The usage of ISBSG data fields in software effort estimation: A systematic mapping study

Fernando González-Ladrón-de-Guevara [a,*], Marta Fernández-Diego [a], Chris Lokan [b]

[a] Department of Business Organisation, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain.
[b] School of Engineering and Information Technology, UNSW Canberra, Northcott Drive, Canberra ACT 2600 Australia.

[*] Corresponding author: Tel.: +34 96 387 76 82, Fax: +34 96 387 97 79

E-mail addresses: fgonzal@omp.upv.es (Fernando González-Ladrón-de-Guevara), marferdi@omp.upv.es (Marta Fernández-Diego), c.lokan@adfa.edu.au (Chris Lokan)

Abstract

The International Software Benchmarking Standards Group (ISBSG) maintains a repository of data about completed software projects. A common use of the ISBSG dataset is to investigate models to estimate a software project's size, effort, duration, and cost. The aim of this paper is to determine which and to what extent variables in the ISBSG dataset have been used in software engineering to build effort estimation models. For that purpose a systematic mapping study was applied to 107 research papers, obtained after a filtering process, that were published from 2000 until the end of 2013, and which listed the independent variables used in the effort estimation models. The usage of ISBSG variables for filtering, as dependent variables, and as independent variables is described. The 20 variables (out of 71) mostly used as independent variables for effort estimation are identified and analysed in detail, with reference to the papers and types of estimation methods that used them. We propose guidelines that can help researchers make informed decisions about which ISBSG variables to select for their effort estimation models.

Keywords

Systematic mapping study

Software engineering

ISBSG data field

Software effort estimation

Missing value

1. Introduction

The International Software Benchmarking Standards Group (ISBSG)[1] maintains a Development & Enhancement Repository (referred to hereafter as the "ISBSG dataset"). This is a large public database about completed software development and enhancement projects. The goal of ISBSG is to help organisations to improve their IT resource management, by performing their own analyses, estimations, comparisons, or benchmarking through the use of this dataset (ISBSG, 2009a). The dataset contains data from a wide range of countries, organisations[2], application types, and development types. A demographic summary of the dataset is presented in (ISBSG, 2009b).

1 www.isbsg.org

The ISBSG dataset has grown over 20 years, in both the number of projects and the number of data fields collected for each project. The most recent release (Release 13) contains data on 6,760 projects, with 118 data fields for each project. The ISBSG dataset is also available to researchers, and has been used in many studies relating to software effort estimation. Compared to other public datasets available to researchers, such as the PROMISE repository of datasets (Menzies et al., 2015), the ISBSG dataset is very much larger and contains more recent data.

The ISBSG dataset offers a wealth of information about completed software projects, regarding practices, tools, and methodologies, accompanied by process and product data, to be used in benchmarking, monitoring, quality control, and performance management purposes during the software development process [S66]. However, there are some issues that need to be considered when using it (Fernández-Diego and González-Ladrón-de-Guevara, 2014). First, It is not a data sample deliberately chosen to be representative of the IT industry ([S73, S84], (Song et al., 2008)). Contributors choose which of their projects to submit to ISBSG, which may introduce bias. Furthermore, there is no data from incomplete projects, so failed or abandoned projects are not represented. Second, the ISBSG dataset suffers from heterogeneity, i.e., the combining of data from heterogeneous sources (Stensrud et al., 2002). Therefore, a data selection, transformation, and preparation process is required before any data analysis (Bibi et al., 2008). Third, many of the 118 data fields (henceforth called variables) in the dataset are not relevant for software effort estimation, and many have a large number of missing values [S95]. In most research studies, requirements about the quality and completeness of the data lead to substantial proportions of the available data being discarded. Deng and MacDonell (2008) describe a methodical approach to the preprocessing of data in order to maximise the retention of data to improve the robustness of software effort estimation models.

When using datasets with many variables, the selection of relevant project variables for software effort estimation purposes is important. For example, estimation by analogy requires the identification of the main project variables that affect effort; the optimal subset of variables is selected based on prediction accuracy and the relation between input and output [S64]. Using all available variables in the estimation process is not effective: it may reduce overall estimation accuracy, and therefore increase the severity of project risk, because some variables may be irrelevant or redundant, or because of model complexity and additional noise [S64]. As models include more variables they become gradually more difficult to build, and more unstable to use (Deng and MacDonell, 2008). To overcome these issues, algorithms known as Feature Subset Selection try to find a subset of variables that provide models with similar or better accuracy than using all available variables. This challenging process is mostly based on expert knowledge. Search algorithms may help to identify the most relevant project variables (Song et al., 2008; Dolado et al., 2007), but this can involve a large amount of computing power and time, and may still only provide poor evidence to support the identified variables [S56].

The problem with selecting the most significant variables for effort estimation from the ISBSG dataset is that there is no agreement yet about which variables should be selected, or how to select them. This paper aims to help fill that gap in knowledge. As reported in (ISBSG, 2009a), it is important that users of the ISBSG dataset – both researchers and practitioners – have a sound knowledge of the dataset prior to analysing or using it. This includes knowing the nature of each variable, including the range and type of information it conveys, and its shortcomings; how the variables are grouped conceptually, and the relationships between them; and finally, how and why they are used in software effort estimation models. With that understanding, researchers can conduct sound and repeatable research using the ISBSG dataset. With the same understanding, and knowledge of their own organisation's goals, practices, software development environment, past projects, and a project to be estimated, practitioners can make effective use of the ISBSG dataset.

The ISBSG dataset is by far the largest available for research in effort estimation, and it has now been used in a large number of studies. The main aim of the paper is to map out research practice when

2 British spelling is used in this paper (e.g. "organisation" rather than "organization"), for consistency with how ISBSG spells its variable names.

using ISBSG variables for research into effort estimation. It emerges that there is great variation in how ISBSG data has been used, and in many papers there is also a lack of clarity in how the data was prepared and used, and even what data was used. This makes it difficult to compare results between studies, and to replicate studies. Thus we also compare research practice with ISBSG's own recommendations for using ISBSG data for effort estimation, to provide some insights to guide the selection of variables in future effort estimation research, and we make some suggestions about how the usage of ISBSG data should be documented.

Some preliminary results, considering the most used ISBSG variables and their characteristics, were reported in (González-Ladrón-de-Guevara and Fernández-Diego, 2014). The present paper extends the previous paper, mainly regarding the usage of independent variables in effort estimation models. In addition, the use of variables is considered from two extra perspectives: data filtering, and dependent variables. Further, the paper presents some analysis of publication venues, and trends over time (two common aspects of systematic mapping studies (Petersen et al., 2015)), and analyses which independent variables tend to be used more with different effort estimation methods.

The rest of this paper is organised as follows. Section 2 describes the mapping process. Section 3 reports results of the mapping. Section 4 discusses the principal findings, limitations of the study, and the implications for research and practice. Section 5 outlines the main conclusions and directions for future research.

2. Methodology

Systematic mapping studies are a type of systematic literature reviews that aim to collect and classify research papers related to a specific topic (Kitchenham and Charters, 2007; Petersen et al., 2008, 2015; Acuña et al., 2012). These studies require a rigorous searching process as well as detailed inclusion and exclusion criteria that are clearly defined in the research protocol and presented in the results report (Budgen et al., 2008). Systematic mapping studies generally have broader research questions than systematic reviews and are primarily concerned with structuring a research area (Petersen et al., 2015). Consequently, the data extraction process for mapping studies is also much broader. This process can be considered as a classification or categorisation stage. Furthermore, the analysis process in a mapping study intends to summarise the data using descriptive statistical parameters and charts instead of performing meta-analysis and narrative synthesis (Kitchenham and Charters, 2007).

This section provides an overview of the steps involved in the process of this systematic mapping study, following Petersen et al. (2008, 2015) including the formulation of the research questions, the search strategy for primary studies, the inclusion and exclusion criteria, and the data collection process.

2.1. Research questions

The primary goal addressed by this study is to analyse the use of ISBSG variables in effort estimation research. For this, three research questions (RQ) were considered:

- RQ1: What are the most used ISBSG variables in effort estimation research?
- RQ2: What are the characteristics of these variables (their meaning, range and distribution of values, extent of missing data, relationship to other variables)?
- RQ3: How, and to what extent, have ISBSG variables been used as independent variables to build effort estimation methods?

2.2. Strategy to search for primary studies

The search process for selecting studies follows (Fernández-Diego and González-Ladrón-de-Guevara, 2014), but the time scope was extended to December 2013.

The following four bibliographic databases were used to make a general search for relevant papers in journals and conference proceedings: IEEE Xplore, ACM Digital Library, ScienceDirect, and Web of Science. These databases were selected because they are the major search engines and digital libraries most frequently used in systematic literature reviews performed by the software engineering community. In (Zhang et al., 2011), eleven search engines used more than once in systematic literature reviews for searching relevant studies in software engineering were ranked in the order of their frequencies. Among them, IEEE Xplore, ACM Digital Library, Science Direct, and Web of Science occupied the first, second, third, and fourth position respectively.

The search was completed by February 2014 and includes conference papers and journal articles published up until December 2013. At that time the most recent version of the ISBSG dataset was Release 12, but most published research used Release 11 or earlier (for this reason, the statistics quoted in this paper are based on Release 11).

To make the search as inclusive as possible, no logical operators were used and the unique search term (“ISBSG”) was applied not only to the title and abstract of the paper, but also to the body of the text. The search, however, within the full document record was not possible in papers indexed in the Web of Science. To limit the impact of this fact on the consistency of the search process, an additional search was performed within the journals that resulted from the general search in the Web of Science after checking they were not already indexed in the three other databases. The journals in question are: Empirical Software Engineering, the Software Quality Journal, the International Journal of Software Engineering and Knowledge Engineering, and the Journal of Software Maintenance and Evolution. This search was possible using their publisher’s search engine directly. A similar workaround was unfeasible for the conference proceedings that resulted from the search in the Web of Science.

This process resulted in 177, 79, 64, and 94 results respectively. Four spurious items from ACM and three from SD were eliminated since only conference papers and journal articles were considered, and thereby, 407 references were obtained before deleting duplicate records. There were 48 duplicate articles and two articles in triplicate from the four databases, resulting in 355 references. Moreover, two false records were detected from the set of papers and three more were excluded since they were not written in English. In the end, 350 useable potential primary studies remained. The process for selecting primary studies is summarised in Figure 1.

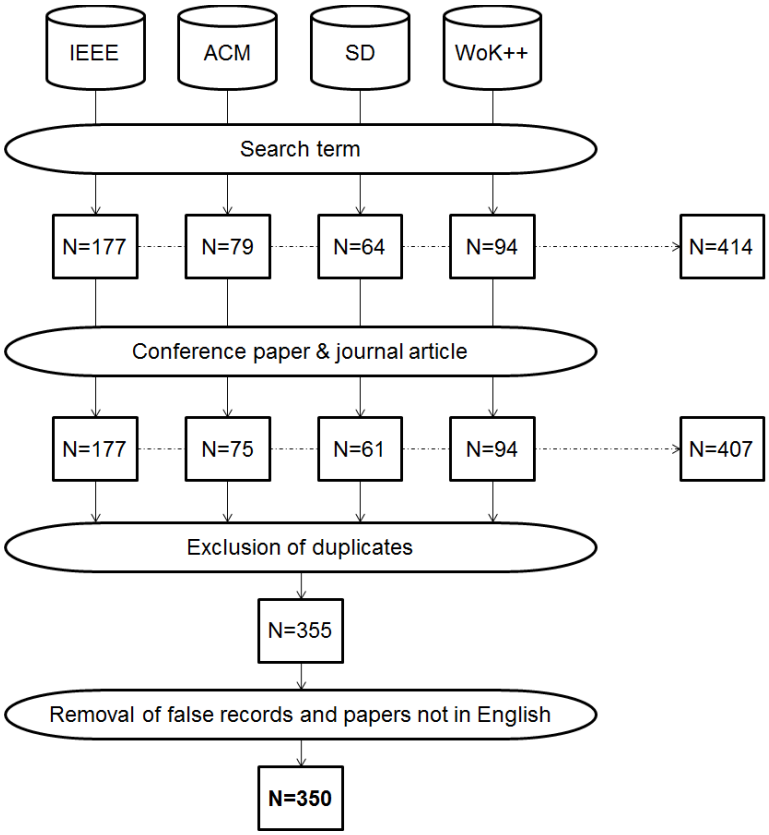

Figure 1. Search process for the selection of studies.

2.3. Inclusion and exclusion criteria

Inclusion and exclusion criteria are required to assess each potential primary study. In order to undertake this process, four filters were defined. The first filter (F1) was defined to verify that the references were really related to the usage of the ISBSG dataset. In this way a reference should be filtered out through F1 in the following situations:

- when “ISBSG” only appeared in the list of references;
- when ISBSG was mentioned as an example of a dataset or when the publication simply referred to ISBSG;
- in cases where the ISBSG dataset was not used in the identified paper, although it was mentioned that the authors of the paper intended to use the ISBSG dataset.

To summarise, whenever the ISBSG dataset is used in any way, not only as a data source, the article passed filter F1. The application of F1 resulted in the exclusion of 130 papers.

A second filter (F2) was applied to the remaining 220 papers, to verify whether data from the ISBSG dataset was used for the research undertaken in the paper. Since the use of the ISBSG dataset as a primary data source is our interest, only those papers were retained by F2. Most of the papers that were discarded at this stage did not perform any experimental work on the ISBSG dataset; rather, for comparative purposes, authors took advantage of ISBSG statistics describing the dataset variables or made use of results from previous works that utilized the dataset. The application of F2 resulted in the exclusion of 53 papers.

A third filter (F3) was defined to exclude papers that were not concerned with estimating effort or productivity. This resulted in the exclusion of 30 papers.

Finally, among the remaining 137 papers, only 107 mention the independent variables that are used in their estimation models (filter F4). This is the final set of papers that will be analysed in this study. These papers are listed in Appendix A.

The filtering process was performed independently by the first two authors. Less than 5% of the papers were debatable, mostly from applying F2. All conflicts were resolved via discussion and in some cases with the support of the third author.

Figure 2 presents the filtering procedure, which reduced the initial set of 350 papers to the final subset of 107 papers.

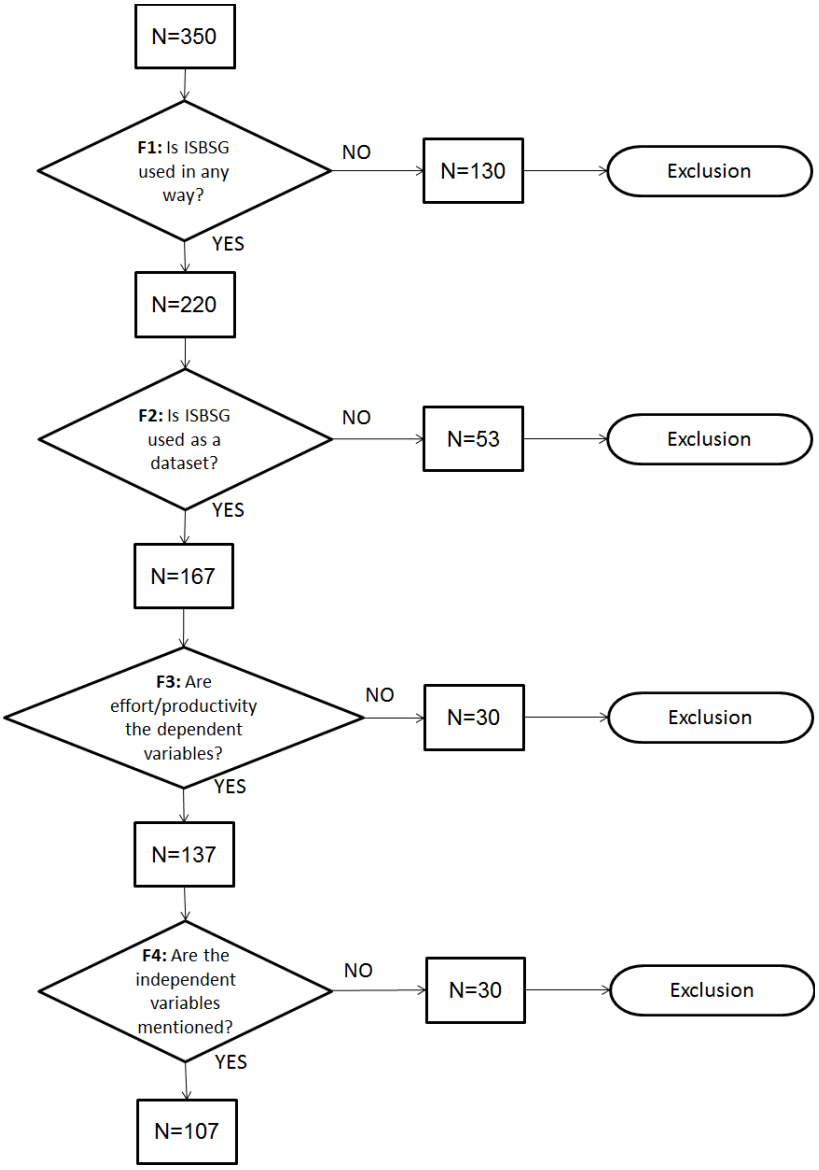

Figure 2. Filtering process.

2.4. Data collection

The 107 identified primary studies were reviewed using a data extraction form (Elberzhager et al., 2012). The resulting database collects general paper information (title, author(s), venue of publication, year, keywords, etc.) and data addressing the research questions. Specifically, for each paper data has been collected concerning all variables used for filtering, all variables used as dependent or independent variables in the estimation models, as well as several other related fields required for the subsequent analysis. Overall, our database includes 110 fields for each paper.

3. Results

The results of the systematic mapping study are presented in this section, following each of the three research questions.

Before doing that, we consider the different publication venues. The analysis of the specific venues of publication is common in systematic mapping studies (Elberzhager et al., 2012; Petersen et al., 2015).

Among the 107 papers, 53 are journal articles. Four journals, Information and Software Technology, Journal of Systems and Software, Empirical Software Engineering, and Software Quality Journal account for 67% of the journal papers, with 13, 9, 8, and 6 papers respectively.

The other 54 publications are conference papers. The most relevant conferences in terms of number of papers published are PROMISE (International Conference on Predictive Models in Software Engineering), IWSM-MENSURA (Joint Conference of International Workshop on Software Measurement and International Conference on Software Process and Product Measurement), ESEM (International Symposium on Empirical Software Engineering and Measurement), and METRICS (International Software Metrics Symposium) with 8, 5, 5, and 4 papers respectively. These results are consistent with (Fernández-Diego and González-Ladrón-de-Guevara, 2014), except for IWSM-MENSURA which recently rises to the top of the list. The other 32 papers are spread across 27 conferences, with no more than two papers each.

Considering the papers published in journals and conferences over time, two periods can be distinguished in Figure 3. The number of papers before 2005 is small (6) with a minimum in 2004 (0 papers). Therefore, this period of time can be considered as an introductory period. In fact, the first papers of the subset were published in 2000, when Release 6 of the ISBSG dataset was delivered. Nevertheless, a few papers used previous releases that were also available to researchers. Then, there is an increase in the number of papers starting in 2005, reaching a peak in the year 2008 (19) followed by a decreasing period (2009-2011) with a recent increase (2012-2013). The proportion among journals and conferences is quite similar except for year 2009 (69% of papers were published in conferences) and 2013 (64% of papers published in journals).

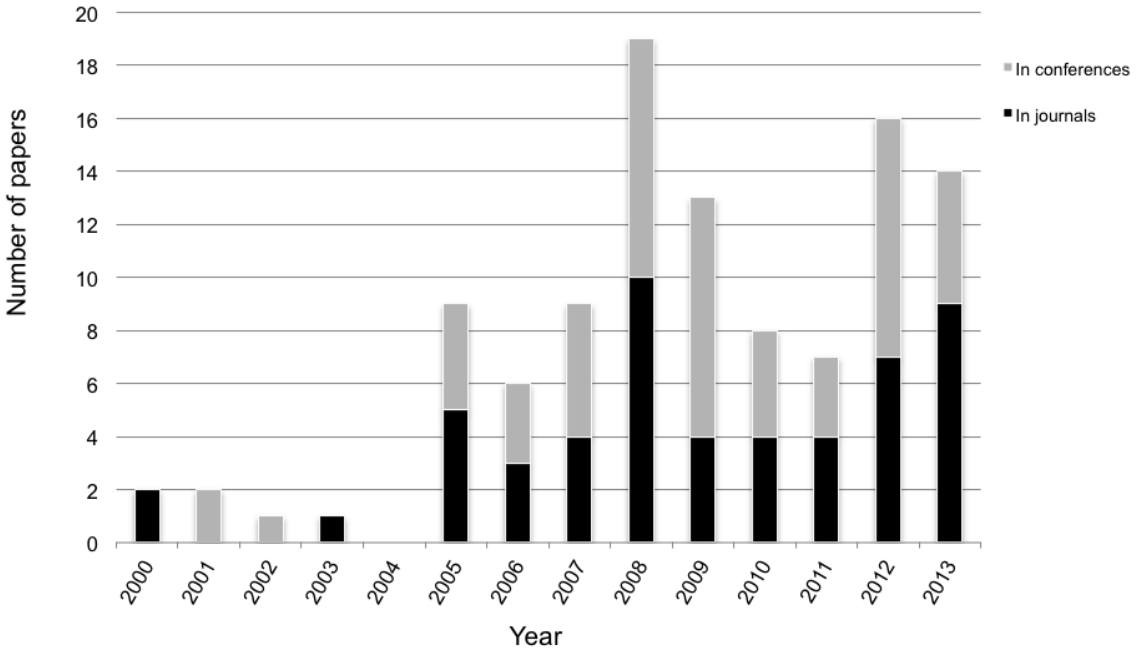

Figure 3. Number of papers published in journals and conferences per year.

3.1. RQ1: What are the most used ISBSG variables in effort estimation research?

ISBSG variables are used for three different purposes, with little overlap between which variables are used for which purpose:

Filtering variables: a filtering stage is needed to select a suitable set of projects for any given analysis. The variables used for this purpose tend not to be used in the analysis itself.

Dependent variables: in research on effort estimation it is no surprise that effort is usually the dependent variable in effort estimation models. However, it is not as simple as that, because three different effort variables are included in the ISBSG dataset. It is important to understand the difference between them, and for researchers to report which one they use.

Independent variables: a large range of the 118 variables in the ISBSG dataset are potentially viewed as effort drivers. Analysing their use is the main aim of this paper.

Descriptions of each ISBSG variable, and the values that each variable can take, are in (ISBSG, 2009d).

3.1.1. Critical variables: size and effort

Before looking in detail at the usage of ISBSG variables for each of the three purposes, we look at two critical variables – size and effort – that are relevant for all three purposes. Each is represented in the ISBSG dataset using multiple variables, which can mean different things.

Size

Size is relevant for filtering, as described in Section 3.1.2; as part of the definition of the dependent variable in some studies, as described in Section 3.1.3; and as an independent variable in most studies, as described in Section 3.1.4.

In software engineering datasets, size can refer to the physical size of the program, measured in lines of code (LOC); or to the functional size of the problem, measured in Function Points. The ISBSG dataset has a variable called Functional Size, and a variable called Lines of Code. In Release 11, 6.1% of the projects report both, 3.6% report LOC but not Functional Size, 90.0% report Functional Size but not LOC, and 0.3% report neither. (In Release 13 the figures were scarcely different, being 5.0%, 2.7%, 92.0% and 0.3% respectively.)

ISBSG's preferred measure of size is Functional Size. Three things need to be considered in order to understand this variable: which approach to measuring functional size was used for the measurement? Does it represent unadjusted function points (UFP) or adjusted function points (AFP)? Which version of the ISBSG dataset was used?

- *Which approach?* The ISBSG variable Count Approach records the approach used to measure size. For almost all projects its value is COSMIC, IFPUG 4+, FiSMA, NESMA, Mark II, IFPUG old, or LOC. COSMIC (ISO, 2011), IFPUG (ISO, 2009), FiSMA (ISO, 2008), NESMA (ISO, 2005) and Mark II (ISO, 2002) refer to the five approaches to Functional Size Measurement that have been approved as international standards. These methods are generally not comparable: if the functional size of the same software specification is determined using different approaches, the results will generally be different. IFPUG 4+ (meaning the use of version 4.0 or later of IFPUG's Counting Practices Manual) is distinguished from older versions of IFPUG because they are not comparable. IFPUG 4+ renders older versions of IFPUG obsolete, and COSMIC renders Mark II obsolete. Since FSM approaches are not comparable (the only exception is that IFPUG 4.2 or later and NESMA are essentially the same (NESMA, 2014)), researchers should not analyse together projects that were sized with different count approaches.
- *UFP or AFP?* Except for COSMIC, all approaches to measuring Functional Size involve two phases. The first determines UFP, considering base functional components of the software, and the second applies a Value Adjustment Factor (VAF) to the UFP value to take into account general system characteristics such as complexity. The characteristics used to compute the VAF vary between count approaches. It is up to researchers to decide whether to use UFP or AFP. UFP is preferred by many researchers for theoretical reasons (Abran and Robillard, 1996) and practical reasons [S20], and has been ISBSG's preference since Release 9. On the other hand, more data is available in the ISBSG dataset if AFP is used. Since Functional Size could be either UFP or AFP, and they vary in what they include, it is important for researchers to report which they use.
- *Which release?* Up to and including Release 8, the ISBSG dataset included the variables Function Points (representing AFP), VAF, and UFP. Since Release 9, released in 2004, it has included the variables Functional Size (representing UFP), VAF, and AFP. Thus researchers who used the generic size variable Function Points or Functional Size used AFP up to Release 8, but UFP since Release 9. For researchers it is important to specify which Release they use; and as readers it is important to be aware of how definitions have changed, when reading papers that conducted research using different Releases of the ISBSG dataset.

Effort

Effort is relevant for filtering, as described in Section 3.1.2; and as a dependent variable in most studies, as described in Section 3.1.3.

Three effort variables are recorded in the ISBSG dataset. The fundamental variable is Summary Work Effort (SWE), measured in staff-hours. It is the total effort for the project, as reported by the contributing organisation. The other two variables are added by ISBSG, to take into account differences in whose effort is included in SWE, and which life cycle phases are included in SWE.

- *Which Resource Level?* Projects report effort involving different participants. This is indicated by the variable Resource Level (RL), which can take a value from 1 to 4. A value of 1 means that effort is reported for the development team only, Levels 2 and 3 add effort for development team support and computer operations involvement, and Level 4 adds effort for end users and clients. For most projects, RL is 1. Of those with higher RL values, some only give an overall total, while others break the effort down by level; thus there are many projects with RL greater than 1, but within which the development team effort (Level 1 effort) is known separately.
- *Which development life cycle phases?* Some projects do not report effort for all development life cycle phases. Comparisons are difficult between projects that include different subsets of life cycle phases. With Release 8, in 2003, ISBSG introduced effort normalisation, whereby ISBSG estimates the amount to be added to SWE to account for any "missing" phases. ISBSG argues that although Normalised Effort values are estimates, and hence their accuracy is less certain than SWE, the comparability of Normalised Effort values is better because all life cycle phases are included for every project (ISBSG, 2009a).

Two variables that include normalisation are included in the ISBSG dataset. Normalised Effort is ISBSG's estimate of the total effort when any "missing" phases are added. Normalised Level 1 Effort is the normalised effort for the development team only.

Considering the values of the three effort variables, we observe that projects fall into six categories:

- SWE covers the whole life cycle (so Normalised Effort is the same as SWE); and RL is 1 (effort is recorded for the development team only). For such projects, the three effort values are the same.
- SWE covers the whole life cycle; RL is greater than 1 (total effort is known for more than just the development team), but the development team's component of total effort is known. SWE and Normalised Effort are the same, but Normalised Level 1 Effort is lower.
- SWE covers the whole life cycle; RL is greater than 1, and the development team's component of total effort is not known. SWE and Normalised Effort are the same, but Normalised Level 1 Effort is unknown.
- SWE does not cover the whole life cycle (normalisation "fills in the gaps", so Normalised Effort is greater than SWE); and RL is 1. Normalised Level 1 Effort and Normalised Effort are the same, but SWE is lower.
- SWE does not cover the whole life cycle; RL is greater than 1, but the development team's component of total effort is known. Normalisation increases SWE, to account for missing phases; separately, only considering the development team's effort reduces SWE by excluding the effort of other contributors. The three effort values can all be different.
- SWE does not cover the whole life cycle; RL is greater than 1, and the development team's component of total effort is not known. SWE and Normalised Effort differ, while Normalised Level 1 Effort is unknown.

All six categories are represented in the repository, with the first being the most common (55% of the projects in Release 11, 65% in Release 13).

The important point for researchers is that ISBSG reports three different effort variables, which can mean different things, and since 2003 there has been no single generic "effort" variable. It is essential that researchers report which specific variable they choose to represent effort.

3.1.2. Which ISBSG variables are used in the filtering process?

As the ISBSG dataset is quite heterogeneous, filtering is needed before any analysis to select a suitable subset of projects for study. The large number of projects in the ISBSG dataset makes it possible to select subsets using several requirements ([S2, S8, S73], (Li et al., 2007)).

The primary reason for filtering projects is to ensure high quality data. When dealing with variables of different units and scales, it is also important that the data subset has integrity (ISBSG, 2009a). That is to say: measurements for the key variables (size and effort) should be defined the same way, and applied to the same things. To this end, researchers using the ISBSG dataset consider one or more of three things:

- *Data quality:* ISBSG evaluates the quality of the data about a project. Two variables are used for this purpose. Both are graded A (best) to D (worst). The grades are assigned by ISBSG's quality reviewers. Data Quality Rating denotes the reliability of the recorded data. In particular, the reviewers base the classification on the completeness of the data (Liebchen and Shepperd, 2008), and on omissions and inconsistencies in the data that might affect reliability (Deng and MacDonell, 2008). Similarly, the Unadjusted Function Point Rating denotes whether the unadjusted function point count was trustworthy, with nothing being identified that could affect its integrity. ISBSG advises that quality ratings of A or B are acceptable, but projects with lower quality ratings should be excluded from analysis (ISBSG, 2009a).
- *Comparable definitions for size and effort:* As discussed in Section 3.1.1, size measures obtained with different approaches are generally not comparable. The Count Approach variable can be used to select projects that were sized using a consistent approach. Most often, projects are only retained for analysis if their Functional Size is measured with IFPUG function points (version 4.0 or later). Also as discussed in Section 3.1.1, it is common to retain projects only if Resource Level is 1, or only if the development team's effort is known separately from any other recorded effort.
- *Confidence in the effort values:* Several researchers choose not to consider projects where the Normalised Effort is significantly different (or different at all) from SWE. Also, some researchers only consider projects where the effort Recording Method is "Staff hours (recorded)".

Filtering for these purposes generally uses variables that are not used for any other purpose.

Of the 107 papers analysed in this systematic mapping study, 90 (85%) described their filtering process, 16 (15%) filtered the projects but did not explain their process, and one did no filtering. Any paper may use several variables for filtering.

Table 1 shows the variables that are used in filtering projects. The first column lists the three reasons for filtering and the second column the variables used for each purpose. The other columns show the percentage of papers (out of the 90 papers that described filtering) that have used each pair of these variables. For example, in the first row '58.9%' indicates the percentage of papers (out of 90) using at least the variables Data Quality Rating and Count Approach in their filtering. This matrix is symmetric and its main diagonal conveys the percentage of papers that have used each variable, alone or combined with others for filtering purposes.

Table 1

ISBSG variables used in filtering projects.

Reason for filtering	**Variable**	Data Quality Rating	UFP Rating	Count Approach	Resource Level	Normalised Effort / SWE	Recording Method
Data quality	Data Quality Rating	**82.2%**	14.4%	58.9%	34.4%	33.3%	8.9%

	UFP Rating	14.4%	**14.4%**	13.3%	5.6%	4.4%	1.1%
Comparable definitions	Count Approach	58.9%	13.3%	**64.4%**	32.2%	31.1%	11.1%
	Resource Level	34.4%	5.6%	32.2%	**35.6%**	20.0%	7.8%
Confidence in effort	Normalised Effort / SWE	33.3%	4.4%	31.1%	20.0%	**33.3%**	0.0%
	Recording Method	8.9%	1.1%	11.1%	7.8%	0.0%	**12.2%**

Regarding data quality (Fernández-Diego et al., 2010), 74 papers took it into account (the 13 considering UFP Rating are a subset of the 74 that considered Data Quality Rating). Thus, 82.2% of the papers considered data quality, but many researchers (17.8% of the papers) have used ISBSG data without indicating that they considered its quality. 64.4% of the papers reported that they considered whether size values were comparable (Count Approach) when selecting projects for analysis, but 35.6% did not. Barely over one third of the papers (35.6%) reported that they considered the comparability of effort values (Resource Level). Finally, note that the most used filtering variables, Data Quality Rating and Count Approach, were used together in only 58.9% of the papers that described the filtering process.

Projects may also be filtered for other reasons, based on the availability or values of particular variables. These tend to be dependent or independent variables in the study; which ones they are depends on the interest of the researcher.

- *Outliers:* several researchers eliminate projects for which the size, effort, or PDR (effort/size) values are identified as outliers. An outlier can be defined as a data point that appears to be inconsistent with the rest of the dataset. Datasets typically contain outliers that can degrade the data quality [S21, S72]. For this reason, the elimination of outliers is appropriate for reliable and accurate software effort estimation based on past projects. Outliers are removed in 36% of the 107 papers. Four papers [S21, S22, S72, S75] focus on the problem of outliers. Several methods are used to identify outliers; a common approach is to use Cook's distance [S13, S28, S30, S77, S95].
- *Missing data:* a problem when producing estimation models from historical data is the existence of missing values, occurring when no data is stored for a variable in a given project. In the ISBSG dataset, many variables have missing values for more than 40% of the projects, substantially reducing the ability to use data to construct effective prediction systems and potentially leading to biased and inaccurate predictive models ([S2, S12, S29, S31, S36], (Song et al., 2008)). Until recently the usual approach has been listwise deletion, excluding projects that have missing data in one or more variables [S73]. This approach is used in 35% of the 107 papers. Missing data can also be used for column pruning; for example, [S13, S103] remove columns with more than 40% of values missing, [S44] removes columns with more than 25% of values missing. Some recent studies present a progressive awareness of missing data treatment (Twala et al., 2005). The most common approach to imputing missing data is to use k-nearest neighbours (k-NN) imputation, which fills in missing data by taking values from other observations in the same dataset (Song et al., 2008). This approach is used in 14% of the 107 papers.
- *Project type:* researchers may be interested in particular types of projects. This might mean focusing on projects from a particular domain, so only projects of that type are retained. Or it might mean that projects are grouped into subsets according to the value of a particular variable (e.g. new development vs. enhancement [S57]), so projects lacking data for that variable are excluded. 43% of 107 papers exclude projects that do not fall within a specific domain of interest, or for which a particular relevant variable is missing. Which variables are used for this filtering purpose depends on the interest of the researcher. Most often they include the development type (e.g. [S3, S19] only considered new developments), organisation type (e.g. [S21] only considered banking projects), application type (e.g. [S9] was only interested in

embedded software), or implementation date (e.g. [S57, S100] excluded projects implemented before the year 2000). Some papers filtered projects by organisation code, to compare the accuracy of estimation models built using within-company and cross-company data [S13, S35, S43, S77, S102, S104][3]. Some contextual factors that researchers and practitioners may wish to use to select homogeneous sets of projects most relevant to their interest (e.g. country, organisation size, organisation culture, team culture) are not available in the ISBSG dataset; this is a limitation of any dataset that must ensure confidentiality to the providers of data.

Trends over time

Figure 4 presents trends over time in the usage of the six filtering variables of Table 1, after the initial introductory period that was noted in Figure 3. Five-year moving averages are presented (except for 2005 to 2008, which present cumulative averages), to smooth the trend.

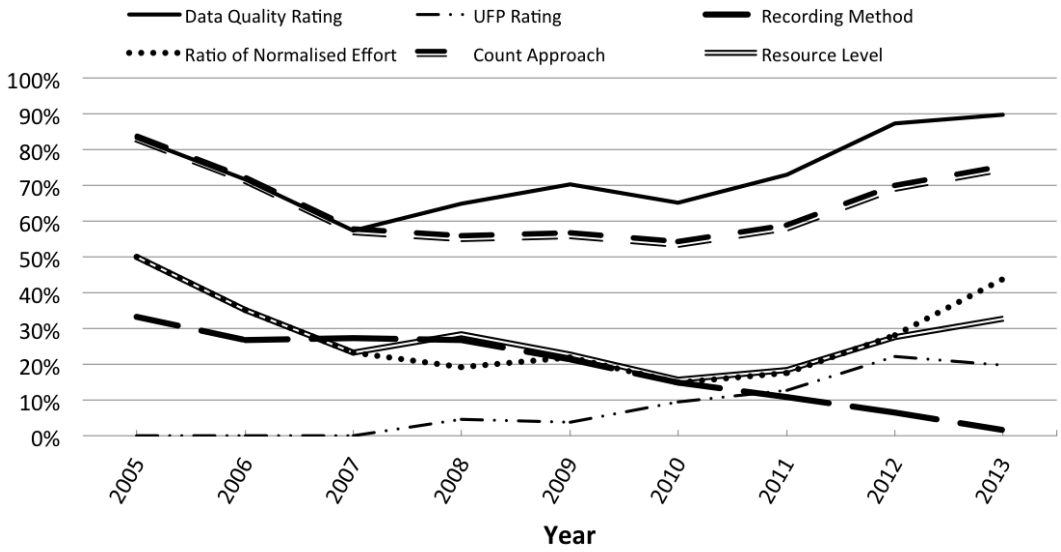

Figure 4. Five-year average evolution of the relative presence of the filtering variables.

Data Quality Rating presents a rising trend since 2007, and has been used in the filtering process for all papers since 2011. Also referring to quality, UFP rating was first used in 2007 and thereafter shows a sustained rise. The second most used variable is Count Approach, used as often as Data Quality Rating in the beginning of the period (2005-2007), and also increasing its participation thereafter but less so than Data Quality Rating. The variable Recording Method decreases until its virtual disappearance in 2013.

3.1.3. Which ISBSG variables are used as dependent variables in effort estimation research?

3 The organisation code variable is not normally made available by ISBSG. Access to that information requires special application to ISBSG.

Table 2 shows how often each variable has been used as the dependent variable in effort estimation models. Papers using Release 8 or later, Release 7 or earlier, whose Release is not stated, or which used the Maintenance and Support repository, are tabulated separately.

Table 2

ISBSG variables used as dependent variables.

Variable	R8 or later	R7 or earlier	Release not stated	Maintenance & Support	Total
Summary Work Effort (SWE)	30	20	3		**53**
Normalised Effort	13				**13**
Normalised Level 1 Effort	9				**9**
Effort (no further information given)	19		1		**20**
Productivity (FP/SWE)		5			**5**
Project Delivery Rate (SWE/FP)	4				**4**
Effort for single life cycle phases	2				**2**
Maintenance and support effort				1	**1**
Total	**77**	**25**	**4**	**1**	**107**

Some papers state which version of effort they use as their dependent variable. Of those that do not, their dependent variable can sometimes be inferred from the description of the preliminary data analysis, but there are several for which it cannot; these are included in the row “Effort (no further information given)”. In 95 of the 107 papers (89%), one of the three effort variables described in section 3.1.1 (SWE, Normalised Effort, and Normalised Level 1 Effort) is the dependent variable. The remaining 12 papers constitute the last four rows of Table 2.

For papers using Release 7 or earlier (25), there is no ambiguity: “effort” meant SWE. This is the case for 20 papers, while the other five used productivity as the dependent variable.

For papers using Release 8 or later (77), which version of “effort” is used as the dependent variable could be ambiguous, and should be made explicit. Of these, Summary Work Effort is stated to be the dependent variable in 17 papers (out of 30 papers that either explicit or implicitly used SWE); Normalised Effort is stated to be the dependent variable in 13 papers.

The effort variable is not stated and cannot be inferred in 19 papers, while it can be inferred in 22 papers. Among these 22 papers, 13 only used projects for which Normalised Effort and SWE were the same and Resource level = 1. This filtering approach means the three effort values were the same, and any of the three effort variables could be regarded as the dependent variable. However, since no normalisation is required in these cases, we have considered them to use SWE as the dependent variable. Therefore, the number of papers using Release 8 or later and SWE sums to 30. On the other hand, eight papers out of the 22 where the dependent variable can be inferred only used projects for which Normalised Effort and SWE were the same, and the development team’s effort was known; since these papers also stated that only the development team’s effort was used, Normalised Level 1 Effort was effectively the dependent variable. One other [S6] used Normalised Effort, allowing it to differ from SWE by up to a small amount, but required Resource Level to be 1, so Normalised Effort Level 1 was effectively the dependent variable.

In the remaining 19 papers using Release 8 or later, the dependent variable was “effort” but no further information was given (the specific effort variable was not stated, and cannot be inferred from the description of filtering). Among them, there are four papers [S41, S56, S57, S105] that use both effort

and duration as dependent variables. Finally, there is also a paper whose release cannot be inferred and in which the effort variable was not stated [S101]. This means that 20 papers (19%) are not repeatable, because their dependent variable is ambiguous or unknown.

No study has investigated more than one of SWE, Normalised Effort, and Normalised Level 1 Effort in the same paper, for example to see whether accuracy statistics change if the effort variable is changed from one of them to another.

Of the 12 projects that did not use some version of effort as the dependent variable, 9 estimated productivity (size/effort) or its reciprocal, Project Delivery Rate (effort/size). This, when also given knowledge of project size, is equivalent to estimating effort. The same considerations apply to these papers as to the others: does the estimate cover all life cycle phases; does it cover just the development team, or does it include other project participants as well? This should be made clear by researchers. In principle, productivity or PDR could be calculated in many ways, with effort being based on 3 possible variables, and size being based on several Count Approaches and either UFP or AFP. Of the 9 papers, 5 used IFPUG AFP/SWE as the dependent variable ([S37, S38, S39, S40, S91], all from the same research group); one used Normalised Level 1 Effort / IFPUG UFP as the dependent variable; and the other three do not define their dependent variable in detail.

The other three papers addressed quite different questions. Two [S47, S94] focused not on estimating total project effort, but rather the effort in individual phases of a project, using data from earlier phases in a project to estimate the effort of later phases in the same project. The other [S50] analysed ISBSG's other data repository, its Maintenance and Support repository.-Maintenance effort and support effort were both used as dependent variables in that study.

Trends over time

Figure 5 presents the evolution of the three effort variables SWE, Normalised Effort, and Normalised Level 1 Effort compared to the total number of papers published along the years.

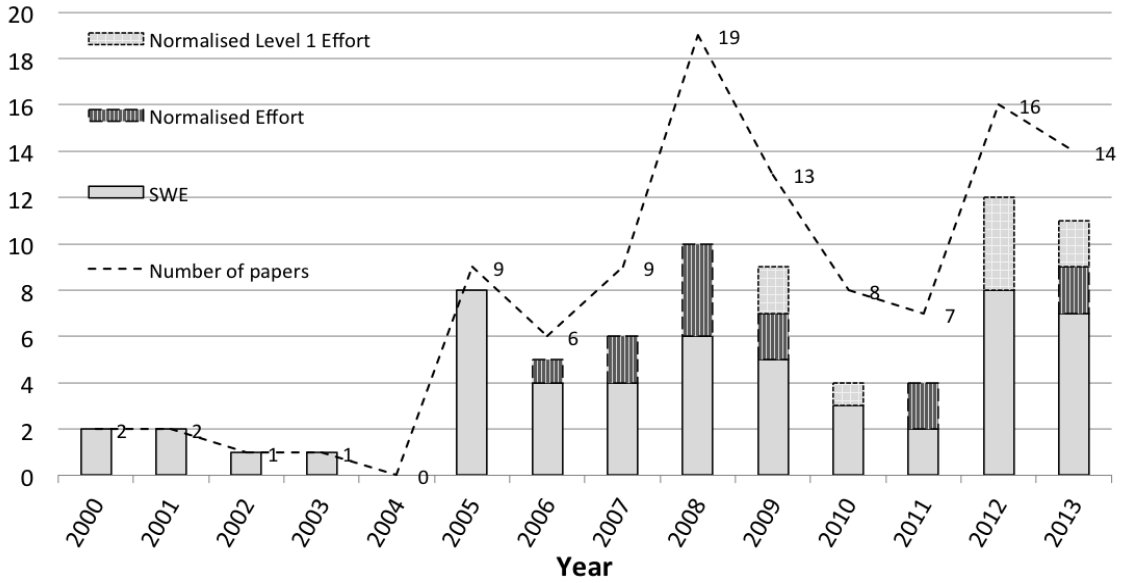

Figure 5. Number of papers per year using SWE, Normalised Effort, and Normalised Level 1 Effort.

The papers published during the period 2000-2005 only used SWE. It should be noted that Release 8, which introduced the normalised effort variables (Normalised Effort and Normalised Level 1 Effort), appeared in 2003. The variable Normalised Effort appeared in one paper published in 2006, and has been significant since then. It is the effort variable that ISBSG recommends. Normalised Level 1 Effort was first used in 2009, and consolidated its position in 2012 and 2013. SWE still dominates over all.

It should be highlighted that in some papers it is not possible to deduce which effort variable was used as the dependent variable. This is something that researchers should report explicitly.

3.1.4. What are the most used ISBSG independent variables in effort estimation research?

The sections above have concentrated on filtering, and dependent variables. Filtering projects, and carefully choosing the dependent variable for effort estimation, are mainly to do with using a valid dataset. These are standard tasks in data preparation. However, the usage of independent variables is not standard: they depend on the research question, and are generally the things of most interest. Hence we concentrate on independent variables in the rest of the paper.

The ISBSG classifies variables into groups of related variables. For example, the variables Development Platform (DP), Language Type (LT), and Used Methodology (UM) are included in the Project attributes group. In this section, we analyse the usage of ISBSG variables as independent variables, first individually and then by group.

3.1.4.1. Individual variables

The ISBSG dataset includes 118 variables. In the 107 papers analysed, 71 of the variables have been used as independent variables for effort estimation. To be precise, these 71 independent variables remain after data preparation procedures such as the preliminary filtering or feature selection steps, and hence they are finally included in the proposed effort estimation models.

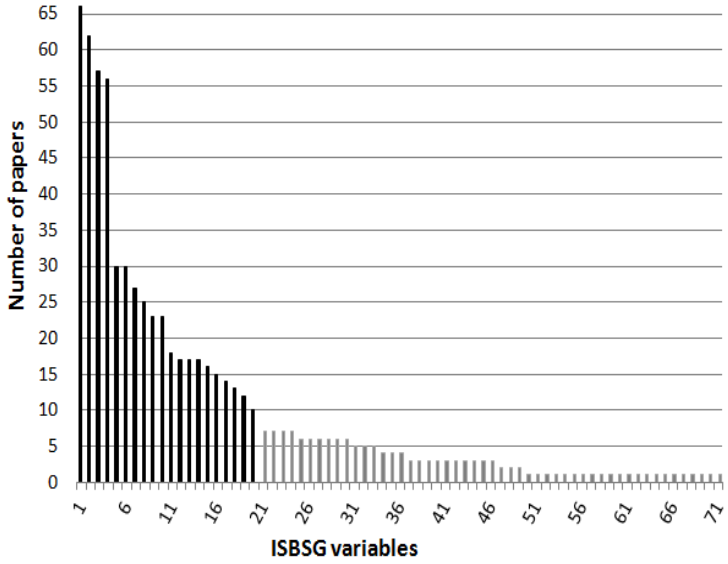

Figure 6. Number of papers that use each ISBSG variable listed in Appendix B.

Figure 6 illustrates the number of papers in which these 71 variables are used as independent variables to construct effort estimation models. These variables are listed in Appendix B. Two sets of variables may be considered: the 20 most used variables, which are all used in at least 10 papers, and the rest, none of which is used in more than 7 papers. The 20 most used variables are presented in Table 3, along with the percentage of papers that employ them, and their frequency of use is shown in black in Figure 6. The most frequently used variables are Functional Size (FS), Development Type (DT), Language Type (LT), and Development Platform (DP), all of which are used in more than 50% of the papers. The next 16 variables in Table 3 have frequencies ranging from 28.0% (Adjusted Function Points) to 9.3% (Average Team Size). The remaining 51 variables (in grey in Figure 6) have each been used in fewer than 7% of the selected papers. The tail of the distribution includes 22 variables that were used only once.

There could be a potential publication bias concerning the venues of publication defined in the search strategy (Jorgensen and Shepperd, 2007). For example, papers published in conference proceedings usually tend to concentrate on narrower questions, perhaps considering fewer variables. There is some indication of that here: while there is no change to which are the 20 most used variables if we consider only papers in journals, the rate of usage of the other 51 variables declines from 10% in journal papers to 7% in all papers.

Table 3

ISBSG variables most frequently used in the selected papers.

Position	Identifier	Variables	Attributes group	Proportion%
1	**FS**	Functional Size	Sizing	61.7
2	**DT**	Development Type	Grouping	57.9
3	**LT**	Language Type	Project	53.3
4	**DP**	Development Platform	Project	52.3
5	**AFP**	Adjusted Function Points	Sizing	28.0
6	**MTS**	Max Team Size	Effort attributes	28.0
7	**OT**	Organisation Type	Grouping	25.2
8	**PPL**	Primary Programming Language	Project	23.4
9	**PET**	Project Elapsed Time	Schedule	21.5
10	**AT**	Application Type	Grouping	21.5
11	**BAT**	Business Area Type	Grouping	16.8
12	**EC**	Enquiry count	Size	15.9
13	**FC**	File count	Size	15.9
14	**IFC**	Interface count	Size	15.9
15	**OC**	Output count	Size	15.0
16	**INC**	Input count	Size	14.0
17	**1DBS**	1st Data Base System	Project	13.1
18	**RL**	Resource Level	Effort attributes	12.1
19	**UM**	Used Methodology	Project	11.2
20	**ATS**	Average Team Size	Effort attributes	9.3

Appendix C shows which of the analysed papers used each of these variables.

The four variables used most often (FS, DT, LT, and DP) are recommended by ISBSG to be the most important criteria for selecting sets of comparable projects (ISBSG, 2009a) for estimation purposes. ISBSG uses these four variables as inputs in its “Early Estimate Checker” tool, which supports early

estimation of project effort and duration, and also in its “Reality checker” tool, which allows checking if the development effort, cost and duration expectations of a project plan are realistic by comparing the planned variables with the results gained in similar completed projects.

In the 107 selected papers, the usage of FS, DT, LT, and DP is as follows: 25 papers used all four of them; 31, 13, 22, and 16 papers used three, two, one and none of these recommended variables respectively. 17 papers only used FS, DT, LT and DP. The four key variables are necessary but not sufficient: they still do not account for a large amount of variation between projects, so researchers normally consider more variables as well. In fact, 74 papers used one or more of FS, DT, LT, and DP in combination with other variables.

In addition, 16 papers do not use any of these four variables. 13 of them used AFP as an alternative for FS. Moreover, 8 papers used variables that convey information about project size in a disaggregated way, such as breaking down FS into counts of inputs, outputs, enquiries, files and interfaces. Other independent variables that have been used in this subset of papers are Normalised PDR (3 papers), variables concerning the duration of the project (PET and Project Inactive Time; 4 papers), and variables concerning the team size (MTS or ATS; 3 papers). Finally, there is also one paper using the breakdown of the work effort during the different project life cycle phases.

ISBSG also recommends using the variables OT, BAT, AT, User Base, and Development Techniques. Indeed, the first three also appear in Table 3.

Trends over time

Concerning the evolution of usage of FS, DT, LT, and DP, Figure 7 presents the proportion of papers in which these four variables have been used since 2005, after the initial introductory period that was noted in Figure 3. Five-year moving averages are presented (except for 2005 to 2008, which present cumulative averages), to smooth the trend.

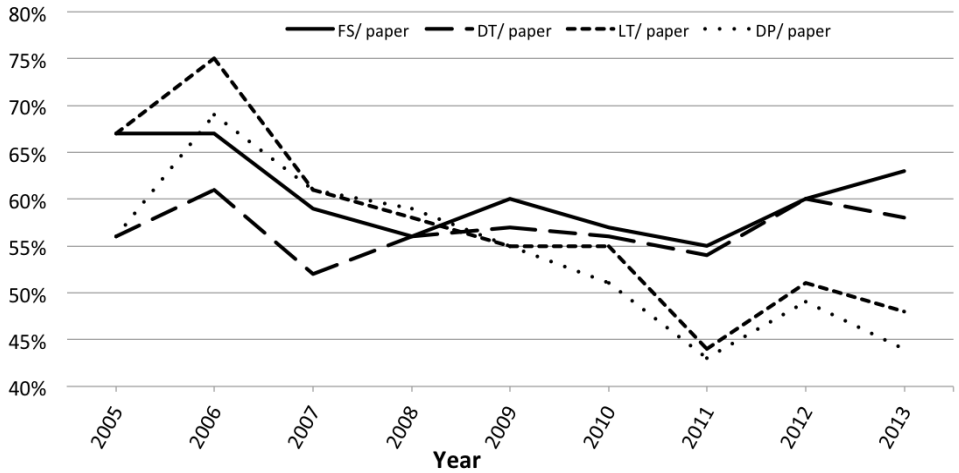

Figure 7. Five-year average evolution of the relative presence of the four most used variables.

FS and DT remain of interest to researchers, but LT and DP have declined in interest (apart from the rise in 2012). The share of variables FS and DT in the selected set of papers has less variability over time than the share of LT and DP. In fact, the standard deviations of these data are 0.11, 0.11, 0.19, and 0.18 respectively. Note, Lokan and Mendes [S67] found in 2009 that the variable DT was influential for a long period of time, but eventually started to disappear from estimation models, seemingly

becoming less useful as enhancement projects began to dominate the set of projects. This variable, however, presents a slight recovery in recent years.

3.1.4.2. Groups of related variables

The ISBSG classifies variables into groups of related variables. These were mentioned in Table 3, in the “Attributes group” column. Table 4 presents (from left to right) the ISBSG groups of variables, the number of times that the variables from a group are used in the papers, the total number of variables that make up each group, the number of variables in each group that have been used in the analysed papers, and the number of variables among the top 20 variables (listed in Table 3) that come from each group.

Table 4

The ISBSG groups of variables.

Attributes group	Number of occurrences	Share of total occurrences	Variables per group	Used variables	Number of top 20 variables included
Project	188	27.0%	24	16	5
Grouping	138	19.8%	12	6	4
Sizing	106	15.2%	4	4	2
Size	102	14.6%	12	12	5
Schedule	58	8.3%	11	10	1
Effort attributes	55	7.9%	6	4	3
Documents & Techniques	10	1.4%	18	4	0
Productivity	9	1.3%	3	2	0
Architecture	8	1.2%	7	3	0
Product	7	1.0%	5	4	0
Size other than FSM	5	0.7%	3	1	0
Software age	4	0.6%	1	1	0
Quality	4	0.6%	4	4	0
Effort	2	0.3%	3	2	0
Rating	1	0.1%	2	1	0
Other metrics	0	0.0%	2	0	0

The group of variables that has the highest number of occurrences is Project attributes, which includes five of the 20 most frequently used variables, including 2 of the 4 most used variables (DP, LT). This group includes 24 technical variables, related to both the development characteristics of the project (such as LT, PPL, and UM), and the development platform (DP and 1DBS), which have been considered in the literature to have a strong influence on the total effort required to develop a project. The 16 variables of this group that have been used at least once to elaborate effort estimation models, with a total of 188 occurrences (27%) across the 107 analysed papers, are listed in Appendix B.

The second group of variables is Grouping attributes (19.8%), which can be considered project context variables (Dolado et al., 2007). It includes one of the 4 most used variables (DT). This group includes organisational variables (such as OT, AT, and BAT) regarding where this software project is developed, and project-specific variables (such as DT, Degree of Customisation and Package Customisation).

Two groups related to the size of the project appear in the third and fourth rows of Table 4. The Sizing group includes FS (the most frequently used variable of all), AFP, Count Approach, and Value Adjustment Factor. All of them have been used in the selected papers. Count Approach describes the method used to size the project software; it is most often used in the data filtering stage (Section 3.1.2), but it is also used as an independent variable in some papers. For projects using other size measures other than functional size (e.g. Lines of Code), the size data is included in the group Size Other than FSM. In general, authors consider that the size variables are closely related to the required effort: FS (the most frequently used variable) and AFP both appear in Table 3. Besides, this is the most intensively used group when the number of variables compared to other groups is taken into consideration. The Size attributes group includes measures of size, but from a more detailed point of view. It includes: input, output, enquiry, file, and interface counts (base functional components for IFPUG sizing); added, changed, and deleted count (Enhancement Data); and entry, exit, read and write counts (base functional components for COSMIC sizing).

Taken together, Sizing, Size attributes and Size Other than FSM represent 30.6% of all variable occurrences used to elaborate effort estimation models.

The fifth group in Table 4 is the Schedule group. This presents a diverse collection of variables, related to the time and the effort of the project allocated throughout the different life cycle stages. Effort for the planning, specifying, designing, building, testing, and implementing stages can be reported separately. Where phase breakdown of effort is provided, and the sum of that breakdown does not equal SWE, the difference is stored in the variable Unphased Effort. Where no phase breakdown is provided, Unphased Effort contains the same value as SWE (ISBSG, 2009d). This group also includes the duration of the project in terms of PET (the only one listed in Table 3), inactive time, and implementation date.

The sixth group in Table 4 is Effort attributes. It includes variables such as MTS, ATS, and RL. This group takes into consideration the project management side from the human resource perspective.

The eight remaining groups of attributes only have 50 total variable occurrences (7.2%) across the set of analysed papers, and do not contain any of the 20 most used variables. Anyway, the seventh group Documents & Techniques includes the variable Development Techniques that is one of the less used ISBSG recommended variables (6.5%). Since these techniques have not been recorded as being phase-specific, they may apply to any part of the development life cycle. This variable, the most frequently used from this group, appears in the 23rd position of Appendix B with a missing value rate of 54.8%. Additionally, the variable User Base is also recommended by ISBSG. This variable encompasses a set of variables (Locations, 45th; Concurrent Users, 48th; Business Units, 70th; and Distinct Users) that are included in the Product attributes group and in total appear six times (5.6%). Incidentally, the variable in this set with the fewest missing values (User Base Locations) has 85% missing values.

Trends over time

Figure 8 shows the relative presence of the six most used ISBSG attributes groups over time. As in Figure 7, five-year moving averages are presented except for 2005 to 2008, which present cumulative averages. Project attributes is the group that presents the higher participation throughout the period. In fact, Project attributes decline slowly, but that does not mean that they are used less, just that more variables from other groups are used as well, so the share of the Project attributes group goes down. Grouping attributes is the second group of variables and presents a similar behaviour over time.

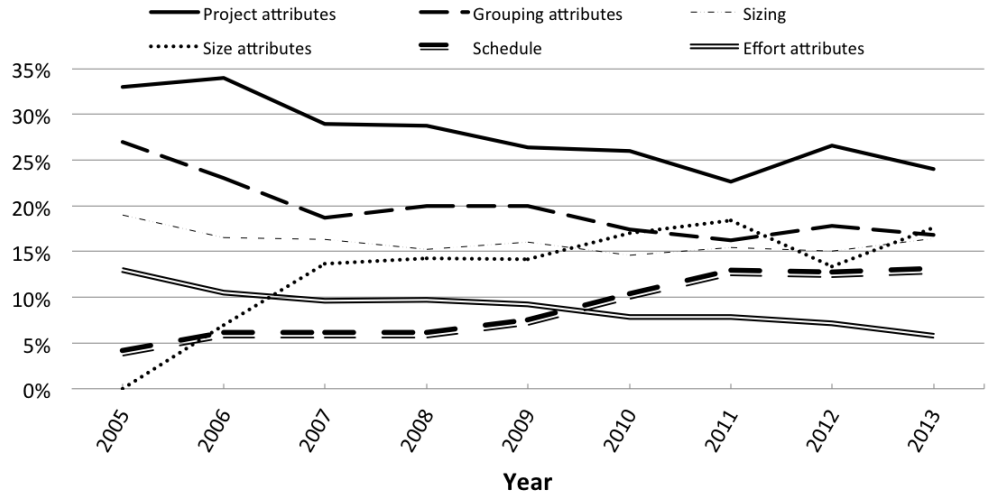

Figure 8. Evolution of the five-year average relative presence of the used ISBSG attributes groups.

Within the groups that measure size, the Sizing group shows steady usage in papers published from 2005 to 2013, with an average value of 16% of all independent variables used: this reflects that FS is very commonly used. Interest has grown in the Size attributes group, which until 2005 had no presence.

The groups Sizing and Effort have quite steady participation and have been used consistently by researchers with a standard deviation of 0.03. Project and Size attributes present a greater variation, with standard deviations of 0.09 and 0.12 respectively. The Effort group has a downward trend, with a presence of only 3% in 2013. The Schedule group presents an increase starting in 2009 which becomes steady after 2011.

In summary, there are groups that have been used consistently since 2005, other groups that have been used for a period but are no longer receiving the same level of attention, and the Size attributes group which has been gradually playing a more important role.

3.2. RQ2: What are the characteristics of these independent variables?

To answer this question, we perform a descriptive statistical analysis focused on the 20 most used independent variables (Table 3), but also considering other variables (Appendix B). This analysis is organised by the groups to which these variables belong, since this arrangement provides a useful classification of complementary and alternative variables, along with their relationships.

For each variable in each group, these features are considered: type, meaning, descriptive statistical parameters, missing data, and for categorical variables the different values and their frequencies. The relationships between variables are also considered. Thereby, when possible, some subsets that cluster related variables are identified within each group. Also the least used variables in each group are considered, to get some insights about the group and the relationships between their variables, and possible reasons for selecting some variables over others.

All of these considerations may assist researchers and practitioners in their selection of independent variables for effort estimation models.

As noted in Section 3.1.2, the ISBSG dataset includes two fields related to Data Quality. One of these (UFP Rating) is less often considered. The most common approach by researchers is to exclude projects with a low Data Quality Rating (C or D) from analysis, following the recommendation of ISBSG

(ISBSG, 2009a). Of the 5052 projects in Release 11 of the ISBSG dataset, 308 had a Data Quality Rating of C or D, and 4744 had a Data Quality Rating of A or B. All following calculations in this paper are performed on those 4744 (93.9%) projects.

3.2.1 Project attributes

The Project attributes group (Table 5) consists of six variables with a generic technical perspective (LT, DP, PPL, UM, Case Tool Used, and How Methodology Acquired) and two subsets of variables specifically related to DP: the 1st platform and 2nd platform attributes (language, hardware, operating system, etc.). The most used variables in this group are summarised below.

Table 5

Variables from Project attributes group.

Position	Variables	Type	Proportion %	Missing values %
3	Language Type (LT)	Nominal	53.3	9.7
4	Development Platform (DP)	Nominal	52.3	18.9
8	Primary Programming Language (PPL)	Nominal	23.4	7.9
17	1st Data Base System (1DBS)	Nominal	13.1	41.8
19	Used Methodology (UM)	Nominal	11.2	42.4
28	CASE Tool Used	Nominal	5.6	66.8
29	How Methodology Acquired	Nominal	5.6	69.6
44	1st Operating System	Nominal	2.8	50.1
47	1st Hardware	Nominal	1.9	42.5
63	1st Language	Nominal	0.9	8.2
64	1st Debugging Tool	Nominal	0.9	86.6
65	1st Other Platform	Nominal	0.9	77.5
66	2nd Hardware	Nominal	0.9	96.6
67	2nd Language	Nominal	0.9	96.6
68	2nd Operating System	Nominal	0.9	97.4
69	2nd Data Base System	Nominal	0.9	98.2

LT, which is in the third position out of 71, defines the language type used for the project. In the dataset, 3rd generation languages dominate (63.4%) and 4th generation languages are also well represented

(33.5%), but 2nd generation languages (0.3%), 5th generation languages (0.02%), and Application Generator (2.7%) are hardly represented. Statistical evidence exists indicating that LT has an impact on effort [S28] and that productivity is dependent on the language generation type ([S96], (Kitchenham, 1992)). In practice, high level programming languages, and in particular all 4GL languages are designed to reduce programming effort, but in contrast they require considerable effort during the design phase [S25].

DP (4th position) defines the primary development platform, determined by the operating system used. Each project is classified as PC (17.5%), Mid Range (9.5%), Mainframe (38.8%), or Multi-platform (34.1%). This variable is clearly determined at the early stage of any software project compared to LT. According to (Hill, 2010), it is the best indicator of the environment in which a project is developed, and does not refer specifically to the hardware platform.

PPL (8th position) indicates the primary language used for development. In Release 11 of the dataset, 136 programming languages are represented. The most frequently used languages are COBOL (17.3%), Java/J2EE/Javascript (14.3%), Visual Basic (7.9%), PL/I/PL/SQL (7.5%), C/C++/C# (7.2%), Oracle (3.8%), .Net (3%), SQL (2.6%), Natural (1.9%), COOL:GEN (1.4%), Access (1.2%), Powerbuilder (1%), and ASP (0.9%). This variable has a similar missing values rate (7.9%) to LT (9.7%). Since particular programming languages belong to one of the language types, this variable is in a way redundant with LT [S96]. When two or more independent variables contain redundant information, only one is to be considered ([S73], (Bibi et al., 2008)). LT is more often used than PPL, except where information about the specific programming language is required.

As mentioned previously, this group also provides variables related to 1st and 2nd platform attributes. These details regarding 1st or 2nd platform attributes are used less frequently. Detailed specifications of a software project are not suitable candidates, since they may require a significant amount of time to be determined and processed. Besides, in most cases, there is no other platform used to build or enhance the software. The first variable in these two subsets is 1DBS, which appears in the 17th position. Where known, this is the primary database used in a project. Where unknown, this variable indicates whether or not the project used a database management system.

UM variable appears in the 19th position. This variable is the fourth of six from this group that are not related to the 1st or 2nd platform attributes. UM states whether (72.6% of the non-missing values) or not (5.2%) a development methodology was used by the development team to build the software (‘Don’t know’ accounts for the other 22.3%).

1DBS has 41.8% of its values missing, and UM has 42.4% missing values, which could explain their position in the tail of Table 5 with respect to the other three variables in this group that appear among the ten most commonly used. In the 1st platform of subset attributes, only one variable has a missing percentage better than 1DBS. This variable is 1st Language (missing values rate of 8.2%), which appears in the 63rd position, but is somewhat redundant with PPL which appears in the 8th position.

The last two variables that convey a generic technical perspective are Case Tool Used (28th position) and How Methodology Acquired (29th). They have 66.8% and 69.6% missing values respectively.

3.2.2 Grouping attributes

The Grouping attributes (19.8%) include organisational variables like OT (7th), AT (10th), and BAT (11th), and project-specific variables such as DT (2nd), Package Customisation (22nd), and Degree of Customisation (57th).

Table 6

Variables from Grouping attributes group.

Position	Variables	Type	Proportion %	Missing values %
2	Development Type (**DT**)	Nominal	57.9	0.0
7	Organization Type (**OT**)	Nominal	25.2	24.8
10	Application Type (**AT**)	Nominal	21.5	27.9
11	Business Area Type (**BAT**)	Nominal	16.8	24.8
22	Package Customisation	Nominal	6.5	69.3
57	Degree of Customisation	Nominal	0.9	98.4

DT describes whether the development was a new development (38% of the values), enhancement (60.2%), or a re-development (1.7%). A re-development is similar to a new development, using new technologies to replace or upgrade an existing software product. This variable has no missing values. It is one of the most important criteria for selecting projects [S68], and is suggested by ISBSG guidelines for use in the estimation process as well as for benchmarking (ISBSG, 2009a). Empirical evidence exists that the development type influences project effort ([S1], (Moses et al., 2006)). Project delivery rates (expressed in terms of hours per function point) for new developments are different from those for enhancements. New developments average eight to twelve hours per function point while enhancements average 12 to 16 hours per function point (Hill, 2010). The difference is probably due to factors unrelated to the development type. For example, a greater proportion of enhancements were within mainframe projects, whereas new developments include more PC projects.

OT identifies the type of organisation that submitted the project. It has 24.8% missing values and presents 142 distinct categories that are usually regrouped. Example values for OT are communications (22.3% of all projects where the organisation type is known), insurance (16.8%), banking (12.6%), financial, property & business services (9.7%), manufacturing (7.7%), and government (6.1%) (ISBSG, 2009b). Other categories of this variable are community services, computers & software, electricity & gas & water, public administration, transport & storage, and wholesale & retail trade. The missing values of this variable can be partly explained by the inherent difficulty for the user to select among a large number of options. The values of OT are usually regrouped by researchers: when a variable has too many distinct levels, it usually requires some preprocessing in order to reduce the number of levels, since it is neither practical nor sensible to perform regression analysis against it (Deng and MacDonell, 2008). The preprocessing, however, could be complex due to the values of this variable being stored as string collections depicting the different options for industries in which an organisation could be classified.

AT identifies the type of application within the business area and organisation/industry type being addressed by the project. The application type is related to its primary intended use (ISBSG, 2009c). Some of the most important application types are Financial transaction process/accounting (31.2%), Transaction/production system (13.3%), Management Information System (MIS) (10.6%), Embedded system (3%), Customer billing/Relationship Management (CRM) (1.8%), Document management (1.6%), Network management (1.5%), Office Information System (OIS) (1.5%), Stock control & order processing (1.2%), Electronic Data Interchange (EDI) (1.2%), etc. This variable has 27.9% missing values and is occasionally used for data partitioning [S37].

The last Grouping variable is BAT, which is conceptually linked to AT and reports the subsystem of the company affected by the project, or in other words the business area within the organisation that the application supports. It may be different to the organisation type or the same as the organisation type (e.g., Manufacturing, Personnel, and Financial). Increasingly, software projects do not have a unique and limited impact on a specific department, but have rather a greater impact, affecting several departments or even the whole organisation. Angelis et al. [S34] identified two homogeneous sets by performing ANOVA and the range tests. One homogeneous set includes R&D, Telecommunications, Engineering, Sales, and Financial while the other includes Banking, Account, Legal, Personnel, Manufacturing, and Inventory. This variable has the same amount of missing values as OT, because it may be the same, different, or assumed to be the same. In some cases, this variable has the value

“Don't know” – effectively a missing value. The variables OT, AT, and BAT present many different discrete values and in some instances, categories that were found in no more than five records in the entire dataset are merged under the label of ‘Other’ [S37].

Other variables in this group that have been used less frequently are: Package Customisation (used in seven papers), which indicates whether the project was a package customisation; and Degree of Customisation (used in one paper): if a project was based on an existing package, this field provides comments on how much customisation was involved.

The other variables in this group have very little data, and are not used at all by researchers. These variables indicate if the project was performed under software standards (CMM, CMMI, SPICE, ISO 9002, TICKIT and others such as SAS70) maintained by international organisations, which define a series of actions and documentation structures as well as the content required to deliver quality software processes. More than 94.9% of values are missing for all of these variables.

3.2.3 Size-related attributes

The variables related to the size of a project are in the groups Sizing, Size attributes and Size Other than FSM.

Table 7

Variables from Sizing and Size attributes groups.

Position	Variables	Type	Attributes group	Proportion %	Missing values %
1	Functional Size (**FS**)	Continuous	Sizing	61.7	29.1
5	Adjusted Function Points (**AFP**)	Continuous	Sizing	28.0	3.5
12	Enquiry count (**EC**)	Continuous	Size	15.9	67.3
13	File count (**FC**)	Continuous	Size	15.9	67.0
14	Interface count (**IFC**)	Continuous	Size	15.9	67.3
15	Output count (**OC**)	Continuous	Size	15.0	67.0
16	Input count (**INC**)	Continuous	Size	14.0	65.6
21	Count Approach	Nominal	Sizing	6.5	0.0
24	Added count	Continuous	Size	6.5	64.3
30	Changed count	Continuous	Size	5.6	80.4
36	Deleted count	Continuous	Size	3.7	80.4
37	Value Adjustment Factor	Continuous	Sizing	2.8	64.3
46	COSMIC (Entry, exit, read, write)	Continuous	Size	2.8	97.4

The variable Functional Size (FS) belongs to the Sizing group. Up to and including Release 8, it represented the size in Adjusted Function Points (AFP). Since Release 9 (released in 2004), it represents the unadjusted function point count (UFP), which reflects the specific countable functionality provided to the user by the project or application (ISBSG, 2009b), before any adjustment for General System Characteristics, such as complexity. FS and AFP have been reported separately in the dataset since Release 9. 38 papers used data up to Release 8. Of the 68 papers using Release 9 or later, 30 used AFP and 38 used FS. (One other paper used ISBSG’s other repository, the Maintenance and Support repository, which uses a variety of size measures.) Thus AFP is the underlying size measure in 68 papers, compared to UFP in 38 papers. However, UFP (under the label of FS) has been preferred by authors over AFP since it has been possible to choose between the two (56% of 68 papers). Both

FS and AFP are dependent on the Functional Size Measurement (FSM) method used: IFPUG, NESMA, FiSMA, and MARK II. The variable AFP only has 3.5% missing values versus FS which has 29.1%.

Concerning the Functional Size Measurement method, while IFPUG projects dominate the repository (76.2%), the second most common method COSMIC represents 7.1%. FS for IFPUG projects has 735 out of 3614 values missing. This variable has a mean value size of 417 fps, median of 194, and a range of 16145 fps (maximum 16148 fps and minimum 3 fps). For COSMIC projects, FS has a mean value of 244 fps and a range of 2087 fps (maximum 2090 fps and minimum 3 fps). As to AFP, IFPUG projects have a mean AFP value of 424 fps, median of 179, and a range of 19997 fps (maximum 20000 fps and minimum 3 fps). AFP for IFPUG projects only has 1 missing value out of 3614 projects.

In the Sizing group, there are two other variables that do not appear in Table 3. The variable Count Approach appears in the 21st position. It is mainly used in the filtering process, to select projects sized with a specific approach, rather than in the estimation itself. It has no missing values. The other variable is Value Adjustment Factor (VAF), which can be considered as a component of AFP. The VAF is calculated from 14 General System Characteristics (GSC), which are technical factors that account for a variety of non-functional system requirements (e.g., performance, reusability, operational ease, etc.). This adjustment factor is studied in three papers, two of which [S20, S70] considered the individual GSC ratings. VAF is missing, and assumed to be equal to one (ISBSG, 2009d), for 64.3% of projects.

The Size Attribute variables provide information about the size magnitude in a disaggregated way. Most often used are five fields that break down IFPUG FS into its base functional components of inputs (INC), outputs (OC), enquiries (EC), files (FC), and interfaces (IFC). These variables appear in the 12th to 16th positions of Table 3. The percentage of missing values is around 67% for these variables. The breakdown of COSMIC FS into its base functional components of entries, exits, reads, and writes is considered in three papers. Function Points representing new or Added functions, Changed functions, and Deleted functions are considered in seven, six, and four papers respectively.

3.2.4 Schedule

The Schedule group only includes one variable among the 20 most frequently used variables. It is the variable PET (9th position), which represents the total elapsed time for the project in calendar months (actual duration). This variable is related to the effort on a software project, when considered with the resources that have been allocated. These resources are to some extent reflected by the team size (the Effort group, discussed in the next section) and dedication of the team. To obtain the actual time spent working on the project, the Project Inactive Time (calendar months with no activity) should be subtracted from PET. PET has a mean value of 9.8 months, a median value of 6 months and a standard deviation of 42.4 months. A high degree of asymmetry and the existence of outliers have also been reported [S55].

Table 8

Variables from Schedule attributes group.

Position	Variables	Type	Proportion %	Missing values %
9	Project Elapsed Time (**PET**)	Continuous	21.5	16.6
26	Project Inactive Time	Continuous	5.6	55.1
31	Effort Plan	Continuous	4.7	86.7
32	Effort Specify	Continuous	4.7	75.4
35	Implementation Date	Ordinal	3.7	12.5

39	Project Activity Scope	Nominal	2.8	50.7
40	Effort Design	Continuous	2.8	86.4
41	Effort Build	Continuous	2.8	70.1
42	Effort Test	Continuous	2.8	72.1
43	Effort Implement	Continuous	2.8	82.3

Other variables in this group that are presented in Appendix B are Project Inactive Time (26th, 6 occurrences), Implementation Date of the project (35rd, 4 occurrences), and Project Activity Scope, which indicates which tasks were included in the project work effort data recorded (39th, 3 occurrences). These tasks, and their corresponding variables which contain the breakdown of the work effort, are: Effort Plan (31st, 5 occurrences), Effort Specify (32nd, 5 occurrences), Effort Design (40th, 3 occurrences), Effort Build (41st, 3 occurrences), Effort Test (42nd, 3 occurrences), and Effort Implement (43rd, 3 occurrences).

3.2.5 Effort attributes

The Effort group records information about the features of the project software team, which reflect the influence of managerial decisions [S37].

Table 9

Variables from Effort attributes group.

Position	Variables	Type	Proportion %	Missing values %
6	Max Team Size (**MTS**)	Continuous	28.0	70.1
18	Resource Level (**RL**)	Nominal	12.1	0.0
20	Average Team Size (**ATS**)	Continuous	9.3	88.0
49	Recording Method	Nominal	1.9	0.7

MTS (6th position on the list) represents the maximum number of people who are assigned to the project at any time (peak team size). MTS is known to be one of the most important factors affecting project delivery rate (Hill, 2010). This number is only given for the development team (Level 1). Teams of 5 or fewer people comprise 46.9% of the projects. Five is the most common maximum size. Teams of fewer than 10 people comprise 77.7% of the projects, and the average of MTS is 8.8 people. There is evidence that once the team size exceeds five people, productivity decreases: projects with maximum team sizes of five or more have significantly worse project delivery rates than projects with smaller teams (Hill, 2010). This variable has a high level of missing values (70.1%).

A complementary variable to MTS is ATS (20th position). ATS records the average number of people that worked on the project, for the development team only. 62.5% of projects have up to five people on the development team and 18.7% have ten or more people, with an average of 6.5 people. The information collected by this variable is very interesting; arguably more so than the peak value from the previous variable. This variable reports a mean value, which represents in a more representative way the resources used in Level 1. However, the reason this variable may not be used more is because it has a high number of missing values (88%), which may be attributed to the difficulty of reliably responding to the question.

MTS and ATS, along with PET, can give an idea of the effort made in a project. However, MTS and ATS suffer from a high number of missing values, and Stamelos et al. [S53] commented that the inclusion of MTS and PET in the estimation model did not produce any satisfactory improvement.

RL (18th position) is a nominal variable in the Effort group. As mentioned in section 3.1.1, this variable indicates the people whose time is included in the work effort data reported. The number in this field indicates that all effort at this level and lower levels is included in the effort values. Four levels have been identified: Level 1 (85.7%) means that effort is only recorded for the development team; Level 2 (4.4%) adds development team support; Level 3 (0.6%) adds computer operations involvement (0.6%); and Level 4 (9.4%) adds effort by end users or clients. This variable presents no missing data and is more often used in the filtering process than in the model itself.

The other variables in this group are scarcely used (Recording Method, used in 12% of papers as a filtering variable) or not used at all (Ratio of Project Work Effort to Non-Project Activity and Percentage of Uncollected Work Effort).

3.3. RQ3: How, and to what extent, have ISBSG variables been used as independent variables to build effort estimation methods?

First, we analyse the frequency of use of different estimation methods, both alone and in combination with other methods. Next, we analyse the number of independent variables used, across the full set of papers and with different estimation methods. Finally, differences are highlighted in the level of usage of the most frequent independent variables with different estimation methods.

3.3.1 Effort estimation methods

In (Fernández-Diego and González-Ladrón-de-Guevara, 2014), three families of estimation methods were identified as predominant in analysing the ISBSG dataset: Regression, Machine Learning and Estimation by Analogy. Regression was the most frequently used family. This is consistent with the fact that regression is considered to provide good accuracy ([S13], (Jorgensen and Shepperd, 2007)), and is often used to contrast the results that have been obtained by other methods.

The Regression family includes a wide range of regression-based estimation models: linear regression, multiple linear regression, ordinary least-squares (OLS) regression, stepwise regression, robust regression, ordinal, categorical and logistic regression, multivariate adaptive regression, ANOVA, and ANCOVA to name a few. The second family is made up of Machine Learning (ML) methods, which includes, among others, Artificial Neural Networks, Genetic Algorithms, Support Vector Machines, Bayesian Networks, and Association Rules (Wen et al., 2012). The third type of method is Estimation by Analogy (EbA), which compares the considered software project with some similar historical projects with known characteristics for deriving software effort estimates [S53].

Table 10 presents the families of methods mentioned above, and displays by rows the number of references in which each family appears (note that some papers used several methods). The first row presents the methods used in the overall set of 107 papers. In the second row, only those papers that used regression are taken into account, and the number of papers that used other methods in addition to regression is presented. The same structure is followed in the other rows.

Table 10

Distribution of the most used families of estimation methods.

Subset of papers by estimation method	Regression	ML	EbA	Others
Total of 107 papers	**76**	**40**	**22**	**24**
76 (Regression)	**76**	21	11	13
40 (ML)	21	**40**	5	11

22 (EbA)	11	5	**22**	5
24 (Others)	13	11	5	**24**

Seventy-six articles (71%) used regression, making it the most frequently used method. In many papers it was used alone (41 out of 107 papers). Next, the ML family includes 40 references, representing 37.4% of papers. In the 40 papers that used machine learning methods, regression is also used in 21 of them, mainly to contrast the results. ML methods are used exclusively in 12 papers. The third largest family is Estimation by Analogy, with 22 papers (20.6%); it was used exclusively in 6 papers. In sum, the distribution obtained for effort estimation is consistent with (Fernández-Diego and González-Ladrón-de-Guevara, 2014). Finally, the category called "Others" includes 24 papers (22.4%) that use an extensive variety of methods such as combination of estimates, function points, fuzzy, simulation, survival analysis, multiple criteria linear programming, production function, sequential quadratic programming, case studies, Particle Swarm Optimisation (PSO), etc., that cannot be include in the previous families. In particular, as many as 11 papers used Fuzzy methods, usually in combination with other methods.

3.3.2 Number of independent variables used to construct effort estimation models

This subsection analyses the average number of independent variables that have been used in the selected papers, overall and by estimation approach. This is the final set of variables included in the proposed effort estimation models, after preprocessing (during the preprocessing stage, it is common that columns are pruned, usually due to a high level of missing data [S1, S13, S103, S107] (Cf. 3.1.2) when no imputation treatment is performed).

Firstly, Figure 9 shows the frequency and cumulative distribution of the number of used independent variables used in the 107 papers. An average of 6.5 variables has been used in the selected papers with a mode of 5 and a standard deviation of 5.3. We can find papers using from 1 to 16 independent variables; one paper [S2] used as many as 44 variables, analysing the root cause of missingness of software effort data.

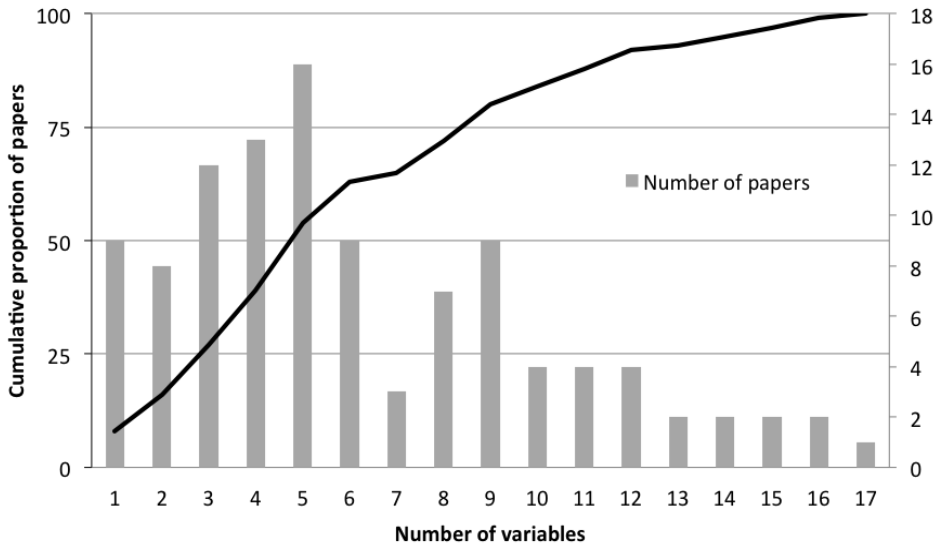

Figure 9. Number of independent variables used per paper.

Table 11 presents a similar analysis for each family of estimation methods. The outlier [S2] has not been considered in the analysis.

Table 11

Number of independent variables used with each family of estimation methods.

	Number of papers	Mean	Mode	Standard Deviation
Regression	76	5.7	5	3.6
Regression exclusively	41	5.3	8	3.1
ML	40	6.4	5	4.0
ML exclusively	12	4.7	4	3.4
EbA	22	8.5	9	4.2
EbA exclusively	6	10.8	16	5.4
Others	24	6.6	5	3.8
Total	**106**	**6.2**	**5**	**3.8**

When regression is used exclusively, an average of 5.3 variables are used as independent variables, and the standard deviation is lowest (3.1). On average, machine learning-based methods (when used exclusively) use the fewest variables (4.7), while the EbA method uses the most (8.5). It should be highlighted that in the case of EbA the mode is as high as 9 variables. When EbA is used exclusively, the average of variables reaches 10.8 with the highest standard deviation (5.4). All these results are related to the research question in the next section.

3.3.3 Level of usage of the most frequent independent variables regarding the estimation methods used.

Table 12 compares the number of times that the 20 most used independent variables are used in papers considering the aforementioned families of estimation methods, with respect to the overall subset of 107 papers. Thereby, the columns present the number of occurrences of each variable in the papers that have used a type of estimation method and the corresponding “Change” columns indicate the usage percentage difference of each variable considering the number of papers that have employed a specific estimation method in relation to the 107 papers. For example, 62 of all 107 papers (57.9%) consider DT; 42 of 76 papers (55.3%) that used regression consider DT; the percentage of papers that used regression and considered DT is 2.7% less than the total percentage of papers that considered DT. These changes reflect variations in the usage of the most frequent variables. Changes exceeding 3% in absolute value have been marked with an arrow. Some of these variations are discussed in the next paragraphs.

Table 12

Distribution of the most frequent independent variables regarding estimation methods.

Variables/ Methods	Total (107)	Regression (76)	Change Regression	Regression exclusively (41)	Change Regression exclusively	ML (40)	Change ML	EbA (22)	Change EbA	Others (24)	Change Others
FS	**66**	48	1.5%	31	13.9%↑	22	-6.7%↓	9	-20.8%↓	9	-24.2%↓

DT	**62**	42	-2.7%	27	7.9%↑	20	-7.9%↓	14	5.7%↑	9	-20.4%↓
LT	**57**	41	0.7%	23	2.8%	22	1.7%	11	-3.3%↓	8	-19.9%↓
DP	**56**	41	1.6%	23	3.8%↑	19	-4.8%↓	13	6.8%↑	10	-10.7%↓
AFP	**30**	18	-4.4%↓	7	-11.0%↓	13	4.5%↑	9	12.9%↑	9	9.5%↑
MTS	**30**	23	2.2%	12	1.2%	6	-13.0%↓	10	17.4%↑	7	1.1%
OT	**27**	20	1.1%	10	-0.8%	10	-0.2%	8	11.1%↑	6	-0.2%
PPL	**25**	18	0.3%	8	-3.9%↓	9	-0.9%	8	13.0%↑	6	1.6%
PET	**23**	15	-1.8%	6	-6.9%↓	10	3.5%↑	5	1.2%	7	7.7%↑
AT	**23**	16	-0.4%	8	-2.0%	9	1.0%	8	14.9%↑	4	-4.8%↓
BAT	**18**	13	0.3%	4	-7.1%↓	8	3.2%↑	8	19.5%↑	4	-0.2%
EC	**17**	6	-8.0%↓	1	-13.4%↓	9	6.6%↑	9	25.0%↑	7	13.3%↑
FC	**17**	7	-6.7%↓	1	-13.4%↓	10	9.1%↑	8	20.5%↑	6	9.1%↑
IFC	**17**	7	-6.7%↓	1	-13.4%↓	9	6.6%↑	9	25.0%↑	7	13.3%↑
OC	**16**	6	-7.1%↓	1	-12.5%↓	8	5.0%↑	9	26.0%↑	7	14.2%↑
INC	**15**	6	-6.1%↓	1	-11.6%↓	8	6.0%↑	8	22.3%↑	6	11.0%↑
1DBS	**14**	9	-1.2%	2	-8.2%↓	6	1.9%	6	14.2%↑	4	3.6%↑
RL	**13**	7	-2.9%	4	-2.4%	5	0.4%	4	6.0%↑	3	0.4%
UM	**12**	10	1.9%	3	-3.9%↓	6	3.8%↑	3	2.4%	1	-7.0%↓
ATS	**10**	7	-0.1%	4	0.4%	3	-1.8%	1	-4.8%	6	15.7%↑

Regression: There are six variables (AFP, EC, FC, IFC, OC, INC) that are used less in regression models than overall, by more than 3%. All these variables belong to the Sizing and Size attributes groups. Other variables used less in regression are RL and DT. The variables MTS, UM, DP, and FS increase their participation in regression models, but by less than 3%. The fact that the increases do not offset the declines can be explained because regression models use fewer variables, as mentioned in previous paragraphs. All in all, the authors of these papers prefer to use FS (which has almost 30% of missing values versus AFP with only 3.5%) to convey the size of the projects. These trends are clearer when considering the papers that use regression exclusively; those papers tend to use more the ISBSG recommended variables (FS, DT, LT, and DP) in comparison with the general behaviour (107 papers). The size variables (AFP, EC, FC, IFC, OC, and INC) are used much less and there are other variables that present significant reductions such as BAT, PET, PPL, and 1DBS.

Machine Learning: The variables FS, DT, DP, and MTS have a reduction in their relative usage (over 3%). On the other side, the variables AFP, PET, BAT, EC, FC, IFC, OC, INC, and UM have a higher presence than in the full set of papers. In contrast to regression methods, authors that work with machine learning methods prefer the group of variables “Count”, which breaks down the size of a project using different components, even though the percentage of missing values is around 67% in all cases.

EbA: 15 variables (both nominal and continuous variables) increase their relative presence by more than 3%. This result is consistent with the fact that EbA methods generally use more variables, as was mentioned before. The nominal variables are more used by this family of methods. Similarly to ML models, the “Count” variables present an increase in their participation, in this case, more than 20% with a corresponding shrinking of FS variable. Moreover, a balance is observed in the relative usage of LT and PPL.

Others: Since it is a miscellany of methods, the behaviour deviates from the standard. However, Table 12 shows that the four variables that are recommended by ISBSG have a lower participation, along with UM and AT. Additionally, this group of methods have been used in 10 out of the 16 papers that did not

use any of ISBSG's four recommended variables. In return, the "Count" variables increase. Another variable of the group Sizing (AFP) also increases, as well as ATS, PET, and 1DBS.

4. Discussion

In this section the principal findings of the study are highlighted, as well as the study limitations and finally the implications for research and practice.

4.1. Principal findings

The usage of ISBSG variables has been described considering filtering, dependent, and independent variables in effort estimation models, with the main focus being on independent variables.

Filtering variables

Researchers normally use a filtering process to select a suitable subset of projects from the ISBSG dataset for analysis. Only one paper did no filtering.

Filtering mainly aims to ensure high quality data, with comparable definitions for the most relevant variables (size and effort), and confidence in effort values. Two variables (Data Quality Rating (82.2% of papers describing the filtering process), and Count Approach (64.4%)) are commonly used. Four other variables are used for filtering in some papers: RL (35.6%), change in effort due to normalisation (33.3%), UFP Rating (14.4%), and Recording Method (12.2%).

Projects may be removed if they are outliers; this occurs in 36% of papers, with the variables considered for detecting outliers being size, effort, and PDR (effort/size).

Projects may be removed due to missing data. The ISBSG dataset has many variables that have missing values for more than 40% of the projects, reducing the available data and potentially leading to biased models. 35% of papers exclude projects that have missing data in one or more relevant variables (listwise deletion). However, there is a growing use of methods for handling missing data, with the k-nearest neighbours imputation method, the most common approach, used in 14% of the papers.

Finally, 43% of papers exclude projects that do not fall within a specific domain of interest. Which variables are used for this filtering purpose depends on the interest of the researcher: most often they include the development type, organisation type, application type, or implementation date.

Dependent variables

Nearly all (89%) of the papers report that effort is the dependent variable in effort estimation models, although some use productivity (4.7%) or project delivery rate (3.7%). However, the ISBSG dataset has three effort variables. While some papers state which effort variable they use as their dependent variable, several do not. The dependent variable can sometimes be inferred from the description of the preliminary data analysis, but not always: 19% of the papers are not repeatable, because their dependent variable is ambiguous or unknown.

The most used effort variable is SWE, which is simply the reported effort value, perhaps not covering all life cycle phases of a project, measured in staff hours. Another effort variable, Normalised Effort, was added in Release 8 to facilitate comparisons between projects that include different subsets of life cycle phases: Normalised Effort (12.1%) is ISBSG's estimate of the total effort when any missing life cycle phases need to be added. However, there can still be some inconsistency between projects, even when using Normalised Effort, because projects report effort at different values of RL. To ensure maximum

consistency, we recommend that researchers use Normalised Level 1 Effort. In any case, it is important that researchers report explicitly which effort variable they use.

Independent variables – considered individually

Seventy-one out of 118 ISBSG variables have been used as independent variables (listed in Appendix B). The twenty most used variables range in use from 61.7% to 9.3% of papers. The other 51 variables have been scarcely used.

The four most frequently used independent variables are FS (61.7%), DT (57.9%), LT (53.3%), and DP (52.3%). Considering the selected set of papers, 17 out of 107 use FS, DT, LT, and DP exclusively and 16 papers out of 107 do not use any of them. The analysed papers generally follow ISBSG's recommendations for defining sets of comparable projects for estimation purposes: 64.5% of the papers use at least two of the variables that ISBSG recommends.

These four key variables are necessary but not sufficient: they still do not account for a large amount of variation in software project effort. Thus, researchers usually consider more variables as well. ISBSG also recommends using the variables OT, BAT, AT, User Base, and Development Techniques. In fact, 74 papers used one or more of FS, DT, LT, and DP in combination with other variables. Sixteen papers do not use any of the four key variables, and either use AFP as an alternative for FS, or consider functional size in its base components of counts of inputs, outputs, enquiries, files and interfaces.

Considering the evolution over time of papers that used ISBSG data, after an introductory period (2000-2004), there was an increase in the number of papers published, reaching a peak in 2008 (19 papers). A decrease followed in 2009-2011, before a recent increase (2012-2013). When considering the usage of the recommended ISBSG variables, FS and DT remain of interest to researchers over time, whereas LT and DP have declined in interest (apart from a spike in 2012).

Independent variables – considered by group

ISBSG organises all 118 variables into groups. The ISBSG groups provide an arrangement of complementary and alternative variables that may guide the selection of variables in effort estimation models. The most used group is Project attributes, which accounts for 27% of total occurrences, followed by Grouping attributes (19.8%), Sizing (15.2%) and Size (14.6%) attributes, and Schedule (8.3%) and Effort attributes (7.9%). The eight remaining groups of attributes only have 50 total variable occurrences (7.2%) in the entire set of papers.

Usage of the Project and Grouping attributes groups has declined slowly in percentage terms, but that is because more variables from other groups are used as well. Some groups have been used consistently since 2005, such as the two aforementioned and Sizing; other groups were used for a period but are no longer receiving the same level of attention, such as Schedule and Effort attributes; the Size attributes group has been gradually playing a more important role: the Base Functional Components (INC, OC, EC, FC, and IFC) comprise 37% of occurrences in effort estimation methods in 2013.

Characteristics of independent variables

Understanding the characteristics of project variables is critical in software development effort estimation. Disregarding them can lead to inaccurate estimation, and eventually project failure. In this sense, it is important for academics and practitioners to have a thorough knowledge of the ISBSG dataset before using it. RQ2 has described the characteristics of the most used ISBSG variables following the ISBSG groups framework, excluding cases with low data quality ratings (C or D) of ISBSG dataset Release 11.

The Project attributes group includes nominal variables with a generic technical project perspective of the project (such as LT (3rd), DP (4th), PPL (8th), or UM (19th)), and variables specifically related to DP such as 1DBS (17th). LT is used more often than PPL, except when more information about the specific used programming language is required, such as in EbA estimation methods. 1DBS and UM have around 40% of missing values, which may explain their lower use. This situation is more obvious for Case Tool Used (28th) and How Methodology Acquired (29th), which have 60.2% and 62.9% of missing values respectively.

Grouping attributes (19.8%) combine nominal context variables that are clearly determined at an early stage of the project. OT (7th), AT (10th), and BAT (11th) can be considered as organisational variables. All these variables present many different discrete values, which are usually regrouped if they are to be used in regression analysis. Indeed, it is important with any estimation method to consider whether all the categories are necessary, or if any of them should be combined, to reduce the complexity of the nominal predictor variables [S34]. This group also includes project-specific variables such as DT (2nd), whose categories have sometimes been combined [S14, S73]. Package Customisation (22nd) and Degree of Customisation (57th) are also project-specific variables, missing 69.3% and 98.4% of their values respectively.

When Project and Grouping attributes are considered, the more missing values a variable has, the less used this variable is compared to the other variables within the same group. Besides, Tsunoda et al. [S78] consider that in organisations that collect other project data in detail, it might not be preferable to use variables like DT, DP, and LT as independent variables in estimation models using early phase effort, because that could possibly worsen the accuracy of the model.

Variables related to project size are included in the groups Sizing, Size attributes, and Size Other than FSM. Together, these variables account for nearly one third (30.6%) of all variable occurrences used in effort estimation models in the 107 papers. FS (1st) belongs to the Sizing group, and is now preferred to AFP (5th) despite having many more missing values (only 3.5% for AFP versus 29.1% for FS). Using adjusted function points, instead of unadjusted function points, as a predictor of effort makes little or no difference in estimation accuracy ([S20], (Kitchenham, 1992; Jeffery and Stathis, 1996; Kemerer, 1987)). In this same group, the variable Count Approach (21st) is a relevant variable for selecting projects. It has no missing values. The Size attributes group provides disaggregated project size information: the breakdown of FS into inputs (INC, 12th), outputs (OC, 13rd), enquiries (EC, 14th), files (FC, 15th), and interfaces (IFC, 16th), with around 67% of values missing for each of these variables.

The Schedule group variables are related to the effort in the different project phases, and the duration of the project such as PET (9th). Variables in this group have a higher presence in ML methods.

Effort attributes record data about the human resource perspective of project management, such as team size and resource level. ATS (20th) may arguably be more interesting than MTS (6th), however it has many more missing values (88%) than MTS (70.1%). Besides, MTS has been identified as one of the most important explanatory variables for effort when using Function Points [S31, S34]. Finally, RL (18th) is more often used in the filtering process than in the model itself, and has no missing data.

Reasons for selecting independent variables

ISBSG independent variables are project characteristics believed to be significant for the goals of a study [S3, S96, S105, S107], and to play a crucial role to improve the accuracy of effort estimation [S106]. While there is no standard method for selecting independent variables [S6], these are some of the reasons:

- Often, variables are chosen because the authors believe a priori that they affect productivity and effort. ISBSG recommendations to use FS, DT, LT, and DP are often followed [S52, S97, S105]. Some authors [S14, S72, S89, S106] include independent variables because they were reported in several studies to potentially influence software effort.

- Some authors [S22, S13] use the same set of independent variables as previous studies. Several papers rely on influential studies [S34, S43, S77, S103] to drive their selection of independent variables. The authors' own previous research has been influential in making this decision [S39, S45, S67]. Some others [S86] base their choice on their own previous experience, but do not explicitly mention the foundation for their decision.
- In studies that use multiple datasets, some independent variables could be chosen or excluded depending on whether they are common to several datasets [S3].
- Some authors use statistical approaches to select independent variables. Song et al. (2008) use an information-theoretic approach. [S64] base their decision on prediction accuracy and the relation between input and output; they use a feature selection algorithm using fuzzy logic. [S76] proposes a framework that applies a set of statistical approaches to the dataset in order to identify relevant variables. [S56] presents a method to statistically evaluate the relationship between useful project features and target features such as effort. Statistical tests of Pearson correlation, and one-way ANOVA, have been used to examine the significance between the independent variables and the actual effort, to eliminate irrelevant variables [S1, S106].
- The level of missing values [S1, S107] and the presence of outliers [S64] have also been considered.
- The model building approach can influence the selection of independent variables. For example, the independence of explanatory variables is a common assumption when building a multivariable model [S76, S107]; OLS regression requires data with a normal distribution of the residuals and also a steady variance [S1].

Independent variables used with different estimation methods

Considering the effort estimation methods, regression methods are the most common family (71% of papers). They may be used alone (38%), or in combination with other methods (33%), mainly for contrasting the performance of proposed methods. The next most used family is machine learning methods, used in 37% of papers (11% exclusively, 26% with other methods as well). The third most used family is Estimation by Analogy, used in 21% of papers (6% exclusively, 15% with other methods as well). These results on the distribution of estimation methods are in agreement with the broader field of empirical software engineering (Jorgensen and Shepperd, 2007). Finally, 22% of the papers use a mixture of methods such as combination of estimates, function points, fuzzy, simulation, survival analysis, multiple criteria linear programming, production function, sequential quadratic programming, case studies, Particle Swarm Optimisation (PSO), etc.

These estimations methods have used, in general, an average of 6.5 variables. The papers that used ML exclusively employed fewer variables than any other family of methods (4.7). In contrast, papers that used EbA usually consider more variables to generate the models. Papers that use regression methods exclusively or in association with other types of effort estimation methods use an average of 5.7 and 5.2 variables respectively.

In regression-based estimation methods, the variable FS is preferred to AFP. The Size attributes group of variables, which breaks down the size of a project into its base functional components, are used less. Machine learning methods make more use of the Size attributes group of variables, despite the high percentage of missing values (around 67%). The use of the Size attributes group of variables is even higher with EbA methods, more than 20%, with a corresponding reduction in the use of FS. EbA methods generally use more nominal variables; another observation relates to the relative usage of LT and PPL: PPL is used more with EbA than with other estimation methods. When regression is used exclusively, it is not surprising that the four most frequently used variables (FS, DT, LT, and DP, i.e. the ones recommended by ISBSG) dominate, particularly FS.

Based on these findings, we present some guidelines in Table 13 for selecting independent variables in

effort estimation methods.

Table 13

Drivers guiding the selection of independent variables for effort estimation methods.

Factor	Description	Evidence
ISBSG recommendations	The four most used independent variables are FS, DT, LT, and DP, which is consistent with the fact that these are the variables whose usage is recommended by ISBSG for data partitioning (ISBSG, 2009a).	23.4% use FS, DT, LT, and DP; 64.5% of the selected papers use at least two variables that ISBSG recommends, 85% of the papers use at least one of them, and only 15% do not use any of them.
Customary	It is customary for researchers to include specific variables in their models when there is evidence that other authors have already used them for effort estimation purposes.	Some authors [S14, S72, S89, S106] include independent variables because they were previously reported to potentially influence software effort. Some authors [S22, S13] use the same set of independent variables as previous studies. Others rely on influential studies [S34, S43, S77, S103]. Appendix C of this paper presents the set of selected papers and the occurrences of the 20 most used variables.
Relevance	Each prospective independent variable should convey pertinent meaning for effort estimation. This meaning is generally shared by the variables of the same group. The most representative variables should be identified and selected while variables and projects with no direct or apparent effect on software estimation should be ignored [S39, S84].	From the large number of project variables provided in the dataset, in [S34] a small set of categorical variables (DT, DP, LT, UM, OT, BAT, and AT) that are intuitively expected to affect the effort were selected. Many authors [S39, S45, S67, S86] base their choice on their own previous experience.
Datasets compatibility	Some independent variables could be chosen or excluded, depending on whether they are common to multiple datasets.	In [S3], variables were only selected if they were common to both datasets analysed. Comparison of estimates across multiple datasets requires these datasets to have comparable variables.
Feature Selection	Some authors use statistical approaches to select independent variables.	A usual approach is to use statistical tests of Pearson correlation, and one-way ANOVA to examine the significance between the independent variables and the actual effort, to eliminate irrelevant variables [S1, S106]. [S64] use a feature selection algorithm based on fuzzy logic.
Missing data	Missing values in relevant variables can result in a significant reduction in the data used to build effort estimation models and leading to biased models [S15, S91]. However, some recent studies have presented an increased awareness of the importance of treating missing data to improve effort estimation consistency (Twala et al., 2005).	In as many as 35% of papers, projects that have missing values for any independent variable are excluded from the dataset [S33, S73]. [S31] uses imputation methods to handle missing data and to increase sample size. When Project and Grouping attributes are considered, the more missing values a variable has, the less used this variable is when compared to the rest of variables within

		the same group.
Statistical distribution	The selection process can be influenced by the analysis of skewness, kurtosis [S31], and outliers. The lack of normality of some variables, even after log-transformation, may require the use of non-parametric methods. Note that the log transformation can be valuable for helping to meet the assumptions of inferential statistics. On the other hand, a small proportion of outliers can affect even simple analyses.	For example, [S21] investigates the influence of outlier elimination upon the accuracy of software effort estimation. [S22, S72, S75] also study the outliers issue.
Categorisation	It is important to consider whether all of the categories for each nominal variable are necessary, or if any of them should be combined to reduce the complexity of the nominal predictor variables [S34].	OT, AT, and BAT present many different discrete values and in some instances, categories that were found in no more than five records in the entire dataset are merged under the label of 'Other' [S37]. Even DT categories have been subject to combinations [S14, S73].
Estimation methods	ISBSG recommended variables are most used in traditional methods such as regression. Other alternatives are most suitable for methods such as EbA or Machine Learning approaches. Besides, the number and type of the selected variables differ considering the estimation method used.	When regression is used exclusively, the four most frequently used variables have even a greater presence. In particular, FS and DT increase their percentage of participation by 13.9% and 7.9% respectively. The papers that used ML exclusively employ fewer variables than any other family of methods (average of 4.7). ML methods make more use of the Size attributes group of variables, despite the high percentage of missing values (around 67%). In the same way, EbA methods also tend to use the Size attributes, and they employ more variables and more nominal variables to construct the models.
Granularity level	The level of information detail that is required is one of the criteria to select one variable from a group. Showing few details contributes to apply the same definition in relation to a particular concept and to reduce the probability of having not comparable historical data.	In regression methods, the variable FS is preferred rather than the Size attributes group of variables that break down the size of a project into base functional components. In EbA, a balance is observed in the relative usage of LT and PPL. [S78] considers that in organisations that collect other project data in detail, it might not be preferable to use variables like DT, DP, and LT as independent variables in the estimation models using early phase effort, because that could possibly worsen the accuracy of the model.
Parsimony	This criterion tries to keep the estimation method as simple as possible, eliminating redundant variables. The resulting models will contain fewer variables, and will be also more stable since potential collinearity between variables will be reduced. Generally, a limited set of highly predictive variables is easier to interpret and preferred over a more complex model.	When two or more independent variables contain redundant information, only one is to be considered ([S73], (Bibi et al., 2008)).

4.2. Study limitations

It is important to consider that the results obtained from a systematic mapping study could be affected by the researchers conducting the review, by the search term selected, and by the chosen time frame (Elberzhager et al., 2012). The main limitations of this mapping study are discussed below.

Concerning the search strategy, the choice of search term ("ISBSG", in the title, abstract or body of the paper) should not miss anything. However, it is possible that some studies were missed due to our choice of databases to search. We searched four: other studies may have been found if further databases were searched. We believe this risk is low, because the databases that we searched are the major search engines and digital libraries most frequently used in systematic literature reviews performed by the software engineering community. Also, because the search engines may not have indexed them yet, some more recent studies may be missing. Finally, only papers dealing with effort or productivity were retained to reach the final collection of 107 papers. This information is not always provided in the abstract or keywords, so it was necessary to read not only the introduction and conclusion sections but also other sections of the primary studies. The risk of missing a relevant paper was mitigated by the first two authors independently reading all of the papers found in the initial search.

While extracting from the papers the variables that have been used to generate effort estimation models, some difficulties were encountered. The authors of the identified references used diverse terms to identify the variables, mainly, but not always, due to using different versions of the ISBSG dataset. This problem was mitigated by performing an additional cross-check.

The calculations in this paper were performed on ISBSG projects from Release 11 with a Data Quality Rating of A or B. The focus on this particular Release and these Data Quality Ratings could produce a bias, compared to analysing the whole dataset. However, since the common practice adopted by researchers (following the ISBSG's recommendation) is to filter out projects with lower quality ratings, the decision is justified.

There is also a limitation concerning RQ3, which presents how the level of usage of ISBSG variables is related to the type of estimation methods that have been used in the 107 papers. When addressing this question we only considered the 20 most frequent independent variables (listed in Table 3). If all of the variables listed in Appendix B were considered, might the results for RQ3 have been different? We believe not, due to the great difference in usage between the first 20 and the rest of the variables, as seen in Figure 6. On the other hand, the fact that some studies were classified in more than one family concerning the estimation methods appears as a potential problem, since no evidence of which variables have been used in each method was collected.

4.3. Implications for research and practice

We have analysed the use by researchers of ISBSG variables (which variables have been used for which purposes, why, in what contexts, and which have been considered most important) for effort estimation. The knowledge gained can help researchers and practitioners to make informed decisions about the selection of ISBSG variables for their effort estimation models, and can help to identify what can be done better in future research and practice. Some of these insights are presented next, organised by topics, such as data quality, reproducibility of the studies, increased knowledge of those variables and their relevance for effort estimation, and contrasting and learning from previous experience.

It is essential for any analysis to be based on sound data:

- Projects with low quality ratings should be excluded.

- The projects analysed in any given study should have comparable definitions for critical variables, such as size and effort. For size, this means considering Count Approach, so that size measures are comparable. For effort, it means considering both normalisation and Resource Level, so that effort measures cover comparable life cycle phases and project participants.
- Information presented in Sections 3.1.1 and 3.1.2 can help users of the ISBSG dataset to select appropriate projects (rows) and variables (columns) for analysis, considering also aspects such as outliers, missing data and particular project types in which researchers may be interested.
- Research practice and/or reporting can improve in this respect: as noted in Section 3.1.2, several researchers have used ISBSG data without indicating that they considered its quality or comparability.

For researchers, it is important that studies are reproducible. In this respect, we note that reporting of research using the ISBSG dataset can improve:

- In the ISBSG dataset the meaning of “size” and “effort” can be ambiguous (see Sections 3.1.1 to 3.1.3). It is important for authors to specify clearly which size and effort variables they use.
- Researchers should specify clearly the filtering process they use, including specific definitions of variables that could be ambiguous.
- Researchers should specify which Release of the ISBSG dataset they use, so that others wishing to replicate their process know the starting set of projects to which filtering was applied.

For both researchers and practitioners, it is important to understand the meaning, range, and distribution of the data:

- Section 3.2 complements and extends ISBSG’s demographic summary of the dataset (ISBSG, 2009b), selecting only those projects with Data Quality Rating A or B, for the 20 most used independent variables identified in this study. The section summarises the concepts they represent, their range and distribution of values, and the extent of missing data.
- Further, for researchers wishing to reduce the problem of heterogeneity in the ISBSG dataset, Section 3.2 (concerning the meaning and values of data) and 3.1.2 (concerning filtering done by past researchers) can be helpful.

For practitioners using ISBSG data to help them estimate the effort for a software project, it is essential to use data that is of high quality, comparable, and relevant:

- Relevance relates to which variables provide useful information to an organisation for its own effort estimation purposes. Section 3.1.4’s analysis of which variables have been considered and why, and Section 3.2’s analysis of the meaning and values of ISBSG variables, can help. ISBSG’s recommendation to use FS, DT, LT and DP is commonly followed.

For researchers, Sections 3.1.4, 3.2, 3.3, and Appendix C constitute a useful background to contrast and learn from previous experience:

- A researcher who plans to select a variable for an estimation model can see references that have previously worked with this variable, which variables have been more or less often studied, and which variables tend to be used more or used less with several estimation methods.
- Researchers may even infer (by implication from the variables considered) how interest in different research topics has changed over time.

All in all, this study can contribute to a better understanding of the usage of the ISBSG dataset by the research community, to the benefit of both the research and practitioner communities.

5. Conclusion and future work

This work presents the results of a systematic mapping study about the usage of ISBSG variables to elaborate effort estimation models, to the end of 2013. After a searching and filtering process, 107 papers were identified that produce effort estimates and list the independent variables. The analysis of these papers describes how and to what extent ISBSG variables and groups of variables have been used in the software engineering literature to build effort estimation models.

The answers to the research questions have provided these valuable results:

- The 71 ISBSG variables that have been used as independent variables to construct effort estimation models have been listed, together with the number of their occurrences.
- The 20 most used independent variables have been described, individually and in related groups, presenting their meaning, range and distribution of values, along with their relationships and some underlying dependencies. This part of the analysis considered projects with a Data Quality Rating of A or B in Release 11 of the ISBSG dataset.
- A matrix shows which of these 20 most used independent variables were used in the collection of the 107 selected references provided.
- The dependent variables and filtering variables used in this collection of papers have been analysed.
- Regarding the estimation methods used in the selected studies, regression-based estimation models are the most frequently used, sometimes alone but often together with other methods in order to contrast the results that have been obtained by other methods. Machine learning methods and EbA methods are also frequently used with the ISBSG dataset.
- We have explored the usage of independent variables with these estimation methods. Differences were identified in how many variables, and which variables, tend to be used with different estimation methods.
- Finally, several factors that can guide the selection of variables have been described.

Thereby, this paper presents some insights that may be useful to guide future effort estimation studies. All in all, improving the knowledge of the most frequently used variables, by considering aspects such as their relevance, redundancy and missing data rate, should contribute to increasing the reliability of effort estimation using the ISBSG dataset.

In the future, the authors intend to address several interesting questions that have arisen in this study. How have the perceived relevance of variables, their quality and the structure of groups of variables evolved when different ISBSG releases are considered? A variable may lose importance as an effort driver over time [S67]: is this reflected in the usage of variables in research studies? What are the relationships of ISBSG variables with similar variables in other datasets and which other variables may affect effort when considering other datasets apart from ISBSG? What is the missing values threshold that will affect the level of variable usage? Which are the most accurate models to estimate effort? How and to what extent is the accuracy of the models related to the usage of particular independent variables (e.g. does using 1, 2, 3 or 4 of the four key variables recommended by ISBSG make a difference)?

To summarise, this systematic mapping study has collected and classified research papers that used ISBSG variables in software effort estimation. The answers to the research questions provide, with the above mentioned limitations, an understanding of how and what extent these data fields have been used for this aim. Lessons learned can help researchers and practitioners to make effective use of the ISBSG dataset for effort estimation, and can help research practice concerning the planning, data selection, reproducibility, and reporting of studies.

5. Bibliography

Abran, A., Robillard, P.N., 1996. Function points analysis: An empirical study of its measurement processes. Softw. Eng. IEEE Trans. On 22, 895–910.
Acuña, S.T., Castro, J.W., Dieste, O., Juristo, N., 2012. A systematic mapping study on the open source software development process, in: Evaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference on. pp. 42–46.
Bakır, A., Turhan, B., Bener, A.B., 2010. A new perspective on data homogeneity in software cost estimation: a study in the embedded systems domain. Softw. Qual. J. 18, 57–80. doi:10.1007/s11219-009-9081-z
Bibi, S., Tsoumakas, G., Stamelos, I., Vlahavas, I., 2008. Regression via Classification applied on software defect estimation. Expert Syst. Appl. 34, 2091–2101. doi:10.1016/j.eswa.2007.02.012
Budgen, D., Turner, M., Brereton, P., Kitchenham, B., 2008. Using mapping studies in software engineering, in: Proceedings of PPIG. pp. 195–204.
Deng, K., MacDonell, S.G., 2008. Maximising data retention from the ISBSG repository, in: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. British Computer Society, pp. 21–30.
Dolado, J.J., Rodríguez, D., Riquelme, J., Ferrer-Troyano, F., Cuadrado, J.J., 2007. A Two-Stage Zone Regression Method for Global Characterization of a Project Database. Adv. Mach. Learn. Appl. Softw. Eng. 1.
Elberzhager, F., Münch, J., Nha, V.T.N., 2012. A systematic mapping study on the combination of static and dynamic quality assurance techniques. Inf. Softw. Technol. 54, 1–15.
Fernández-Diego, M., González-Ladrón-de-Guevara, F., 2014. Potential and limitations of the ISBSG dataset in enhancing software engineering research: A mapping review. Inf. Softw. Technol. 56, 527–544. doi:10.1016/j.infsof.2014.01.003
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, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering, PROMISE '10. ACM, New York, NY, USA, pp. 13:1–13:9. doi:10.1145/1868328.1868348
González-Ladrón-de-Guevara, F., Fernández-Diego, M., 2014. ISBSG Variables Most Frequently Used for Software Effort Estimation: A Mapping Review, in: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14. ACM, New York, NY, USA, pp. 42:1–42:4. doi:10.1145/2652524.2652550
Hill, P., 2010. Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration. McGraw Hill Professional.
ISBSG, 2009a. Guidelines for use of the ISBSG data. Estimating Benchmarking & Research Suite Release 11.
ISBSG, 2009b. ISBSG Release 11 Repository Demographics.
ISBSG, 2009c. ISBSG Release 11 Glossary of Terms for Application Software Maintenance and Support V2.2.
ISBSG, 2009d. ISBSG Repository Data Release 11. Field Descriptions.
ISO, 2002. ISO/IEC 20968:2002. Software engineering – Mk II Function Point Analysis – Counting Practices Manual.
ISO, 2005. ISO/IEC 24570:2005. Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis.
ISO, 2008. ISO/IEC 29881:2008. Information technology – Software and systems engineering – FiSMA 1.1 functional size measurement method.
ISO, 2009. ISO/IEC 20926:2009. Software and systems engineering – Software measurement – IFPUG functional size measurement method 2009.
ISO, 2011. ISO/IEC 19761:2011. Software engineering – COSMIC: A functional size measurement method.

Jeffery, R., Stathis, J., 1996. Function point sizing: Structure, validity and applicability. Empir. Softw. Eng. 1, 11–30. doi:10.1007/BF00125809
Jorgensen, M., Shepperd, M., 2007. A systematic review of software development cost estimation studies. Softw. Eng. IEEE Trans. On 33, 33–53.
Kemerer, C.F., 1987. An empirical validation of software cost estimation models. Commun ACM 30, 416–429. doi:10.1145/22899.22906
Kitchenham, B., 1992. Empirical studies of assumptions that underlie software cost-estimation models. Inf. Softw. Technol. 34, 211–218. doi:10.1016/0950-5849(92)90077-3
Kitchenham, B., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering (Technical No. EBSE-2007-01). Software Engineering Group, School of Computer Science and Mathematics, Keele University.
Liebchen, G.A., Shepperd, M., 2008. Data sets and data quality in software engineering, in: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering (PROMISE). ACM, Leipzig, Germany, pp. 39–44. doi:10.1145/1370788.1370799
Li, J., Al-Emran, A., Ruhe, G., 2007. Impact Analysis of Missing Values on the Prediction Accuracy of Analogy-based Software Effort Estimation Method AQUA, in: Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on. pp. 126 –135. doi:10.1109/ESEM.2007.10
Menzies, T., Rees-Jones, M., Krishna, R., Pape, C., 2015. The Promise Repository of Empirical Software Engineering Data; http://openscience.us/repo. North Carolina State University, Department of Computer Science.
Moses, J., Farrow, M., Parrington, N., Smith, P., 2006. A productivity benchmarking case study using Bayesian credible intervals. Softw. Qual. J. 14, 37–52. doi:10.1007/s11219-006-6000-4
NESMA, 2014. FPA according to NESMA and IFPUG – the present situation. http://nesma.org/themes/sizing/nesma-and-ifpug/. Viewed 27 May 2015.
Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering. p. 1.
Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18.
Song, Q., Shepperd, M., Chen, X., Liu, J., 2008. Can k-NN imputation improve the performance of C4.5 with small software project data sets? A comparative evaluation. J. Syst. Softw. 81, 2361–2370. doi:10.1016/j.jss.2008.05.008
Stensrud, E., Foss, T., Kitchenham, B., Myrtveit, I., 2002. An empirical validation of the relationship between the magnitude of relative error and project size, in: Software Metrics, 2002. Proceedings. Eighth IEEE Symposium on. pp. 3–12.
Twala, B., Cartwright, M., Shepperd, M., 2005. Comparison of various methods for handling incomplete data in software engineering databases, in: Empirical Software Engineering, 2005. 2005 International Symposium on. pp. 105 –114. doi:10.1109/ISESE.2005.1541819
Wen, J., Li, S., Lin, Z., Hu, Y., Huang, C., 2012. Systematic literature review of machine learning based software development effort estimation models. Inf. Softw. Technol. 54, 41–59. doi:10.1016/j.infsof.2011.09.002
Zhang, H., Babar, M.A., Tell, P., 2011. Identifying relevant studies in software engineering. Inf. Softw. Technol. 53, 625–637. doi:10.1016/j.infsof.2010.12.010

Appendix A

Final set of papers sorted alphabetically by title.

[S1] S.-J. Huang, N.-H. Chiu, Y.-J. Liu, A comparative evaluation on the accuracies of software effort estimates from clustered data, Inf. Softw. Technol. 50 (2008) 879–888. doi:10.1016/j.infsof.2008.02.005.

[S2] W. Zhang, Y. Yang, Q. Wang, A comparative study of absent features and unobserved values in software effort data, Int. J. Softw. Eng. Knowl. Eng. 22 (2012) 185–202. doi:10.1142/S0218194012400025.
[S3] R. Jeffery, M. Ruhe, I. Wieczorek, A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data, Inf. Softw. Technol. 42 (2000) 1009–1016. doi:10.1016/S0950-5849(00)00153-1.
[S4] A.B. Nassif, M. Azzeh, L.F. Capretz, D. Ho, A comparison between decision trees and decision tree forest models for software development effort estimation, in: 2013 Third International Conference on Communications and Information Technology (ICCIT), 2013: pp. 220–224. doi:10.1109/ICCITechnology.2013.6579553.
[S5] J. Li, G. Ruhe, A. Al-Emran, M.M. Richter, A flexible method for software effort estimation by analogy, Empir Software Eng. 12 (2007) 65–106. doi:10.1007/s10664-006-7552-4.
[S6] V.K. Bardsiri, D.N.A. Jawawi, S.Z.M. Hashim, E. Khatibi, A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons, Empir Software Eng. (2013) 1–28. doi:10.1007/s10664-013-9241-4.
[S7] E. Papatheocharous, A.S. Andreou, A Hybrid Software Cost Estimation Approach Utilizing Decision Trees and Fuzzy Logic, Int. J. Softw. Eng. Knowl. Eng. 22 (2012) 435–465. doi:10.1142/S0218194012500106.
[S8] W. Xia, L.F. Capretz, D. Ho, F. Ahmed, A new calibration for Function Point complexity weights, Inf. Softw. Technol. 50 (2008) 670–683. doi:10.1016/j.infsof.2007.07.004.
[S9] A. Bakır, B. Turhan, A.B. Bener, A new perspective on data homogeneity in software cost estimation: a study in the embedded systems domain, Software Qual J. 18 (2010) 57–80. doi:10.1007/s11219-009-9081-z.
[S10] O. Adalier, A. Ugur, S. Korukoglu, K. Ertas, A new regression based software cost estimation model using power values, in: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao (Eds.), Intelligent Data Engineering and Automated Learning - Ideal 2007, Springer-Verlag Berlin, Birmingham, ENGLAND, 2007: pp. 326–334.
[S11] L.L. Minku, X. Yao, A principled evaluation of ensembles of learning machines for software effort estimation, in: Proceedings of the 7th International Conference on Predictive Models in Software Engineering, ACM, New York, NY, USA, 2011: pp. 9:1–9:10. doi:10.1145/2020390.2020399.
[S12] V.K. Bardsiri, D.N.A. Jawawi, S.Z.M. Hashim, E. Khatibi, A PSO-based model to increase the accuracy of software development effort estimation, Software Qual J. 21 (2013) 501–526. doi:10.1007/s11219-012-9183-x.
[S13] E. Mendes, C. Lokan, R. Harrison, C. Triggs, A replicated comparison of cross-company and within-company effort estimation models using the ISBSG database, in: Software Metrics, 2005. 11th IEEE International Symposium, 2005: pp. 1 –10. doi:10.1109/METRICS.2005.4.
[S14] V.K.Y. Chan, W.E. Wong, T.F. Xie, A Statistical Methodology to Simplify Software Metric Models Constructed Using Incomplete Data Samples, International Journal of Software Engineering and Knowledge Engineering. 17 (2007) 689–707. doi:10.1142/S0218194007003495.
[S15] Y.F. Li, M. Xie, T.N. Goh, A study of the non-linear adjustment for analogy based software cost estimation, Empir Software Eng. 14 (2009) 603–643. doi:10.1007/s10664-008-9104-6.
[S16] S. Amasaki, A Study on the Use of COSMIC BFCs for Effort Estimation, in: 2012 Fourth International Workshop on Empirical Software Engineering in Practice (IWESEP), 2012: pp. 58–60. doi:10.1109/IWESEP.2012.11.
[S17] J.J. Cuadrado-Gallego, M.-A. Sicilia, An algorithm for the generation of segmented parametric software estimation models and its empirical evaluation, Comput. Inform. 26 (2007) 1–15.

[S18] L.L. Minku, X. Yao, An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation, in: Proceedings of the 9th International Conference on Predictive Models in Software Engineering, ACM, New York, NY, USA, 2013: pp. 8:1–8:10. doi:10.1145/2499393.2499396.
[S19] M. Hericko, A. Zivkovic, I. Roman, An approach to optimizing software development team size, Inf. Process. Lett. 108 (2008) 101–106. doi:10.1016/j.ipl.2008.04.014.
[S20] C.J. Lokan, An empirical analysis of function point adjustment factors, Information and Software Technology. 42 (2000) 649–659. doi:10.1016/S0950-5849(00)00108-7.
[S21] Y.-S. Seo, K.-A. Yoon, D.-H. Bae, An empirical analysis of software effort estimation with outlier elimination, in: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering, ACM, New York, NY, USA, 2008: pp. 25–32. doi:10.1145/1370788.1370796.
[S22] M. Tsunoda, T. Kakimoto, A. Monden, K. Matsumoto, An empirical evaluation of outlier deletion methods for analogy-based cost estimation, in: Proceedings of the 7th International Conference on Predictive Models in Software Engineering, ACM, New York, NY, USA, 2011: pp. 17:1–17:10. doi:10.1145/2020390.2020407.
[S23] J.J. Cuadrado-Gallego, M.A. Sicilia, M. Garre, D. Rodriguez, An empirical study of process-related attributes in segmented software cost-estimation relationships, J. Syst. Softw. 79 (2006) 353–361. doi:10.1016/j.jss.2005.04.040.
[S24] P.C. Pendharkar, J.A. Rodger, G.H. Subramanian, An empirical study of the Cobb–Douglas production function properties of software development effort, Information and Software Technology. 50 (2008) 1181–1188. doi:10.1016/j.infsof.2007.10.019.
[S25] P. Chatzipetrou, E. Papatheocharous, L. Angelis, A.S. Andreou, An Investigation of Software Effort Phase Distribution Using Compositional Data Analysis, in: 2012 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2012: pp. 367–375. doi:10.1109/SEAA.2012.50.
[S26] M. Azzeh, D. Neagu, P.I. Cowling, Analogy-based software effort estimation using Fuzzy numbers, J. Syst. Softw. 84 (2011) 270–284. doi:10.1016/j.jss.2010.09.028.
[S27] J. Li, G. Ruhe, Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+, Empir Software Eng. 13 (2008) 63–96. doi:10.1007/s10664-007-9054-4.
[S28] C. Lokan, E. Mendes, Applying moving windows to software effort estimation, in: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, Washington, DC, USA, 2009: pp. 111–122. doi:10.1109/ESEM.2009.5316019.
[S29] W.E. Wong, J. Zhao, V.K.Y. Chan, Applying statistical methodology to optimize and simplify software metric models with missing data, in: Proceedings of the 2006 ACM Symposium on Applied Computing, ACM, New York, NY, USA, 2006: pp. 1728–1733. doi:10.1145/1141277.1141687.
[S30] Y.-S. Seo, D.-H. Bae, R. Jeffery, AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning, Information and Software Technology. 55 (2013) 1710–1725. doi:10.1016/j.infsof.2013.03.007.
[S31] J. Moses, M. Farrow, Assessing Variation in Development Effort Consistency Using a Data Source with Missing Data, Software Qual J. 13 (2005) 71–89. doi:10.1007/s11219-004-5261-z.
[S32] A. Živkovič, I. Rozman, M. Heričko, Automated software size estimation based on function points using UML models, Information and Software Technology. 47 (2005) 881–890. doi:10.1016/j.infsof.2005.02.008.
[S33] S. Bibi, I. Stamelos, G. Gerolimos, V. Kollias, BBN based approach for improving the software development process of an SME - a case study, J. Softw. Maint. Evol.-Res. Pract. 22 (2010) 121–140. doi:10.1002/spip.418.

[S34] L. Angelis, I. Stamelos, M. Morisio, Building a software cost estimation model based on categorical data, in: Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International, 2001: pp. 4 –15. doi:10.1109/METRIC.2001.915511.
[S35] L.L. Minku, X. Yao, Can cross-company data improve performance in software effort estimation?, in: Proceedings of the 8th International Conference on Predictive Models in Software Engineering, ACM, New York, NY, USA, 2012: pp. 69–78. doi:10.1145/2365324.2365334.
[S36] P. Sentas, L. Angelis, Categorical missing data imputation for software cost estimation by multinomial logistic regression, J. Syst. Softw. 79 (2006) 404–414. doi:10.1016/j.jss.2005.02.026.
[S37] S. Bibi, I. Stamelos, L. Angelis, Combining probabilistic models for explanatory productivity estimation, Information and Software Technology. 50 (2008) 656–669. doi:10.1016/j.infsof.2007.06.004.
[S38] N. Mittas, L. Angelis, Combining regression and estimation by analogy in a semi-parametric model for software cost estimation, in: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, New York, NY, USA, 2008: pp. 70–79. doi:10.1145/1414004.1414017.
[S39] N. Mittas, L. Angelis, Comparing cost prediction models by resampling techniques, Journal of Systems and Software. 81 (2008) 616–632. doi:10.1016/j.jss.2007.07.039.
[S40] N. Mittas, L. Angelis, Comparing Software Cost Prediction Models by a Visualization Tool, in: Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference, 2008: pp. 433 –440. doi:10.1109/SEAA.2008.23.
[S41] S. Berlin, T. Raz, C. Glezer, M. Zviran, Comparison of estimation methods of cost and duration in IT projects, Information and Software Technology. 51 (2009) 738–748. doi:10.1016/j.infsof.2008.09.007.
[S42] C.-J. Hsu, C.-Y. Huang, Comparison of weighted grey relational analysis for software effort estimation, Software Qual J. 19 (2011) 165–200. doi:10.1007/s11219-010-9110-y.
[S43] C. Lokan, E. Mendes, Cross-company and single-company effort models using the ISBSG database: a further replicated study, in: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ACM, New York, NY, USA, 2006: pp. 75–84. doi:10.1145/1159733.1159747.
[S44] K. Dejaeger, W. Verbeke, D. Martens, B. Baesens, Data Mining Techniques for Software Effort Estimation: A Comparative Study, Software Engineering, IEEE Transactions on. 38 (2012) 375 –397. doi:10.1109/TSE.2011.55.
[S45] M. Fernandez-Diego, J.-M. Torralba-Martinez, Discretization methods for NBC in effort estimation: An empirical comparison based on ISBSG projects, in: 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2012: pp. 103–106. doi:10.1145/2372251.2372268.
[S46] C. Gencel, L. Buglione, Do Base Functional Component types affect the relationship between software functional size and effort?, in: J.J. Cuadrado-Gallego, R. Braungarten, R.R. Dumke, A. Abran (Eds.), Software Process and Product Measurement, Springer-Verlag Berlin, Palma de Majorque, SPAIN, 2008: pp. 72–85.
[S47] Y. Kultur, E. Kocaguneli, A.B. Bener, Domain specific phase by phase effort estimation in software projects, in: Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on, 2009: pp. 498 –503. doi:10.1109/ISCIS.2009.5291873.
[S48] C. Comstock, Z. Jiang, J. Davies, Economies and diseconomies of scale in software development, Journal of Software Maintenance and Evolution: Research and Practice. 23 (2011) 533–548. doi:10.1002/smr.526.
[S49] M. Castejon-Limas, J. Ordieres-Mere, A. Gonzalez-Marcos, V. Gonzalez-Castro, Effort estimates through project complexity, Ann. Oper. Res. 186 (2011) 395–406. doi:10.1007/s10479-010-0776-0.

[S50] J. Aziz, F. Ahmed, M.S. Laghari, Empirical Analysis of Team and Application Size on Software Maintenance and Support Activities, in: Information Management and Engineering, 2009. ICIME '09. International Conference on, 2009: pp. 47 –51. doi:10.1109/ICIME.2009.51.
[S51] D. Rodriguez, M.A. Sicilia, E. Garcia, R. Harrison, Empirical findings on team size and productivity in software development, J. Syst. Softw. 85 (2012) 562–570. doi:10.1016/j.jss.2011.09.009.
[S52] L.L. Minku, X. Yao, Ensembles and locality: Insight on improving software effort estimation, Information and Software Technology. 55 (2013) 1512–1528. doi:10.1016/j.infsof.2012.09.012.
[S53] I. Stamelos, L. Angelis, M. Morisio, E. Sakellaris, G.L. Bleris, Estimating the development cost of custom software, Information & Management. 40 (2003) 729–741. doi:10.1016/S0378-7206(02)00099-X.
[S54] Q. Liu, W.Z. Qin, R. Mintram, M. Ross, Evaluation of preliminary data analysis framework in software cost estimation based on ISBSG R9 Data, Softw. Qual. J. 16 (2008) 411–458. doi:10.1007/s11219-007-9041-4.
[S55] A.S. Andreou, E. Papatheocharous, C. Skouroumounis, Evolving Conditional Value Sets of Cost Factors for Estimating Software Development Effort, in: Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, 2007: pp. 165 –172. doi:10.1109/ICTAI.2007.74.
[S56] J. Keung, B. Kitchenham, Experiments with Analogy-X for Software Cost Estimation, in: Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on, 2008: pp. 229 –238. doi:10.1109/ASWEC.2008.4483211.
[S57] C. Symons, Exploring Software Project Effort versus Duration Trade-offs, IEEE Software. 29 (2012) 67–74. doi:10.1109/MS.2011.126.
[S58] T.K. Le-Do, K.-A. Yoon, Y.-S. Seo, D.-H. Bae, Filtering of Inconsistent Software Project Data for Analogy-Based Effort Estimation, in: Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual, 2010: pp. 503 –508. doi:10.1109/COMPSAC.2010.56.
[S59] K. Toda, A. Monden, K. Matsumoto, Fit data selection for software effort estimation models, in: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, New York, NY, USA, 2008: pp. 360–361. doi:10.1145/1414004.1414084.
[S60] M. Azzeh, D. Neagu, P.I. Cowling, Fuzzy grey relational analysis for software effort estimation, Empir Software Eng. 15 (2010) 60–90. doi:10.1007/s10664-009-9113-0.
[S61] M. Tsunoda, S. Amasaki, A. Monden, Handling categorical variables in effort estimation, in: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, New York, NY, USA, 2012: pp. 99–102. doi:10.1145/2372251.2372267.
[S62] M. Tsunoda, S. Amasaki, C. Lokan, How to treat timing information for software effort estimation?, in: Proceedings of the 2013 International Conference on Software and System Process, ACM, New York, NY, USA, 2013: pp. 10–19. doi:10.1145/2486046.2486051.
[S63] L. Buglione, C. Gencel, Impact of base functional component types on software functional size based effort estimation, in: A. Jedlitschka, O. Salo (Eds.), Product-Focused Software Process Improvement, Proceedings, Frascati, ITALY, 2008: pp. 75–89.
[S64] M. Azzeh, D. Neagu, P. Cowling, Improving analogy software effort estimation using fuzzy feature subset selection algorithm, in: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering, ACM, New York, NY, USA, 2008: pp. 71–78. doi:10.1145/1370788.1370805.
[S65] Y.-S. Seo, K.-A. Yoon, D.-H. Bae, Improving the Accuracy of Software Effort Estimation Based on Multiple Least Square Regression Models by Estimation Error-Based Data Partitioning, in: Software Engineering Conference, 2009. APSEC '09. Asia-Pacific, 2009: pp. 3 –10. doi:10.1109/APSEC.2009.57.

[S66] O.O. Top, B. Ozkan, M. Nabi, O. Demirors, Internal and External Software Benchmark Repository Utilization for Effort Estimation, in: Software Measurement, 2011 Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA), 2011: pp. 302 –307. doi:10.1109/IWSM-MENSURA.2011.41.
[S67] C. Lokan, E. Mendes, Investigating the use of chronological split for software effort estimation, Software, IET. 3 (2009) 422 –434. doi:10.1049/iet-sen.2008.0107.
[S68] C. Lokan, E. Mendes, Investigating the Use of Duration-Based Moving Windows to Improve Software Effort Prediction, in: Software Engineering Conference (APSEC), 2012 19th Asia-Pacific, 2012: pp. 818–827. doi:10.1109/APSEC.2012.74.
[S69] V.K. Bardsiri, D.N.A. Jawawi, A.K. Bardsiri, E. Khatibi, LMES: A localized multi-estimator model to estimate software development effort, Engineering Applications of Artificial Intelligence. 26 (2013) 2624–2640. doi:10.1016/j.engappai.2013.08.005.
[S70] M.A. Al-Hajri, A.A. Abdul Ghani, M.N. Sulaiman, M.H. Selamat, Modification of standard Function Point complexity weights system, Journal of Systems and Software. 74 (2005) 195–206. doi:10.1016/j.jss.2003.12.033.
[S71] K. Haaland, I. Stamelos, R. Ghosh, R. Glott, On the Approximation of the Substitution Costs for Free/Libre Open Source Software, in: Informatics, 2009. BCI '09. Fourth Balkan Conference in, 2009: pp. 223 –227. doi:10.1109/BCI.2009.38.
[S72] Y.-S. Seo, D.-H. Bae, On the value of outlier elimination on software effort estimation research, Empir Software Eng. 18 (2013) 659–698. doi:10.1007/s10664-012-9207-y.
[S73] S.-J. Huang, N.-H. Chiu, Optimization of analogy weights by genetic algorithm for software effort estimation, Inf. Softw. Technol. 48 (2006) 1034–1045. doi:10.1016/j.infsof.2005.12.020.
[S74] V.K.Y. Chan, W.E. Wong, Optimizing and simplifying software metric models constructed using maximum likelihood methods, in: Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International, 2005: pp. 1 –6. doi:10.1109/COMPSAC.2005.116.
[S75] V.K.Y. Chan, W.E. Wong, Outlier elimination in construction of software metric models, in: Proceedings of the 2007 ACM Symposium on Applied Computing, ACM, New York, NY, USA, 2007: pp. 1484–1488. doi:10.1145/1244002.1244319.
[S76] Q. Liu, R. Mintram, Preliminary data analysis methods in software estimation, Softw. Qual. J. 13 (2005) 91–115. doi:10.1007/s11219-004-5262-y.
[S77] E. Mendes, C. Lokan, Replicating studies on cross- vs single-company effort models using the ISBSG Database, Empir. Softw. Eng. 13 (2008) 3–37. doi:10.1007/s10664-007-9045-5.
[S78] M. Tsunoda, K. Toda, K. Fushida, Y. Kamei, M. Nagappan, N. Ubayashi, Revisiting software development effort estimation based on early phase development activities, in: Proceedings of the 10th Working Conference on Mining Software Repositories, IEEE Press, Piscataway, NJ, USA, 2013: pp. 429–438. http://dl.acm.org/citation.cfm?id=2487085.2487163 (accessed November 6, 2013).
[S79] I. Gonzalez-Carrasco, R. Colomo-Palacios, J. Luis Lopez-Cuadrado, F.J. Garcia Penalvo, SEffEst: Effort estimation in software projects using fuzzy logic and neural networks, Int. J. Comput. Intell. Syst. 5 (2012) 679–699. doi:10.1080/18756891.2012.718118.
[S80] D. Rodríguez, J.J. Cuadrado, M.A. Sicilia, R. Ruiz, Segmentation of software engineering datasets using the m5 algorithm, in: Proceedings of the 6th International Conference on Computational Science - Volume Part IV, Springer-Verlag, Berlin, Heidelberg, 2006: pp. 789–796. doi:10.1007/11758549_106.
[S81] M. Garre, J.J. Cuadrado, M.A. Sicilia, M. Charro, D. Rodriguez, Segmented parametric software estimation models: using the EM algorithm with the ISBSG 8 database, in: Information Technology

Interfaces, 2005. 27th International Conference on, 2005: pp. 181 – 187. doi:10.1109/ITI.2005.1491119.
[S82] J. Aroba, J.J. Cuadrado-Gallego, M.-A. Sicilia, I. Ramos, E. Garcia-Barriocanal, Segmented software cost estimation models based on fuzzy clustering, J. Syst. Softw. 81 (2008) 1944–1950. doi:10.1016/j.jss.2008.01.016.
[S83] M. Fernández-Diego, M. Martínez-Gómez, J.-M. Torralba-Martínez, Sensitivity of results to different data quality meta-data criteria in the sample selection of projects from the ISBSG dataset, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering, ACM, New York, NY, USA, 2010: pp. 13:1–13:9. doi:10.1145/1868328.1868348.
[S84] E. Papatheocharous, A.S. Andreou, Software cost estimation using artificial neural networks with inputs selection, in: J. Cardoso, J. Cordoso, J. Filipe (Eds.), Insticc-Inst Syst Technologies Information Control & Communication, Funchal, PORTUGAL, 2007: pp. 398–407.
[S85] J.S. Pahariya, V. Ravi, M. Carr, Software cost estimation using computational intelligence techniques, in: Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 2009: pp. 849 –854. doi:10.1109/NABIC.2009.5393534.
[S86] A.S. Andreou, E. Papatheocharous, Software Cost Estimation using Fuzzy Decision Trees, in: Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering, IEEE Computer Society, Washington, DC, USA, 2008: pp. 371–374. doi:10.1109/ASE.2008.51.
[S87] C. Lopez-Martin, C. Isaza, A. Chavoya, Software development effort prediction of industrial projects applying a general regression neural network, Empir Software Eng. 17 (2012) 738–756. doi:10.1007/s10664-011-9192-6.
[S88] L.L. Minku, X. Yao, Software effort estimation as a multiobjective learning problem, ACM Trans. Softw. Eng. Methodol. 22 (2013) 35:1–00*35:32. doi:10.1145/2522920.2522928.
[S89] M. Fernandez-Diego, S. Elmouaden, J. Torralba-Martinez, Software Effort Estimation Using NBC and SWR: A Comparison Based on ISBSG Projects, in: 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2012: pp. 132–136. doi:10.1109/IWSM-MENSURA.2012.28.
[S90] R. Setiono, K. Dejaeger, W. Verbeke, D. Martens, B. Baesens, Software Effort Prediction Using Regression Rule Extraction from Neural Networks, in: Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on, 2010: pp. 45 –52. doi:10.1109/ICTAI.2010.82.
[S91] P. Sentas, L. Angelis, I. Stamelos, G. Bleris, Software productivity and effort prediction with ordinal regression, Information and Software Technology. 47 (2005) 17–29. doi:10.1016/j.infsof.2004.05.001.
[S92] J.J. Cuadrado Gallego, D. Rodriguez, M. Angel Sicilia, M.G. Rubio Angel, A.G. Crespo, Software project effort estimation based on multiple parametric models generated through data clustering, J. Comput. Sci. Technol. 22 (2007) 371–378. doi:10.1007/s11390-007-9043-5.
[S93] Y. Shan, R.I. McKay, C.J. Lokan, D.L. Essam, Software project effort estimation using genetic programming, in: Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, 2002: pp. 1108 – 1112 vol.2. doi:10.1109/ICCCAS.2002.1178979.
[S94] M. Azzeh, P.I. Cowling, D. Neagu, Software Stage-Effort Estimation Based on Association Rule Mining and Fuzzy Set Theory, in: Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, 2010: pp. 249 –256. doi:10.1109/CIT.2010.76.
[S95] S. Amasaki, C. Lokan, The Effects of Moving Windows to Software Estimation: Comparative Study on Linear Regression and Estimation by Analogy, in: 2012 Joint Conference of the 22nd

International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2012: pp. 23–32. doi:10.1109/IWSM-MENSURA.2012.13.
[S96] Z. Jiang, C. Comstock, The Factors Significant to Software Development Productivity, in: C. Ardil (Ed.), Proceedings of World Academy of Science, Engineering and Technology, Vol 19, World Acad Sci, Eng & Tech-Waset, Bangkok, THAILAND, 2007: pp. 160–164.
[S97] L. Song, L.L. Minku, X. Yao, The impact of parameter tuning on software effort estimation using learning machines, in: Proceedings of the 9th International Conference on Predictive Models in Software Engineering, ACM, New York, NY, USA, 2013: pp. 9:1–9:10. doi:10.1145/2499393.2499394.
[S98] P.C. Pendharkar, J.A. Rodger, The relationship between software development team size and software development cost, Commun. ACM. 52 (2009) 141–144. doi:10.1145/1435417.1435449.
[S99] Y. Zorgios, O. Vlismas, G. Venieris, The SECI Model and the Learning Curve Phenomenon, in: C. Stam (Ed.), PROCEEDINGS OF THE EUROPEAN CONFERENCE ON INTELLECTUAL CAPITAL, Academic Conferences Ltd, INHolland Univ Appl Sci, Haarlem, NETHERLANDS, 2009: pp. 589–599.
[S100] L. Lavazza, S. Morasca, G. Robiolo, Towards a simplified definition of Function Points, Information and Software Technology. 55 (2013) 1796–1809. doi:10.1016/j.infsof.2013.04.003.
[S101] A.B. Nassif, D. Ho, L.F. Capretz, Towards an early software estimation using log-linear regression and a multilayer perceptron model, Journal of Systems and Software. 86 (2013) 144–160. doi:10.1016/j.jss.2012.07.050.
[S102] C. Lokan, E. Mendes, Using chronological splitting to compare cross- and single-company effort models: further investigation, in: Proceedings of the Thirty-Second Australasian Conference on Computer Science - Volume 91, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2009: pp. 47–54. http://dl.acm.org/citation.cfm?id=1862659.1862668 (accessed July 16, 2012).
[S103] R. Jeffery, M. Ruhe, I. Wieczorek, Using public domain metrics to estimate software development effort, in: Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International, 2001: pp. 16 –27. doi:10.1109/METRIC.2001.915512.
[S104] L.L. Minku, X. Yao, Using unreliable data for creating more reliable online learners, in: The 2012 International Joint Conference on Neural Networks (IJCNN), 2012: pp. 1–8. doi:10.1109/IJCNN.2012.6252711.
[S105] N. Mittas, L. Angelis, Visual comparison of software cost estimation models by regression error characteristic analysis, Journal of Systems and Software. 83 (2010) 621–637. doi:10.1016/j.jss.2009.10.044.
[S106] S.S. Bajwa, C. Gencel, What Are the Significant Cost Drivers for COSMIC Functional Size Based Effort Estimation?, in: A. Abran, R. Braungarten, R.R. Dumke, J.J. Cuadrado Gallego, J. Brunekreef (Eds.), Software Process and Product Measurement, Proceedings, Springer-Verlag Berlin, Hogesch van Amsterdam, Amsterdam, NETHERLANDS, 2009: pp. 62–75.
[S107] C. Lokan, What should you optimize when building an estimation model?, in: Software Metrics, 2005. 11th IEEE International Symposium, 2005: pp. 1 –10. doi:10.1109/METRICS.2005.55.

Appendix B

List of the 71 independent variables that have been used to construct effort estimation models.

Position	Variables	Attributes group	Number of appearances in journals	Total number of appearances	Proportion %
1	Functional Size	Sizing	31	66	61.7
2	Development Type	Grouping	30	62	57.9
3	Language Type	Project	26	57	53.3
4	Development Platform	Project	28	56	52.3
5	Adjusted Function Points	Sizing	16	30	28.0
6	Max Team Size	Effort attributes	15	30	28.0
7	Organisation Type	Grouping	15	27	25.2
8	Primary Programming Language	Project	16	25	23.4
9	Project Elapsed Time	Schedule	12	23	21.5
10	Application Type	Grouping	17	23	21.5
11	Business Area Type	Grouping	13	18	16.8
12	Enquiry count	Size	11	17	15.9
13	File count	Size	12	17	15.9
14	Interface count	Size	12	17	15.9
15	Output count	Size	11	16	15.0
16	Input count	Size	11	15	14.0
17	1st Data Base System	Project	11	14	13.1
18	Resource Level	Effort attributes	8	13	12.1
19	Used Methodology	Project	7	12	11.2
20	Average Team Size	Effort attributes	6	10	9.3
21	Count Approach	Sizing	3	7	6.5
22	Package Customisation	Grouping	5	7	6.5
23	Development Techniques	Documents & Techniques	5	7	6.5
24	Added count	Size	2	7	6.5
25	Normalised PDR (ufp)	Productivity	3	6	5.6
26	Project Inactive Time	Schedule	3	6	5.6
27	Architecture	Architecture	3	6	5.6
28	CASE Tool Used	Project	5	6	5.6
29	How Methodology Acquired	Project	5	6	5.6
30	Changed count	Size	2	6	5.6
31	Effort Plan	Schedule	1	5	4.7
32	Effort Specify	Schedule	1	5	4.7
33	Lines of Code	Size other than FSM	3	5	4.7
34	Year of Project	Software age	2	4	3.7
35	Implementation	Schedule	3	4	3.7

	Date				
36	Deleted count	Size	1	4	3.7
37	Value Adjustment Factor	Sizing	3	3	2.8
38	Pre 2002 PDR (afp)	Productivity	2	3	2.8
39	Project Activity Scope	Schedule	3	3	2.8
40	Effort Design	Schedule	1	3	2.8
41	Effort Build	Schedule	1	3	2.8
42	Effort Test	Schedule	1	3	2.8
43	Effort Implement	Schedule	1	3	2.8
44	1st Operating System	Project	3	3	2.8
45	User Base - Locations	Product	2	3	2.8
46	COSMIC (Entry, exit, read, write)	Size	0	3	2.8
47	1st Hardware	Project	2	2	1.9
48	User Base - Concurrent Users	Product	2	2	1.9
49	Recording Method	Effort attributes	1	2	1.9
50	Data Quality Rating	Rating	1	1	0.9
51	Normalised Work Effort Level 1	Effort	1	1	0.9
52	Normalised Work Effort	Effort	0	1	0.9
53	Minor Defects	Quality	1	1	0.9
54	Major Defects	Quality	1	1	0.9
55	Extreme Defects	Quality	1	1	0.9
56	Total Defects Delivered	Quality	0	1	0.9
57	Degree of Customisation	Grouping	1	1	0.9
58	Client Server?	Architecture	1	1	0.9
59	Type of Server	Architecture	1	1	0.9
60	Specification Techniques	Documents & Techniques	1	1	0.9
61	Design Techniques	Documents & Techniques	1	1	0.9
62	Functional Sizing Technique	Documents & Techniques	1	1	0.9
63	1st Language	Project	1	1	0.9
64	1st Debugging Tool	Project	1	1	0.9
65	1st Other Platform	Project	1	1	0.9
66	2nd Hardware	Project	1	1	0.9
67	2nd Language	Project	1	1	0.9
68	2nd Operating System	Project	1	1	0.9
69	2nd Data Base System	Project	1	1	0.9
70	User Base - Business Units	Product	1	1	0.9
71	Intended Market	Product	1	1	0.9

Appendix C

Set of selected papers and occurrences of the 20 most used independent variables.

Ref.	FS	DT	LT	DP	AFP	MTS	OT	PPL	PET	AT	BAT	EC	FC	IFC	OC	INC	1DBS	RL	UM	ATS	Total
S1		X	X	X	X	X				X									X		7
S2		X	X	X	X		X	X	X	X	X		X	X			X	X	X		14
S3	X		X	X		X															4
S4					X							X	X	X	X	X					6
S5		X		X	X			X		X	X	X	X	X	X	X	X				12
S6		X			X		X			X		X	X	X	X	X					9
S7					X				X											X	3
S8												X	X	X	X	X					5
S9			X	X				X	X												4
S10					X																1
S11	X	X	X																		3
S12					X							X	X	X	X	X					6
S13	X	X	X																		3
S14	X	X	X	X		X			X									X			7
S15		X		X	X		X	X		X	X	X	X	X	X	X					12
S16	X																				1
S17	X																				1
S18	X	X	X																		3
S19	X	X	X	X		X															5
S20	X	X	X	X			X			X	X										7
S21		X	X	X	X				X												5
S22	X	X		X				X													4
S23	X																		X		2
S24	X																			X	2
S25			X	X	X		X	X													5
S26					X							X	X	X	X	X					6
S27		X		X	X			X		X	X	X	X	X	X	X	X				12
S28	X	X	X	X					X												5
S29		X	X	X	X	X			X									X			7
S30					X	X			X												3
S31		X	X	X	X	X				X											6
S32	X					X															2
S33	X	X	X							X									X		5
S34		X	X	X		X	X			X	X										7
S35	X	X	X	X																	4
S36	X	X	X	X			X	X		X	X										8
S37	X	X	X	X		X	X	X		X	X						X				10
S38		X	X	X		X	X	X		X	X						X		X		10
S39		X	X	X		X	X	X		X	X						X		X		10
S40		X	X	X		X	X	X		X	X						X		X		10
S41	X																				1
S42	X					X			X												3
S43	X	X	X	X																	4
S44	X	X	X	X			X	X		X	X						X			X	10
S45	X		X	X															X		4
S46	X																				1
S47	X	X					X														3
S48		X	X	X	X															X	5
S49		X		X	X			X			X							X			6
S50	X																			X	2
S51	X	X	X	X		X	X	X												X	8
S52	X	X	X																		3
S53	X	X	X	X		X	X	X	X	X	X						X	X			12
S54	X	X	X	X			X	X	X	X	X						X	X	X		12
S55					X							X	X								3
S56					X							X	X	X	X	X		X			7
S57	X	X						X													3
S58	X	X	X			X			X												5

S59	X	X						X										X			4
S60					X							X	X	X	X	X		X			7
S61	X	X		X				X													4
S62	X	X		X			X	X													5
S63	X	X																			2
S64		X			X	X			X			X		X	X			X		X	9
S65		X	X		X	X			X												5
S66	X																				1
S67	X	X	X	X																	4
S68	X	X	X	X			X														5
S69	X	X		X			X					X	X	X	X	X					9
S70	X																				1
S71	X	X	X																		3
S72					X	X			X												3
S73			X	X		X						X	X	X	X	X					8
S74	X		X	X		X															4
S75			X	X	X	X															4
S76	X	X	X	X			X		X								X	X			8
S77	X	X	X	X																	4
S78	X	X	X	X					X												5
S79		X	X	X					X								X	X			6
S80	X	X	X	X					X												5
S81	X																				1
S82	X																				1
S83					X																1
S84	X				X				X			X	X	X	X	X		X			9
S85												X	X	X	X	X					5
S86					X				X											X	3
S87					X																1
S88	X	X	X				X														4
S89	X		X	X															X		4
S90	X	X	X	X		X	X	X		X							X		X		10
S91		X	X	X		X	X	X		X	X						X		X		10
S92	X																				1
S93	X					X															2
S94																					0
S95	X	X	X	X			X														5
S96		X	X	X																X	4
S97	X	X	X	X																	4
S98	X																			X	2
S99	X																				1
S100	X							X		X		X	X	X	X	X					8
S101	X																				1
S102	X	X	X	X		X	X														6
S103	X		X	X		X	X				X										6
S104	X	X	X	X																	4
S105	X	X	X	X					X												5
S106	X	X	X	X		X				X											6
S107		X	X	X	X	X	X	X	X	X	X										10