on software engineering. His main research interests are empirical software engineering, software effort and cost estimation, software benchmarking, complex adaptive systems, and data mining. He is a member of the ACM, and the IEEE Computer Society.

Emilia Mendes is Full Professor in Computer Science at the Blekinge Institute of Technology (Sweden), and also Distinguished Finnish Professor at the University of Oulu (Finland). She obtained her PhD in Computer Science from the University of Southampton (UK) in 1999, and then initiated her full time academic career at the Computer Science Department at the University of Auckland (NZ), where she worked for 12 years. After leaving NZ, and prior to moving to Sweden, she was Associate Professor at Zayed University (UAE) for a year. Her main research contributions were made to date in the application of machine learning techniques to Web and software effort estimation; however, she also carries out research in collaboration with colleagues and via the supervision of PhD students, in a wider spectrum of research topics (e.g. value-based decision making, software process improvement, personality in software development teams, computer science education, machine learning applied to healthcare). Overall, she has authored/co-authored over 200 research papers at international journal & conference, with seven best paper awards to date (2 at ESEM). She is the co-editor of a book (2005 - Web Engineering) and sole author of two books (2007 - Cost Estimation Techniques for Web Projects; 2014 – Practitioner's Knowledge Representation: a pathway to improve software effort estimation). Finally, she worked in the ICT industry for ten years as programmer, business analyst and project manager prior to moving to the UK in the end of 1995 to initiate her PhD studies.