Productivity of Software Enhancement Projects: an Empirical Study

Luigi Lavazza[1][0000–0002–5226–4337] Geng Liu[2][0000-0003-4686-0834] Roberto Meli[3][0000-0003-1069-8548]

[1] Università degli Studi dell'Insubria, Varese, Italy
luigi.lavazza@uninsubria.it
[2] Hangzhou Dianzi University. Hangzhou, China
liugeng@hdu.edu.cn
[3] DPO Srl. Rome, Italy
roberto.meli@dpo.it

Abstract.
Background. Having a correct, although approximate, knowledge of software development productivity is clearly important. In some environments, the belief that software enhancement projects are characterized by higher productivity than new software development has emerged.
Aim. We want to understand whether the mentioned belief is rooted on solid bases or is due to some cognitive biases.
Method. An empirical study was performed, analyzing the data from a large dataset that collects data from real-life projects. Several statistical methods were used to evaluate the unitary cost (i.e., the cost per Function Point) of enhancement projects and new developments.
Results. Our analyses show that–contrary to some popular beliefs–software enhancement costs more than new software development, at least for projects greater than 300 Function Points.
Conclusions. Project managers and other stakeholders interested in the actual cost of software should reject ill-based evaluations that the productivity of software enhancement is greater than new software development. More generally, objective evaluations based on the analysis of representative data should be preferred to evaluations affected by cognitive biases.

Keywords: Function Point Analysis, Functional Size Measurement, Software measurement, Software Development Productivity, Software Maintenance. Software Enhancement.

1 Introduction

Software cost models are relevant to the market since they influence budget allocation on individual projects, tenders, contracts and finally they affect the quality of customer-supplier relationships. Since the number of factors that may influence, at various levels of impact, the productivity in delivering software systems is huge, it is unavoidable to make some assumptions and to focus on a subset of variables considered as significant. Assumptions may be based on direct personal experiences, on common sense or on empirical evidence derived from collected data. Sometimes the experts that build software cost models are induced to introduce in models some biases that may affect the quality of correlation among the considered variables. This is done in absence of awareness due to the cognitive biases effect.

Specifically, we have observed the tendency to consider the enhancement of existing software as less demanding–in terms of total effort–than the development of new software. Since such beliefs can have quite relevant consequences (e.g., setting unrealistic prices for software enhancement contracts), it is of great importance that we show if the belief is rooted on solid bases, or it is affected by cognitive biases.

In this paper, we apply the empirical method of testing the assumptions using adequate data to address the problem of evaluating the actual effort of software enhancement, especially compared to the effort of developing new software. In this way, we contribute to reduce or inhibit the impact of cognitive biases on software cost models.

The paper is organized as follows. Section 2 presents the motivations of the work and highlights the importance of getting objective evaluations of the productivity of software enhancement. Section 3 describes the empirical study through which we derive objective effort evaluations. In Section 4 we discuss the threats to the validity of the empirical study. Section 5 accounts for related work. Finally, Section 6 draws some conclusions and outlines future work.

2 Motivations and Goals

Cognitive sciences have shown that even the finest expert may be affected by cognitive biases [17]. Intuition, in software engineering, is often blurred by prejudice, confirmation bias, overconfidence, group-thinking, availability bias, framing [1817].

Prejudice is a pre-defined cause-effect relationship, based on hyper-generalizations, that does not need a test, validation, adaptation in order to be considered true by its "owner".

Confirmation bias is the tendency to pay undue attention to sources that confirm our existing beliefs while ignoring sources that challenge our beliefs. Once we have elaborated a theory, we tend to look for confirmations instead of unexplained facts.

Overconfidence bias is the tendency to overestimate one's skills and abilities. This may lead to ignore some factors in the model only because we do not think that we may personally be affected by that factor.

Group-thinking is a psychological phenomenon that occurs within a group of people in which the desire for harmony or conformity in the group results in an irrational or dysfunctional decision-making outcome, by inhibiting critical thinking to avoid conflicts. This may happen even to people that do not know one each other, but that belong to a community ruled by opinion leaders and recognized experts. It is very difficult for anybody to swim against the flow.

Availability bias is a tendency to allow information that is easier to recall unduly influence preconceptions or judgments. We use in models the parameters that are the easiest to be measured, regardless of the relevance they really have.

Finally, the *framing effect* is the tendency to react differently to situations that are fundamentally identical but presented (or framed) differently.

All these cognitive biases may induce to build a model that may be poorly representative of reality. The only way to know if this has happened is to “de-bias” the reasoning with specific approaches possibly supported by empirical data.

The wrong definition of a software cost model may be favored by situation in which the outcome of a production process is influenced by two or more factors that may affect the result in opposite ways and we are wrong in the identification of the resultant of the vector composition.

In this paper, we will focus on the productivity associated to a functional enhancement project compared to a new development project. If we believe that adding, changing and deleting a certain amount of functionalities in an existent application is much easier than adding the same amount of functionalities in a new application (due to reuse, fundamentally), we must expect a higher productivity in enhancement maintenance. On the contrary, if we consider the difficulty of adding, changing and deleting functionalities that are not well documented, were written with old technologies, by not particularly skilled people in an unmanaged environment, then we will expect a lower productivity with respect to new development. Which one is true or more often true in the market? The right thing to do is to collect data and derive statistical driven inference. The most frequent thing that has been done in the past is the application of the cognitive biases illustrated before.

The initial idea that changing and deleting functionalities should be associated to a lower functional measure, if compared with the operation of adding new functions, was supported by a NESMA (Netherlands Software Metrics Users Association) document [19]. This work proposed a way to consider the impact of change in enhancement maintenance projects that is mainly associated to a reduction in size. The source of the document is highly reliable and the approach was also adopted for a while by IFPUG (International Function Point User’s Group), who gave it an “institutional” benediction. This favored the group-thinking bias in the community of practitioners and subsequent confirmation bias that led to prejudice. So, at least in Italy–which is one of the most advanced country in using functional size in contracts–the sizing proposal was transformed into a pricing proposal that assigned to a changed function a relevant discount (50%) and to a deleted function a huge discount (90%). A framing bias was set up presenting the situation as a case of pure profitable reuse. The idea was reasonable and consequently “attractive” for practitioners. This approach, in turn, became a “precedent” for all succeeding contracts (again a framing bias). No empirical evidences have been given to support this approach but it became steady as a rock.

The specific situation may have, and actually had, a huge impact on markets and outsourcing contracts (in the order of millions of euro). Consider that the “discount approach” for enhancement initiatives generates less than half the revenues with respect to the neutral assumption (same cost as development by scratch) and less than 35% of the opposite assumption (cost of enhancement is 150% of development by scratch).

A cost model for contractual goals should be built according to sound articulated approaches and empirically derived knowledge [20].

In this paper we present an empirical study–based on data from real-life projects–to determine *objectively* (although possibly approximatively) the real relation between the cost of enhancing existing software and the cost of developing new software.

3 The Empirical Study

3.1 The dataset

We analyzed data from the ISBSG dataset [8]. However, not all the data included in the dataset were used. First of all, we selected the data concerning projects whose size was measured in IFPUG Function Points. In addition, following the recommendations by ISBSG [7], only records having “Data Quality Rating” equal to B or better were used; that is, only projects with the highest data integrity were considered. Similarly, we selected records having “UFP rating” equal to B or better, i.e., the UFP counting was evaluated as sound by the ISBSG quality reviewers. Projects too big or too small were also removed. Specifically, projects having size smaller than 50 UFP were ignore, because their development effort is so small that effort estimation is not even convenient, since estimation cost would be a significant fraction of the development cost. Projects having size greater than 800 UFP we discarded according to the criteria described below, when analogy-based estimation is introduced.

The descriptive statistics of the analyzed datasets are given in Table 1. Specifically, Table 1 illustrates the characteristics of the New developments and Enhancements datasets. For each dataset, statistics concerning the size expressed in UFP, the effort expressed in PH (person hours) and the number of person hours required per function point are given.

Table 1. Descriptive statistics for New development and Enhancement projects.

	New Development	Enhancement
n	861	2935
Size range	[50, 800]	[50, 800]
Size mean	292	176
Size st. dev.	186	149
Size median	246	119
Effort range	[64, 45778]	[31, 92380]
Effort mean	3852	3032
Effort st. dev.	4655	5029
Effort median	2385	1532

3.2 Description of the study

First, we performed a very simple comparison of the effort per FP required by New development and enhancement projects.

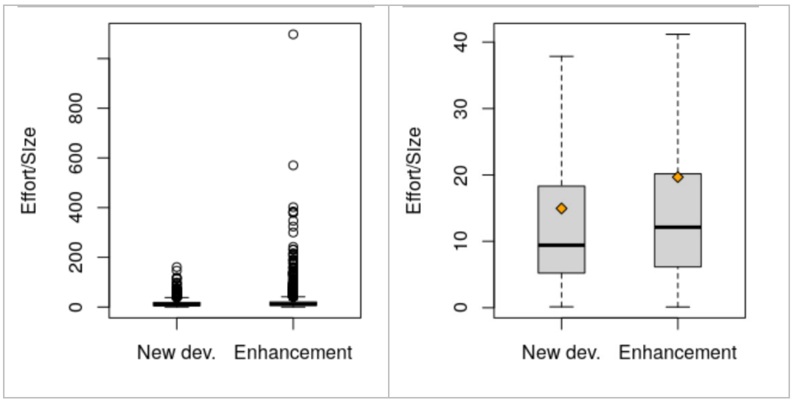

Fig. 1. Boxplots showing the distributions of Effort per FP. In the picture on the right hand side outliers are omitted, and mean values are shown as orange diamonds.

Fig. 1 shows that the quantity of effort per FP required by enhancement projects is generally greater: both the mean and the median are greater than new developments'. Similarly, several enhancement projects required a very large amount of effort per FP, as shown in the left part of Fig. 1. The main statistics concerning the required effort per FP are given in Table 2. It appears that Enhancement projects require more effort per FP than new development projects. This fact was tested by means of the Wilcoxon rank sum test [21], which confirmed that the probability that a randomly selected New development effort per FP is less than a randomly selected Enhancement effort per FP is significantly greater than the probability of picking a greater or equal effort per FP value.

Table 2. Effort/Size statistics for New development and Enhancement projects.

	New Development	Enhancement
Median	9.41	12.12
Range	[0.13, 160.9]	[0.11, 1097.4]
Mean	14.95	19.67
St. dev.	16.75	36.65

Another simple evaluation of the effort per FP required for New developments and enhancement projects can be obtained via lowess curves of effort vs. size. Lowess (locally weighted scatterplot smoothing) is a nonparametric method for fitting a smooth curve between two variables; in this nonparametric method, the linearity assumptions of conventional regression methods are relaxed.

Fig. 2 shows the lowess curves of effort vs. size for the considered project types. It can be observed that for sizes up to just over 200 UFP the effort per FP required is approximately the same for the two types of projects. For larger projects, the effort per FP slightly increases for Enhancement projects, while it decreases substantially for New developments.

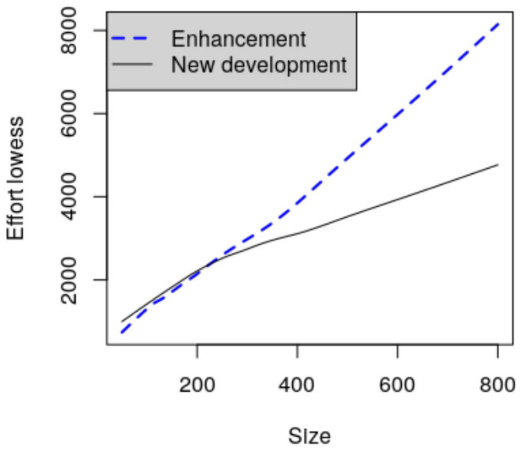

Fig. 2. Lowess curves of Effort vs Size for New development and Enhancement projects.

Several project managers could consider the observations reported above to be sufficient to conclude that–except for small projects–the productivity of New developments is greater than the productivity of Enhancement projects. However, we would be more comfortable if we could provide evidence based on proper model, which provide a statistically sound synthesis of the relationship that links effort and size. To this end, we proceeded to build model of effort as a function of size.

All effort models were derived via ordinary least square (OLS) regression. We used Cook's distance to identify possible influential observations. Data points with Cook's distance greater than 4/n (n being the cardinality of the training set) were considered for removal as suggested by Kitchenham and Mendes [5]. All the models illustrated below were checked for the usual characteristics of OLS regression [3]. All the results reported are statistically significant at the α = 0.05 level, as is common in Empirical Software Engineering and many other disciplines.

Table 3. OLS linear models for New development and Enhancement projects.

	New Development	Enhancement
Model	Effort = 9 Size	Effort = 11.48 Size
Num. outliers	292 (34%)	1172 (40%)
P value	$< 2^{-16}$	$< 2^{-16}$
Adjusted R^2	0.76	0.82
Normal residuals	NO	NO

We started building linear models. The obtained models and their characteristics are given in Table 3.

The OLS linear regression models described in Table 3 provide some interesting indications, in that 1) they confirm that Enhancement projects are more expensive than New development projects; 2) the obtained coefficients are very close to the median Effort/Size values given in Table 2. However, both models do not conform to the OLS regression constraints, having not normally distributed residuals. In addition, they were built by discarding as outliers a large fraction of the projects.

Therefore, we looked for non-linear OLS regression models. Specifically, we performed logarithmic transformations of both the independent and dependent variables. This choice was suggested by both the desire to be compliant with earlier research (for instance, Boehm's COCOMO [1,2]) and to deal with the characteristics of the data distributions (as suggested by Kitchenham and Mendes [15], among others).

Table 4. OLS log-log models for New development and Enhancement projects.

	New Development	Enhancement
Model	Effort = 29.5 Size$^{0.801}$	Effort = 18.7 Size$^{0.912}$
Num. outliers	100 (12%)	454 (15%)
P value	$< 2^{-16}$	$< 2^{-16}$
Adjusted R^2	0.31	0.42
Normal residuals	Yes	NO

The log-log model for Enhancements has not normal residuals; the log-log model for New developments has normal residuals, but a rather low coefficient of determination. However, they tend to conform that Enhancement projects have greater cost per FP than New developments. The difference in terms of required effort can be observed by looking at the models' curves, shown in Fig. 3.

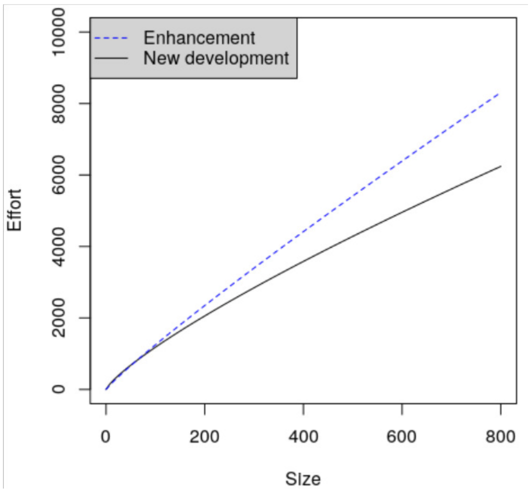

Fig. 3. OLS log-log regression models.

Since no really adequate OLS regression models could be achieved, we adopted analogy-based estimation (AbE). With AbE, the effort required by a project is estimated based on the effort that was required by "similar" projects.

Given a project P, we selected the projects that contribute to estimate the development or maintenance effort for P as follows:

1) Let sp=0.02.
2) Let N_P be the set of projects such that $p \in N_P$ iff (1-sp) size(P) ≤ size(p) ≤ (1+sp) size(P)
3) If $|N_P| \geq 7$, let the estimated effort for P be the median of the efforts of the projects belonging to N_P.
4) Otherwise, increase sp by 0.01 and go back to step 2).

Before proceeding to apply AbE, we determined the size range in which there are enough data points to support AbE. The histograms in Fig. 4 show that–as could be expected–there are few large Enhancement projects. Specifically, above 800 UFP projects are few and sparse, hence it is difficult to find "similar" projects for AbE. Therefore, in the rest of this study we consider only projects having size not greater than 800 UFP. Noticeably, for New development there are no problems of the type discussed above: there are many projects having size greater than 800 UFP, actually, there are many projects up to 2000 UFP. Nonetheless, since we are mainly interested in comparison between Enhancement and New development projects, we need to study both types of projects in the same size range.

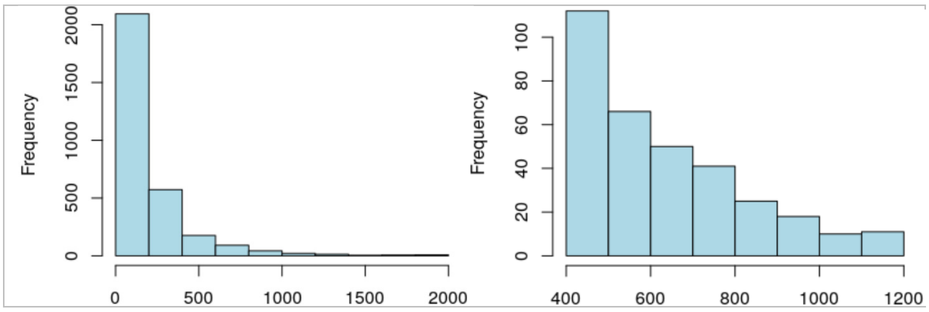

Fig. 4. Size distribution of Enhancement projects. The picture on the right hand size provides a view restricted to the [400,1200] UFP range.

To evaluate whether AbE estimates are worth considering, we have to verify that they provide better performance than baseline models. To this end, Shepperd and MacDonell [9] proposed that an estimation model be taken into consideration only if it provides better estimates than a baseline model; they also proposed to use random estimation as a baseline model.

Shepperd and MacDonell also proposed that the accuracy of a given estimation method be measured via the Mean Absolute Residual (MAR), i.e., the mean of the absolute values of errors, where errors are computed as actuals minus estimates.

A random effort estimation for a project is obtained by picking at random the actual effort of any of the other projects. Of course, in this way there are n−1 possible estimates for every project; therefore, to compute the MAR of the random model we need to average all these possible values. Shepperd and MacDonell suggest to make a large number of random estimates (typically 1000), and then compute the mean MAR. Shepperd and MacDonell observed also that the value of the 5% quantile of the random estimate MARs can be interpreted like α for conventional statistical inference. Accordingly, the MAR of a proposed model should be compared with the 5% quantile of the random estimate MARs, to make us reasonably sure that the model is actually more accurate than the random estimation.

We also used “constant” models as baseline, proposed, among others, by Lavazza and Morasca [10] and Di Martino et al. [11]. With the mean–respectively, median–constant model, the estimated effort for a project is given by the mean–respectively, median–of all other projects' actual efforts.

The models’ estimation errors are given in Table 5 and visualized via a boxplot in Fig. 5, where outliers are not show, for readability. In Fig. 5, the orange diamonds represent the means, i.e., the MARs. The dashed line is the MAR of the mean constant model, the dotted line is the MAR of the median constant model, the continuous line is the 5% quantile of the random estimate MARs. It can be observed that the MARs of AbE estimates are smaller than the baselines' MARs, hence AbE is an improvement over baseline estimation methods. This fact is also confirmed by Wilcoxon sign rank test.

Table 5. MAR of AbE models for New development and Enhancement projects.

Model	New Development	Enhancement
AbE	2380	1940
Const. mean	2997	2664
Const. median	2701	2267
5% MARrnd	3830]	3427

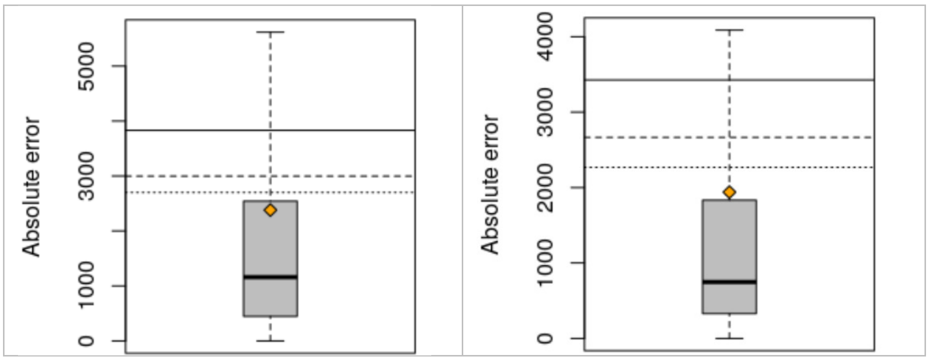

Fig. 5. Distributions of analogy-based effort estimation error for New developments (left) and Enhancement (right) projects (outliers not shown).

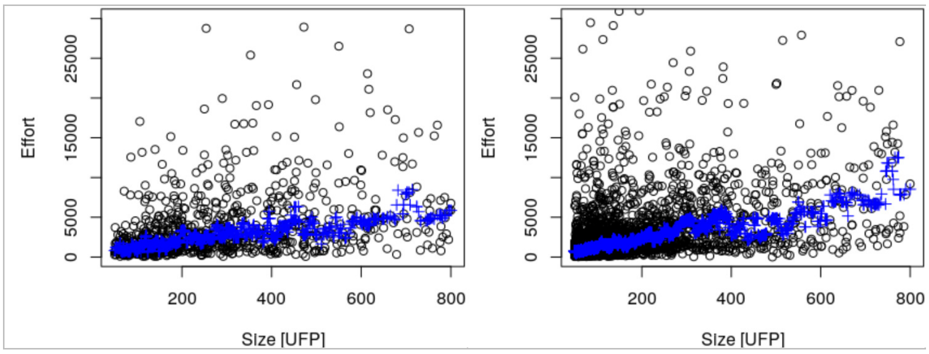

Fig. 6. AbE effort estimates (blue crosses) and actual effort (black circles) with respect to Size, for New developments (left) and Enhancements (right).

Fig. 6 show Effort estimates, in comparison with actual effort values. Understanding the trend of estimated effort vs size (hence of effort per FP) from Fig. 6 is not easy. Thus, we use again the lowess curves to provide a more readable representation of the estimated effort vs. size. Fig. 7 shows such lowess curves for New developments and Enhancement, respectively. It can be observed that, just like in Fig. 2, the slope of the New development curve decreases around 300 FP, while the Enhancement curve appears approximately straight.

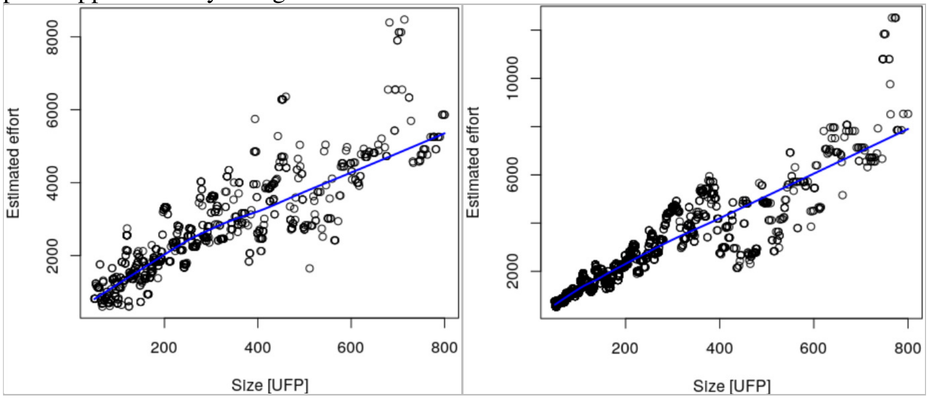

Fig. 7. AbE effort estimates and lowess curve for New developments (left) and Enhancements (right).

Fig. 8 compares the lowess curves given in Fig. 7, highlighting that the effort per FP required by Enhancement projects is greater than the effort per FP required by New developments. Fig. 8 is remarkably similar to Fig. 2 and Fig. 3: this fact reinforces the observation given above.

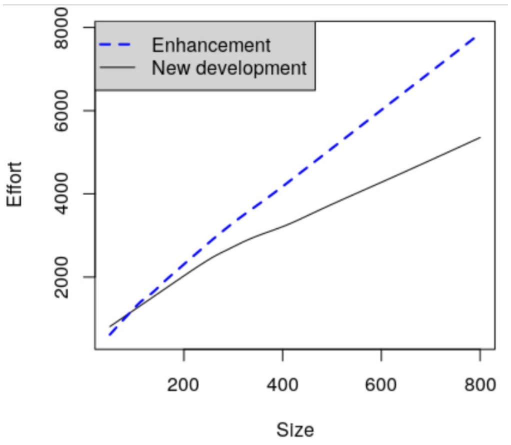

Fig. 8. AbE effort estimate lowess curves for New developments and Enhancements.

3.3 Discussion of Results

In the previous sections, we evaluated the effort per UFP in several ways: using simple statistics, building regression models, and via analogy-based estimation. In all cases we got clear indications that–at least when the project size is larger than 300 UFP, the effort required by Enhancement projects is larger than the effort required by New developments.

However, large variations of the effort per UFP were observed in the ISBSG dataset: Table 2 shows that the standard deviation is larger than the mean for both New developments and Enhancements; for the latter it is almost twice the mean. Therefore, practitioner should be careful in using the productivity values presented in this paper: they should take into account some variability.

4 Threats to Validity

Internal validity of the study could be affected by the way measures were obtained. We mitigated this threat by carefully selecting the measures that are classified as most reliable in the ISBSG dataset.

The models presented in Section 3 do not qualify as proper regression model, since most of them have not normally distributed residuals (and sometimes other problems as well). Nonetheless, together with the other findings given in Section 3, they provide reasonably reliable indications, at least qualitatively.

Concerning the generalizability of the proposed results, we analyzed the largest public available dataset, (see the descriptive statistics in Table 1). Although we cannot claim that our findings are generally valid, the fact that they are based on a large dataset, which collects data from many different software development organizations, supports the hypothesis that our findings are representative of many software projects.

It can be noticed that we built effort models based only on size, while usually effort models account for multiple factors that are believed to affect development or maintenance effort. In this paper, we limited the investigation to size based models to get straightforward indications concerning the amount of effort needed in relation to the size of the project. In this respect, it is worth noting that currently many public administrations and private organizations, worldwide, adopt contractual cost models that are based on the size of the software to be delivered as the only independent variable. Although such practice has evident limits, it is widely used. This paper provides results that help applying the mentioned practice based on objective empirical knowledge, thus avoiding macroscopic mistakes, like assuming that enhancement cost less–on a unitary basis–than new development.

Another possible threat comes from the possibility that the observed differences are due to the managerial choices concerning new development and enhancement projects. For instance, managers could assign more skilled people to new development projects and less skilled people to enhancement projects: this would explain the observed differences, at least partly.

Finally, we have to notice that the available dataset contains little data concerning enhancement projects larger than 800 FP. Accordingly, we limited the study to projects not greater than 800 FP. We cannot make any claim concerning larger projects.

5 Related Work

Approaches to effort prediction are generally grouped into three general categories: expert judgement, algorithmic models, and analogy [12].

Estimation by analogy predicts the effort of the target project using information from former similar project at the system level and sub-system level. The primary steps in this method consist of: 1) choosing the right analogy, usually measured via Euclidean distance, 2) investigating similarities and differences, 3) examining analogy quality, 4) providing the estimation [13].

The key factor of successful EbA method is finding the appropriate right analogy. The simple method is using a fixed number of analogies starting from k=1 and increase this number until non further improvement on the accuracy can be obtained [14,15]. However, in our case, finding additional analogies to be used in conjunction with size analogy was not appropriate: having to devise models that support the notion of effort per FP, we needed to restrict the independent variable to size, specifically, size measured in Function Points.

In fact, the great majority of software effort estimation methods proposed in the literature use software size or size dependent elements as a major explanatory variable. Size measures are considered as the most influential predictors for estimation. Several functional size measures have been defined, including Function Points, Use Case Points, NESMA, FiSMA, Mark II FP, COSMIC FP, SiFP and many others. In this paper we investigate the relationship of effort required between enhancement projects and new development projects, using only FP as a size measure. Investigations using other measures is an objective for future work.

Concerning the study of productivity of software enhancement projects, this is definitely not a new research field. Back in 1993, Abran and Robillard performed an empirical study based on the analysis of 21 projects [16]: among other results, they found that the mean effort per function point was 18.96 PH/FP, quite close to 19.67 PH/FP, the mean value we found in the ISBSG dataset (see Table 2).

6 Conclusions

Having a–possibly approximate–quantitative knowledge of the unitary cost of software projects in terms of effort per function point is very important for several purposes. Specifically, when considering the possibility of making a bid, having a reasonably good idea of the cost of the project is essential.

Relatively little research was performed concerning the unitary cost of enhancement projects, at least in comparison with the huge amount of work performed on investigating the cost of new software development. The relatively small amount of empirical data is favoring ill-based guessing. In some environment, the false believe that software enhancement activity cost less than new development–on a unitary basis–is spreading.

In this paper, we looked at what reliable and objective knowledge we can derive from real project data.

Not all the results we presented here are perfectly reliable from a statistical point of view. Nonetheless, the consistency of results we obtained via different analysis techniques seems to indicate that the indications we derived are–at least qualitatively–correct and reliable.

In summary, we found that:

- Enhancement projects have a unitary cost that is generally greater than new development projects.
- Specifically, the unitary cost of enhancements and new developments is similar for projects up to around 300 FP, while for larger projects the unitary cost of enhancements is greater.
- As shown in Fig. 6, the unitary cost is largely variable, even for projects having approximately the same size. Therefore, the data presented here must be regarded as indicating tendencies, but are not necessarily valid for all projects.

From the cognitive bias point of view, our empirical study showed that, based on the data available in a very reputed dataset (namely, the ISBSG dataset), the assumption that productivity is higher for functional enhancement projects than for new development projects is not supported by evidence. Instead, the opposite is true, except for fairly small projects. We may thus state that–most likely–many huge contracts have been undervalued for years because of an apparently reasonable assumption, not confirmed by empirical data.

Future work include, among other activities, looking for factors that let us select project classes characterized by small variations in unitary cost, and experimenting with different techniques for building effort models.

Acknowledgments

This work has been partially supported by the “Fondo di ricerca d’Ateneo” of the Università degli Studi dell’Insubria.

References

1. Boehm, B.: Software Engineering Economics. Prentice Hall (1981).
2. Boehm, B., Madachy, R., Steece, B., et al.: Software cost estimation with Cocomo II. Prentice Hall PTR (2000).
3. Maxwell, K. Applied statistics for software managers. Prentice Hall PTR Englewood Cliffs (2002).
4. Albrecht, A.J.: Measuring application development productivity. In: Proceedings of the joint SHARE/GUIDE/IBM application development symposium, pp. 83–92. IBM (1979).
5. Mendes, E., and Kitchenham, B. Further comparison of cross-company and within-company effort estimation models for web applications. In Proceedings of the 10th International Symposium on Software Metrics. IEEE, 348–357 (2004)
6. LNCS Homepage, http://www.springer.com/lncs, last accessed 2016/11/21.
7. ISBSG. Productivity Data Query (PDQ) Tool User Guide. Technical Report. ISBSG (2017).
8. ISBSG, International Software Benchmarking Standards Group. ISBSG repository–release R13 (2015).
9. Shepperd, M. and MacDonell, S. Evaluating prediction systems in software project estimation. Information and Software Technology 54(8) 820–827 (2012).
10. Lavazza, L. and Morasca, S. On the Evaluation of Effort Estimation Models. In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. ACM, 41–50 (2017)
11. Di Martino, S., Ferrucci, F., Gravino, C. and Sarro, F. Assessing the Effectiveness of Approximate Functional Sizing Approaches for Effort Estimation. Information and Software Technology (2020)
12. Shepperd, M., Schofield, C., & Kitchenham, B. Effort estimation using analogy. In Proceedings of the 18th International Conference on Software Engineering. IEEE, 170–178 (1996)
13. Azzeh, M., and Nassif, A. B. Analogy-based effort estimation: a new method to discover set of analogies from dataset characteristics. IET Software (2), 39–50 (2015).
14. Mendes, E., Mosley, N., and Counsell, S. A replicated assessment of the use of adaptation rules to improve Web cost estimation. In Proceedings of the International Symposium on Empirical Software Engineering, IEEE, 100–109 (2003)
15. Azzeh, M. and Marwan, A. Value of ranked voting methods for estimation by analogy, IET Software, 7(4), 195–202 (2013)
16. Abran, A., and Robillard, P. N. Reliability of function points productivity model for enhancement projects (a field study). In Proceedings of the International Conference on Software Maintenance, IEEE 80-87 (1993)
17. Gerd Gigerenzer. Reckoning with Risk: Learning to Live with Uncertainty. Penguin books (2002)
18. Mohanani, R., Salman, I., Turhan, B., Rodríguez, P., and Ralph, P. Cognitive biases in software engineering: a systematic mapping study. Transactions on Software Engineering. IEEE (2018)
19. NESMA, Function Point Analysis for Software Enhancement, Guidelines Version 1.0, NESMA (2001)

20. Meli, R. and Iorio, T. Software measurement issues in contractual environments. In Proceedings of the Software Measurement European Forum. (2005)
21. Wilcoxon, F. Individual comparisons by ranking methods. Breakthroughs in statistics. Springer, New York, NY, 196-202 (1992).