An Empirical Analysis of Software Productivity Over Time

Rahul Premraj
Bournemouth University, UK
rpremraj@bmth.ac.uk

Martin Shepperd
Brunel University, UK
martin.shepperd@brunel.ac.uk

Barbara Kitchenham*
National ICT, Australia
Barbara.Kitchenham@nicta.com.au

Pekka Forselius
STTF Oy, Finland
pekka.forselius@kolumbus.fi

Abstract

OBJECTIVE - the aim is to investigate how software project productivity has changed over time. Within this overall goal we also compare productivity between different business sectors and seek to identify major drivers.
METHOD - we analysed a data set of more than 600 projects that have been collected from a number of Finnish companies since 1978.
RESULTS - overall, we observed a quite pronounced improvement in productivity over the entire time period, though, this improvement is less marked since the 1990s. However, the trend is not smooth. We also observed productivity variability between company and business sector.
CONCLUSIONS - whilst this data set is not a random sample so generalisation is somewhat problematic, we hope that it contributes to an overall body of knowledge about software productivity and thereby facilitates the construction of a bigger picture.
Keywords*: project management, projects, software productivity, trend analysis, empirical analysis.*

1. Introduction

Given the importance and size of the software industry it is no surprise that there is a great deal of interest in productivity trends and in particular whether the industry, as a whole, is improving over time. Obviously this is a complex question for at least three reasons.

First, productivity is difficult to measure because the traditional definition, i.e. the ratio of outputs to inputs requires that we have objective methods of measuring both commodities. Unfortunately, for software the notion of output is not straightforward. Lines of code are problematic due to issues of layout, differing language and the fact that most software engineering activity does not directly involve code. An alternative is Function Points (FPs), in its various flavours, which although subject to some criticism [1] are in quite widespread use and so in a sense represent the least bad alternative. In our analysis the output (or size) measure collected is Experience Points 2.0 [8], a variant of FPs.

Second, productivity is impacted by a very large number of factors, many of which are inherently difficult to assess, e.g. task difficulty, skill of the project team, ease of interaction with the customer/client and the level of non-functional requirements imposed such as dependability and performance.

Third, there are clear interactions between many of these factors so for instance, it is easier to be productive if quality can be disregarded.

Despite these caveats, this paper seeks to analyse software project productivity trends from 1978-2003 from a data set of more than 600 projects from Finland. The projects are varied in size (6 - 5000+ FPs), business sector (e.g. Retail) and type (New Development or Maintenance). However, we believe there are sufficient data to draw some preliminary conclusions.

The remainder of the paper is organised as follows. The next section very briefly reviews some related work including a similar, earlier study by Maxwell and Forselius [16]. Next we describe the data set used for our analysis. We then give the results of our analysis, first overall and then after splitting the data set into groups of more closely related projects. We conclude with a discussion of the significance of the results and some comments on the actual process of analysing the data.

* Barbara Kitchenham is also with Keele University, UK Barbara@cs.keele.ac.uk

2. Related Work

The topic of software productivity has generated considerable interest over the years. This comes from four different vantage points.

First to consider different ways of measuring productivity. This work has generally focused upon problems of defining meaningful measures of software output, see for example Behrens on Function Points [3] or Dale and van der Zee [7] who make more general comments upon the business context for productivity measurements.

Second to facilitate effort prediction, see for example Jeffery and Low [9] and Jørgensen et al. [10] who use productivity to build simple effort prediction tools. Another, more complex approach is given by Sentas et al. [21] who use productivity and ordinal regression to assess the reliability of a given project cost classification.

Third, as a form of benchmarking, for example making international comparisons such as Cusumano et al. [6] who analysed productivity and software development practices across four regions (India, Japan, USA and Europe).

Fourth, to explore empirically what factors influence productivity, with the long term objective of guiding the software industry to more effective practices. This is our vantage point. Our main motivation is a previous study conducted by Maxwell and Forselius [16] on an older version of the same data set that we use. In their study, Maxwell and Forselius found Company and Business Sector as the most important variables in explaining productivity. Then they explored the data set in more depth by splitting it across different business sectors and analysing them separately. Then, for each business sector, the most significant variables influencing productivity were identified and later used to construct productivity benchmark equations. Their main conclusions were that there are substantial differences in productivity between companies and to a lesser extent between business sectors.

Finally we wish to point out that a subset of the authors of this paper published a preliminary empirical analysis (using the same data set) on software productivity in the Late Breaking Paper Proceedings of Metrics 2004 [20]. [RP] regrets that due to a clerical error, some of the results were flawed. The current analysis not only rectifies some previous errors but also considerably revises the experimental setup to uncover deeper trends potentially masked by various confounding factors and problems of unbalanced data sets [12].

3. Background on the "Finnish" Data Set

The data set[1] used for analysis in this paper is derived from the 2004 release of the Experience data set usually referred to as the "Finnish" data set. This section briefly introduces the Experience Pro initiative and the data editing performed to make the data set suitable for our analysis.

The Experience data set is a result of a commercial initiative by Software Technology Transfer Finland (STTF) to provide support for software development organisations for both project cost estimation and productivity analyses. This has resulted in a data set which includes software projects between 1978 and 2003. In its current form, the data set comprises 622 projects. Organisations pay an annual fee to gain access to the data via a tool called Experience Pro. The same tool can be used to submit their own project data upon which they are entitled to a discount on their annual fee. The use of the tool for project data submission facilitates standardisation of variables included. In addition, the project data are carefully assessed at STTF by experts before being added to the data base. More information about Experience Pro is available at their website [8].

The projects are derived from a wide range of business sectors spanning financial to telecommunication projects and embrace a range of different platforms and development technologies. The data set includes both New Development (approximately 93% of observations) and Maintenance projects. Project data include size information in function points (FPs), effort and a range of factors to characterise the type of project, factors to characterise the development circumstances, development and target technology. In total, 102 variables are collected, though some are difficult to analyse due to a significant proportion of missing values. A fuller description of the data set may be found in Maxwell and Forselius [16].

Given the long duration that data has been collected, there has been an evolution path for how Size has been collected moving from original IFPUG Function Point Counting to Experience 2.0 Function Point Analysis. Until 1990 the original Albrecht FPs [3] were used, followed by LATURI FPA 1.0 from 1991-5. The main changes were a 5 rather than 3 point complexity scale and changes in measuring databases so that a database was not interpreted as single logical file but instead one per entity plus algorithmic functions were added to the method. Finally Experience Points 2.0 were deployed in 1995 (the impact of total

1 The authors regret that presently the data set is not publicly available. Bona fide researchers are welcome to approach STTF to discuss research opportunities based upon the data set and to enable independent scrutiny of our data analysis. However, in order to protect the commercial needs of STTF – who have invested considerable effort in collecting the data – researchers will not be allowed to publish the data set in its entirety nor to give the data set to other parties without the consent of STTF.

Variable	Mean	Median	Min	Max
Project Size (FPs)	543	329	6	5060
Effort (person hrs)	3967	1789	55	63694
Productivity	0.21	0.16	0.034	0.92

Table 1. Basic Summary Data for the "Finnish602" Data Set

number of entities was dropped from the complexity of logical files (i.e. entities). Both Finnish FSM variants are conformant with the ISO/IEC 14143-1 FSM standard.

All the old projects were re-counted after 1995, so that the whole data set is now based on similar counting practices. Since then the measurement method has remained stable.

Although the initial data set comprised 622 projects, we removed a number of observations in order to avoid suspect values and extreme outliers. To achieve this we applied the following rules to remove projects that had:

- not yet completed (3)
- non standard size measurement (5)
- implausible[2] delivery rates (i.e. < 1 hrFP^{-1} (6) and $>$ 30 hrFP^{-1} (6)).

The numbers in parentheses indicate the count of projects removed for each rule. Thus, in total 20 projects (3.2%) were removed from the analysis. In addition, we relabelled a number of values that were recorded as -1 with 3, which upon discussion with STTF implied that these values were initially incorrectly recorded in the data set. For clarity we refer to the original data set as Finnish622 and the edited data set as Finnish602.

Table 1 presents some basic summary statistics for size, measured in a variant of function points known as Experience Points. Effort is recorded in person hours and raw productivity is defined as the ratio of size to effort, i.e. FPs per hour. Note that for all three variables the distributions are highly skewed with extreme outliers in terms of size, effort and to a lesser extent productivity (in part due to our data editing procedure).

Table 2 gives some overall impression of the diversity of projects included in the data set with banking and insurance being dominant. Note that the category 'other' in fact combines a range of more infrequent categories such as telecommunications, publishing, services, construction, etc. We now proceed to examine productivity and productivity trends in more detail.

2 Implausible values were determined in discussion with staff from STTF. We used a delivery rate value rather than productivity since this was easier to visualise. Our view was essentially that for projects with very short or excessively long times delivery rates, other unreported or misreported factors must come into play. Either way we did not wish to jeopardise our overall analysis.

4. Results

The results are organised such that we analyse the data set as a whole before breaking it down by project type (New Development or Maintenance), by business sector and process model.

4.1. Productivity and Scale Economies

A topic that has generated considerable discussion is whether software development exhibits economies, diseconomies, both or fixed returns to scale [2, 13]. A presumption made by many researchers and embodied in many models such as COCOMO [4] is that of diseconomies, in other words if we assume some production function of the form $E = a\ (S)^b$ where E is project effort, S size, then b is greater than one. We derived this production function by building a linear regression model using the natural log transformation of the data (i.e. $ln(E) = b(ln(S))$) and then, re-transforming the data back into its original scale.

For all 602 projects, we derived $E = 7.345\ (S)^{0.9614}$, which suggests that the relationship indicates very slight economies of scale. However, the 95% confidence limits on b indicate that it is not significantly different from one ($0.909 \leq b \leq 1.014$).

A visual inspection of Figure 1 indicates that the relationship is very close to linear. The goodness of fit statistic or adjusted $R^2 = 0.683$ implies that 68.3% the variability in Effort can be explained by Size. In other words, a simple predictor of project effort based on size in FPs will not be particularly accurate. This is also confirmed by the distribution of residuals (not included) where we saw a clear relationship with Size such that larger errors are associated with larger projects thus the distribution was heteroscedastic.

We then attempted to rebuild the regression model by removing outliers or high influence points, that were identified as having a Cook's distance $D > 4/n$ (where n is number of data points) on the log-log regression model. Removing 31 outliers had very little impact on the slope and exponent of the revised regression model derived as $E = 6.129\ (S)^{0.9932}$ with the 95% confidence limits for b again indicating by a small margin that it is not significantly different from one ($0.94 \leq b \leq 1.047$). For this reason, the remainder of the analysis includes all projects contained within Finnish602.

There has been some debate between researchers about the nature of software production functions with some arguing for economies of scale, e.g. Walston [22] suggested

	Mainframe	Midrange	Multi-platform	Not Defined	Other	PC-Network	Standalone PC	total
Banking	65	39	7	2	0	12	7	132
Insurance	154	24	27	3	2	12	2	224
Manufacturing	20	10	2	1	0	35	5	73
Other	6	6	8	12	0	13	1	46
Public Admin	17	20	26	1	3	16	9	92
Retail	11	14	0	1	0	5	4	35
Total	272	113	70	20	5	93	28	602

Table 2. Business Sector and Hardware Frequencies for "Finnish602" Data Set

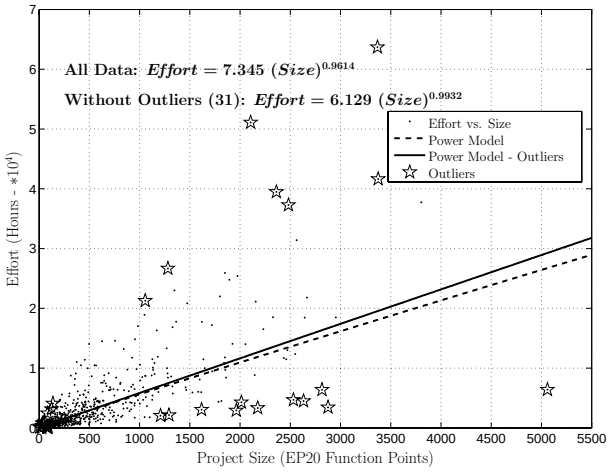

Figure 1. Regression Analysis of Project Size and Effort for all projects

a value of $e = 0.91$ whereas others have argued for diseconomies of scale, e.g. Boehm suggested a value of $e = 1.20$ for embedded mode projects [4]. Yet other researchers have indicated that a linear form best fits the data, or at least that the model cannot be shown to be significantly non-linear, though of course wide confidence limits do not exclude a substantial range of non-linear models. Our results seem to best fit into this category. For a more detailed discussion see Kitchenham [13].

4.2. Productivity Trends Over Time

A question of considerable interest to software engineers and the software industry as a whole is – 'How has productivity changed over time?' Clearly this is a complicated question because many factors influence productivity and we don't wish them to confound our analysis.

As was discussed in section 2, a similar productivity analysis was carried out by Maxwell and Forselius [16] on projects completed prior to 1997. In their study they examined data from a total of 206 projects from 26 companies. We refer to this as the FinnishMF data set. STTF have continued the collection of project data, hence Finnish622 includes the projects from the earlier study.

	FinnishMF	Finnish602 - MF
Start dates	1978-94	1997-2003
No. of companies	26	17
No. of projects[a]	206	401
Project sizes (FPs)	33-3375	27-5060
Productivity $FPhr^{-1}$	0.177	0.233

a Note that the total number of projects does not sum to exactly 602 because although the data set Finnish622 completely subsumes FinnishMF this is not the case for Finnish602 since our exclusion rules in Section 3 cover a small number of FinnishMF projects.

Table 3. Naïve Productivity Comparison of 1978-94 and 1997-2003

In order to make a somewhat simplistic comparison, in Table 3 we compare the reported[3] mean productivity by [16] with the 1997-onwards projects from Finnish602 (i.e. Finnish97+). Whilst the headline result is an improvement in raw productivity of approximately 33% we have to be cautious since there are differences between the two samples. The distribution of projects between business sectors over time is not constant (see Figure 3), nor is project size with a tendency towards smaller projects over time (see Figure 2), and in addition the 1997-onwards data contain a mix of New Development and Maintenance projects whereas the FinnishMF projects are New Development only.

As we discussed in Section 4.1, there is weak – though not conclusive – evidence for economies of scale. In which case, the next question is whether any trend in project size could be a confounding factor for our observation of an improvement in software productivity. However, when we examine project size, in FPs, by year (see Figure 2) we see if anything there is a tendency for projects to decrease in size

3 In fact Maxwell and Forselius [16] give productivity figures by business sector however, we aggregate these into a single value weighted by the number of observations per sector.

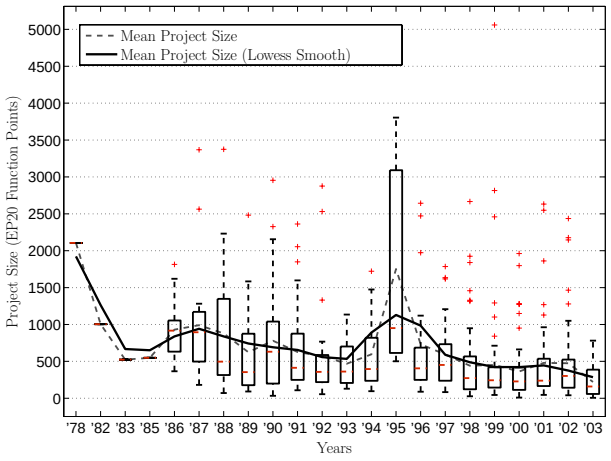

Figure 2. Software Project Size Trends (1987-2003)

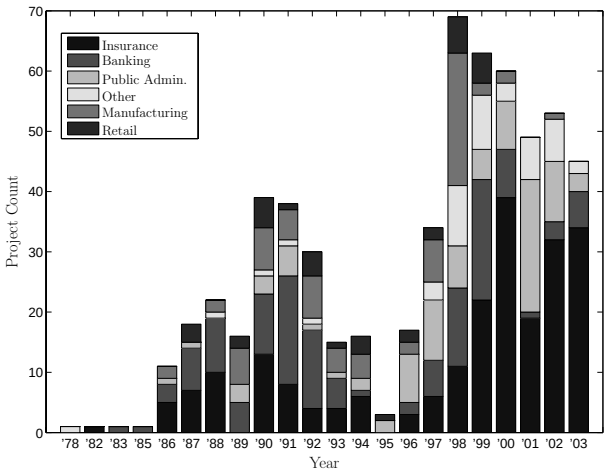

Figure 3. Distribution of Projects by Business Sector by Year)

as indicated by the smoother. The boxplot for each year indicates the spread of project sizes with the horizontal bar within the box showing the median value and '+'s denoting outliers. Note that the peak for 1995 is probably the artefact of having only three observations for that year.

Likewise another source of difficulty is the distribution of projects from different Business Sectors over time. Figure 3 shows the proportions of projects as a bar chart. Probably the clearest pattern is the increasing proportion of Insurance projects in recent years. Later (see Section 4.7) we show that this is one of the least productive Business Sector so again it may be we are understating the productivity gains over time.

Given these problems, we decided to examine the trends in productivity over time from a different perspective as recommended by Kitchenham and Mendes [15]. We constructed a regression equation from data where each year 1978, ..., 2003 became a dummy variable, S_{yr} with the project size in FPs for projects that commenced that year, and zero otherwise. In addition we added boolean dummy variables for each business sector and also for project type (New or Maintenance). This is because previous work [16] and our own analysis (see Section 4.7) has indicated that these are potentially an important source of productivity variation and the proportions of projects are not constant over time, i.e. we have an unbalanced data set. The dependent variable was effort. We then forced all the dummy variables into the regression. From the regression model we derived the value for each β_{yr} together with the 95% confidence limits. We then plotted these values against time with the following interpretation (see Figure 4). The lower the value of β_{yr}, the more productive a software project since this implies less effort is required to implement a given project because the slope of the regression equation is less steep.

Where the shaded area in the plot has greatest breadth, it indicates we have least confidence in the true value of productivity for a given year. This variability in productivity can arise for two reasons. Either there is considerable underlying variation in the productivity for the projects for the particular year, or we have very few data points. Note that the total number of projects for any given year is given underneath the plot. Note also that for 1979-81 and 1984 we have no data and that in general we have little data, and therefore low confidence in the productivity data before the late 1980s.

A visual inspection of the plot (Figure 4) indicates a trend towards an improvement in productivity over time. Since there are productivity fluctuations between the years we have also applied a Lowess smoother (shown as a dark continuous line in Figure 4) to give a clearer idea of the overall trend. This suggests an overall pattern of improvement until the early 1990s, then a deterioration until about 1997 and then another perod of more gentle improvement subsequent levelling out.

This is confirmed by a Spearman correlation analysis of β_{yr} with time for all years ($r_S = -0.88, p < 0.0001$). A possible explanation is the increasing emphasis upon a disciplined and repeatable software process within software development organisations. However, the Spearman correlation for the period of 1992 onwards is still significant but more modest ($r_S = -0.62, p < 0.035$) suggesting some continued improvement but not at the rate of the 1980s. In passing, we note that our initial model that did not contain dummy variables for business sector failed to detect the post

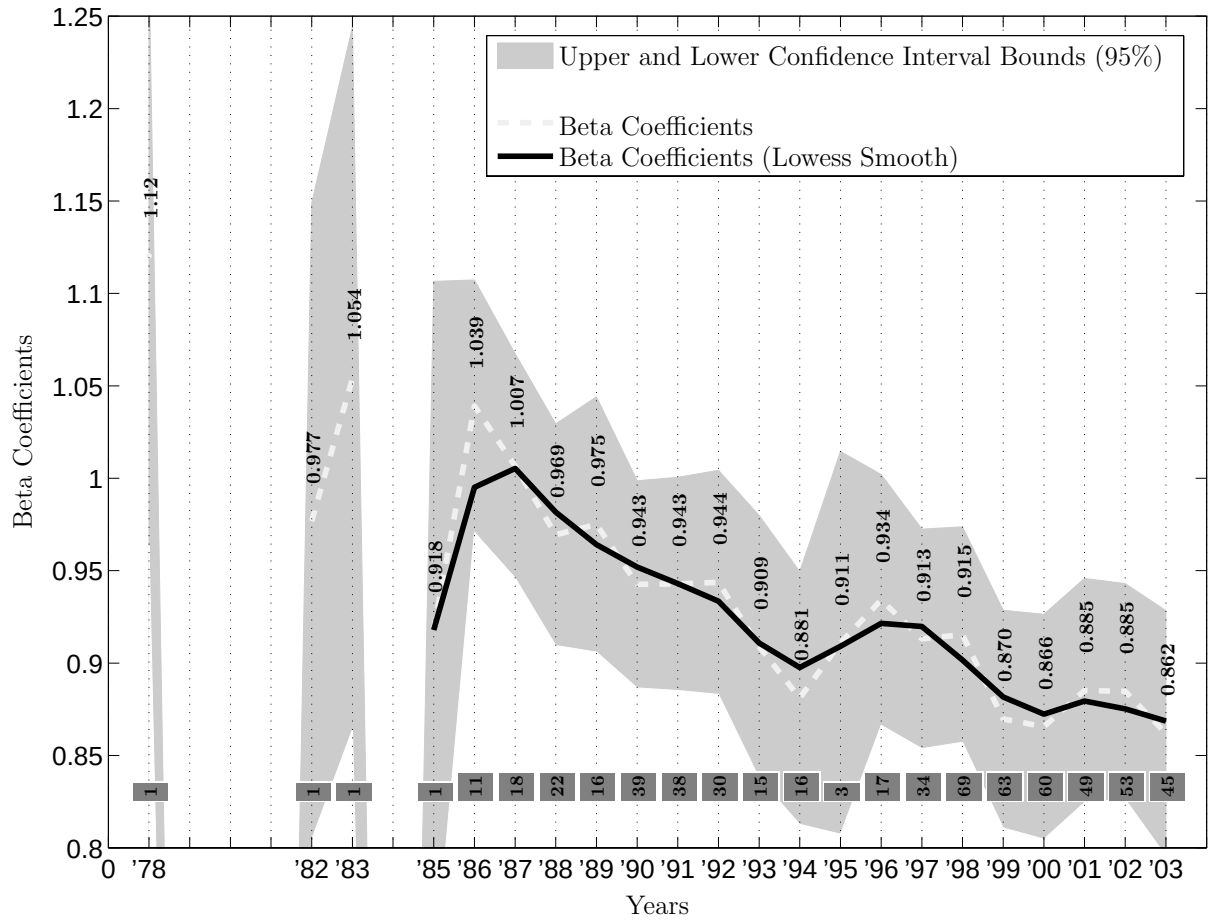

Figure 4. Software Project Productivity Trends (1987-2003)

1992 improvement in productivity due to the confounding effect of an increasing proportion of banking and insurance sector projects (see Figures 3) which are also noted to be the two least productive sectors in the data set (see Figure 8). This is more evidence of the dangers of over-simplistic productivity analysis.

Next we explore other factors that impact software productivity in more detail.

4.3. New Development and Maintenance Project Productivity

First we divide the projects by type into those that are New Development and Maintenance.

ANOVA highlights that there are significant ($p < 0.001$) differences in size and effort between New Development and Maintenance projects with the Maintenance projects being smaller. Given that there is weak evidence for economies of scale this suggests that using raw productivity could be misleading in that Maintenance projects could appear less productive than they really are due to the fact that they tend to be smaller. In addition, Maintenance projects have only started to occur since 1997.

For this reason we examine the model coefficient for the Project Type dummy variable from the regression model used in the previous section. We have $\beta_{NewDev} = 0.1198, p = 0.265$ with 95% confidence limits of -0.091 and 0.331. The interpretation is that a positive value implies more Effort for New Development projects than for Maintenance projects (since the latter will have a zero in the dummy variable). However, this value is not significantly different from zero since it is straddled by the confidence limits. So we conclude, in line with Kitchenham *et al.* [14], that there is no significant difference in productivity between New Development and Maintenance projects.

We explored a step further by building regression models (as in Section 4.1) for both, New Development and Maintenance projects exclusively. In Figure 5, we examine New

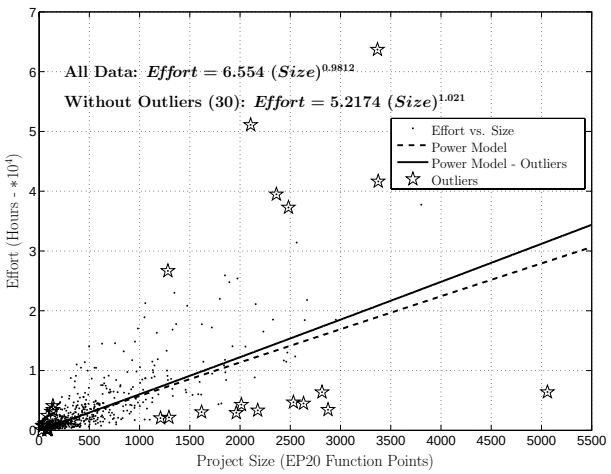

Figure 5. Regression Analysis of Project Size and Effort for New Development Projects

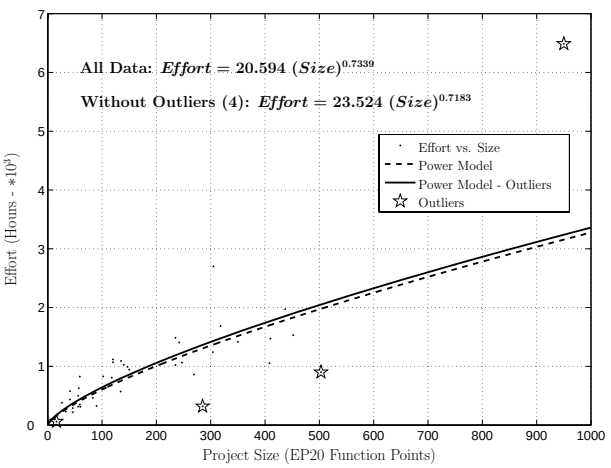

Figure 6. Regression Analysis of Project Size and Effort for Maintenance Projects

Development projects. We can see that even with the removal of 30 outliers (star data points) using Cook's D as in Section 4.1, the value of b changes marginally from 0.9912 to 1.021 suggesting little evidence for anything other than a linear relationship between Size and Effort. However,when we studied the distribution of residuals, this again indicated that the model is heteroscedastic and therefore may not be a good predictor for the entire range of project size.

For Maintenance projects (Figure 6), we observe significant economies scale where $b = 0.734$ with confidence intervals $0.6129 < b < 0.856$. Upon removing 4 outliers (star data points), we observe a marginal change with b reducing to 0.7183 with confidence intervals $0.615 < b < 0.821$ and thus, strongly indicating economies of scale. Hence, for every unit increase of size for Maintenance projects, the marginal effort required to complete the project diminishes. Also, we verified the distribution of residuals which showed that both power models are a good fits for the data i.e. the models are homoscedastic.

To summarise, the New Development projects show approximately constant returns to scale whilst the Maintenance projects show a pronounced economies of scale ($b = 0.7183$). Overall, despite the seemingly similar productivity rates between New Development and Maintenance projects, we would urge some caution in building explanatory models and suggest it may be best to analyse different types of projects separately at some stage to explain causality.

4.4. Sources of Variance

In this subsection, we consider further variables in the data set that are strongly influential on productivity.

Variable	% of variance "explained"
Company	26.2%
Process model	12.6%
Business sector	11.7%
Year	8.4%
Hardware	5.6%

Table 4. Productivity Factors

We applied an ANOVA procedure. The factors identified in Table 4 are the most significant in terms of variance they explain[4] for productivity again defined as $\hat{e}/e$. All variables are significant at least at $p = 0.01$.

The most influential is company, followed by process model and business sector. This is an interesting observation that is essentially consistent with previous studies e.g. [17, 18, 23] which suggest that local models may be more beneficial to companies. We go onto consider the three most influential factors in the following sections, though future work will need to explore other factors such as hardware in more detail.

4.5. Productivity across Companies

We begin by first investigating productivity across companies which accounts for the largest variance explained, i.e. 26.6% with ($p < 0.0001$). Finnish602 comprises 32 different companies with number of projects contributed by each

4 The proportion of variance explained in ANOVA is calculated by dividing the sum of squares between groups by the sum of squares total. This ratio represents the proportion of variance explained.

company varying from 1 to 132. Relabelling all companies contributing 5 or less projects to 'Other' reduces variance explained to 19.8%. However, on removing projects from relabelled companies, variance explained increases to 21.1% again significant at $p < 0.0001$. Interestingly, these results are in tune with the earlier analysis performed by Maxwell and Forselius [16] who found company to be the most important factor influencing productivity. Unfortunately, we have not included the ANOVA plot due to lack of space.

On cross-tabulating company and business sector (which 'explains' the third largest source of variance in productivity), we observe that nearly all companies developed projects exclusive to one business sector with an exception of a few companies that developed projects across banking and insurance sectors. This suggests that companies usually tend to specialise in projects within exclusive business sectors and thus, vary substantially in their productivity in comparison to each other as business sector have different requirements. Also, the choice of technology adopted for development, process models, staff skills, etc. (which are often company specific factors) may interact with each other and cause productivity to vary substantially.

For the time being, with such in-depth analysis being out of scope for the current paper, we move on to explore other significantly influencing factors that impact productivity.

4.6. Process Model and Project Productivity

Next, we investigate the process model used by projects which explained for the second largest variance in productivity (see Table 4). However, it is important to note that this may be an effect of 19 missing values and the presence of 23 different grouping values with many of them having less than 5 observations. Hence, to reconfirm the variance explained by Process Model, we relabelled all but the top 5 most frequently used Process Models to 'Other' and recomputed the variance explained which dropped to 5.14% but still significant at $p = 0.01$.

Our feeling is that contrary to Table 4, Process Model may not that important in explaining variance in Productivity and that it may be acting more as a proxy for other significantly influencing factors. To our support, we performed a Chi-square test on the 2-way contingency table of Process Model and Company resulting in a value of $747.8, p < 0.0001$. This suggests that this may indeed be the case

However, we further analyse this factor simply by investigating grouping values with 30 or more corresponding projects. In Figure 7 we look at the five most frequently used process models in the data set. We found that by including only those projects in Figure 7, Process Model was still a significant variable (at $p = 0.01$) explaining 3.9% of

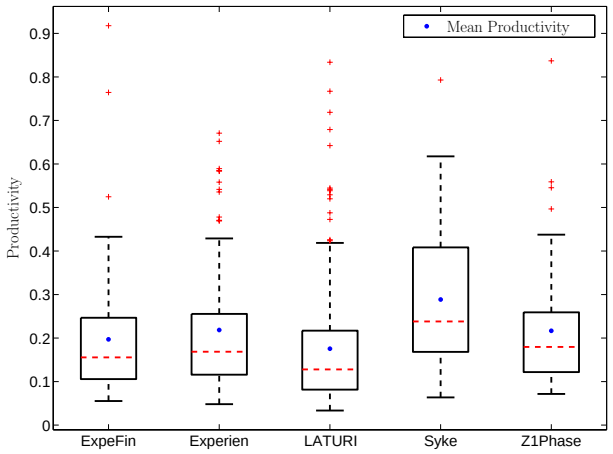

Figure 7. Side by Side Boxplots comparing Productivity by Process Model

variance in productivity. The Syke model with 30 (smallest) observations has the highest mean value of productivity and is broadly spread. By contrast, the Laturi process model has the largest number of observations (194) but the lowest mean and median productivity values. Hence, with an exception of Syke, all other process models seem to have comparable productivity.

4.7. Business Sector and Project Productivity

Lastly, we consider the influence of Business Sector which after the re-analysis of Process Model is left as the third largest source of productivity variance. Recall from Table 2 that the data are grouped into six sectors (Banking, Insurance, Manufacturing, Public Admin, Retail plus an Other category for business sectors that are not widely represented).

An analysis of project productivity (see Figure 8) reveals quite marked differences in productivity (this time measured in the more traditional sense of FPhr^{-1} in order to allow comparisons with the FinnishMF data set [16]). The differences are significant using an ANOVA test ($p < 0.001$). Banking and Insurance are the least productive (and least variable) sectors whilst Manufacturing stands out as the most productive (and most variable).

Interestingly there is relatively little variation in project size between the sectors, other than for Public Admin projects that tend to be substantially larger. For example the mean size for the Banking sector is 536 FPs, 455 for Insurance, but 828 FPs for Public Admin.

We also consider changes in productivity by sector between the pre – 1995 (FinnishMF data) and 1997 – onwards projects (Finnish97+) in Figure 8. Most noteworthy are the

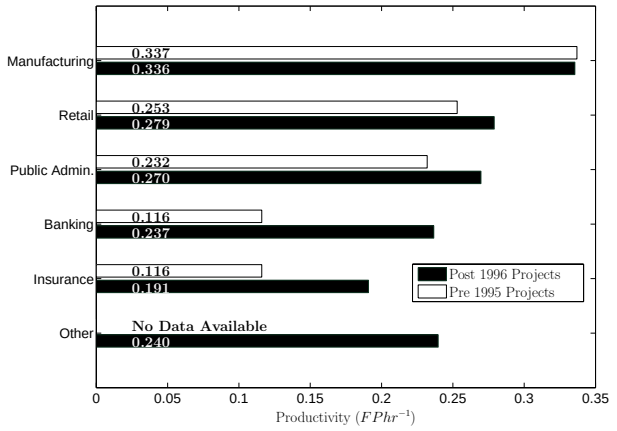

Figure 8. Software Project Mean Productivity by Business Sector

substantial improvements in the Banking and Insurance sectors. It has been suggested by STTF that a possible explanation is that the Banking sector have invested heavily during the 1990s in initiatives such as three level architectures and that more recently they have been reaping the rewards of such investments.

In contrast, Manufacturing exhibits little change, although is still the most productive sector. This implies it may not be wise to always treat projects as being homogeneous since we observe quite different characteristics for different sectors. We also observe that no Sector has decreased in productivity which is perhaps something of a relief!

However, there is again a danger that such analysis could be confounded by other factors such as the uneven distribution of Business Sectors over time (see Figure 3). Thus we use the same procedure as for analysing Project Type and investigate the regression model coefficients (see Table 5). The interpretation remains that positive values imply lower productivity so confirming the above analysis we see Banking and Insurance are the least productive and Manufacturing and Retail the most productive Business Sectors. Differences can be deemed significant when there is no overlap between the confidence limits. Thus, we have Insurance $<$\{PublicAdmin, Manufacturing, Retail\} and Banking $<$ \{Manufacturing, Retail\}. No other differences are statistically significant.

5. Conclusions

There are two groups of lessons to be derived from this analysis. The first, and obvious group, relates to the analysis. The second group relates to the *process* of conducting the analysis. We address each in turn.

Sector	$\beta_{BusSect}$	Lower Bound	Upper Bound
Insurance	0.2434	0.0494	0.4374
Banking	0.1980	-0.0085	0.4046
Public Admin	-0.1766	-0.3934	0.0401
Manufacturing	-0.5572	-0.7846	-0.3298
Retail	-0.3986	-0.6665	-0.1306

Table 5. Comparative Business Sector Productivity

The projects within Finnish602 show the following characteristics with regard to productivity.

- Overall there seems some evidence of productivity improvement with an observed increase of 33% in raw productivity measured as FPhr$^-$1 between the projects that commenced up until 1994 and those that commenced 1997 onwards.
- Whilst the productivity trends by year show some fluctuation, by applying a smoother we see strongest evidence of an improvement during the 1980s and early 1990s followed by subsequent weaker improvement rates. This is confirmed by significant correlation tests between year and productivity.
- There is no evidence of diseconomies of scale, and no significant evidence of non-linearity for software projects other than for Maintenance projects which show pronounced economies of scale.
- In terms of productivity we see little difference between New Development and Maintenance projects, however, there are underlying differences including economies of scale and size with New projects tending to be significantly larger.
- The most significant factors in 'explaining' productivity are in decreasing order of importance (i) Company, (ii) Business Sector, (iii) Year and (iv) Hardware.

Of course, the difficult but rather important question is to what extent can we generalise from these results? The projects are quite diverse but are all based in Finland. It would be interesting to conduct further analysis along the lines of Cusumano et al. [6] to consider what differences, if any, exist between different countries. However, the strongest basis for generalisation is through replication of this analysis for different software projects in different circumstances.

Second, we are aware that presently, there is an increasing move towards the sharing of data and encouraging replications as a necessary basis for meta-analyses i.e. the combining of results from multiple studies. We believe this to

be a good thing and essential if we are able to interpret multiple, but sometimes inconsistent, results. A problem with more complex data sets such as the Finnish data set is that they can be hard to fully understand, and for this reason it could be easy to perform erroneous analysis. We [RP and MJS] made a number of incorrect assumptions during the course of this analysis which were corrected through discussions with STTF. There are, therefore potential risks with analysing complex data sets unless the researchers have a good channel communication with those associated with the actual data collection.

Finally, we are aware that this short analysis has only scratched the surface of a large and complex data set which is potentially a valuable resource for the empirical software engineering community. We believe that the fact that this data is available and the fact that it is enhancing communication and cooperation between industry and researchers augurs well for the future.

Acknowledgments

This work was funded by the UK Engineering and Physical Sciences Research Council under grant GR/S45119.

References

[1] A. Abran and P. N. Robillard, "Function points analysis: An empirical study of its measurement processes," *IEEE Trans. on Softw. Eng.*, vol. 22, pp. 895-910, 1996.

[2] R. D. Banker, H. Chang, and C. F. Kemerer, "Evidence on economies of scale in software development," *Info. & Softw. Technol.*, vol. 36, pp. 275-282, 1994.

[3] C. A. Behrens, "Measuring the Productivity of Computer Systems Development Activities with Function Points," *IEEE Trans. on Softw. Eng.*, vol. 9, pp. 649-658, 1983.

[4] B. W. Boehm, "Software engineering economics," *IEEE Trans. on Softw. Eng.*, vol. 10, pp. 4-21, 1984.

[5] L. Briand and I. Wieczorek, "Resource Modeling in Software Engineering," in *Encyclopedia of Software Engineering*, J.J. Marciniak, Ed., 2nd ed. New York: John Wiley, 2002.

[6] M. Cusumano, M. A. MacCormack, C. F. Kemerer, and B. Crandall, "Software development worldwide: The state of the practice.," *IEEE Software*, vol. 20, pp. 28-34, Nov./Dec. 2003.

[7] C. J. Dale and H. van der Zee, "Software productivity metrics: who needs them?" *Info. & Softw. Technol.*, vol. 34, pp. 731-738, 1992.

[8] Experience Pro Internet Site. www.sttf.fi/eng/products/experience/indexexperience.htm.

[9] D. R. Jeffery and G. C. Low, "Calibrating estimation tools for software development," *Software Engineering Journal*, vol. 5, pp. 215-221, 1990.

[10] M. Jørgensen, U. Indahl, and D. I. K. Sjøberg, "Software effort estimation by analogy and 'regression toward the mean'," *J. of Systems & Software*, vol. 68, pp. 253-262, 2003.

[11] B. Kitchenham, "Empirical studies of assumptions that underlie software cost estimation models," *Info. & Softw. Technol.*, vol. 34, pp. 211-218, 1992.

[12] B. Kitchenham, "A procedure for analysing unbalanced datasets," *IEEE Trans. on Softw. Eng.*, vol. 24, pp. 278-301, 1998.

[13] B. Kitchenham, "The question of scale economies in software - why cannot researchers agree?" *Info. & Softw. Technol.*, vol. 44, pp. 13-24, 2002.

[14] B. Kitchenham, S. Pfleeger, B. Mccoll and S. Eagan,"An empirical study of maintenance and development accuracy". *J. of Systems & Software*, 64, pp. 57-77, 2002.

[15] B. Kitchenham and E. Mendes, "Software productivity measurement using multiple size measures," *IEEE Trans. on Softw. Eng.*, vol. 30, pp. 1023-1035, 2004.

[16] K. D. Maxwell and P. Forselius, "Benchmarking software development productivity," *IEEE Software*, vol. 17, pp. 80-88, 2000.

[17] K. D. Maxwell, L. Van Wassenhove, and S. Dutta, "Performance evaluation of general and company specific models in software development effort estimation," *Management Science*, vol. 45, pp. 787-803, 1999.

[18] E. Mendes and B. Kitchenham, "Further comparison of cross-company and within-company effort estimation models for web applications," presented at *10th IEEE Intl. Softw. Metrics Symp.*, Chicago, USA, 2004.

[19] K. Moløkken and M. Jørgensen "A review of surveys on software effort estimation", *2nd IEEE/ACM Intl. Symp. on Empirical Software Engineering*, pp. 223-230, 2003.

[20] R. Premraj, B. Twala, P. Forselius and C. Mair, "Productivity of Software Projects by Business Sector: An Empirical Analysis of Trends," Late Breaking Paper presented at *10th IEEE Intl. Softw. Metrics Symp.*, Chicago, USA, 2004.

[21] P. Sentas, L. Angelis, I. Stamelos, and G. Bleris, "Software productivity and effort prediction with ordinal regression," *Info. & Softw. Technol.*, vol. 47, pp. 17-29, 2005.

[22] C. E. Walston and C. P. Felix, "A method of programming measurement and estimation," *IBM Systems Journal*, vol. 16, pp. 54-73, 1977.

[23] I. Wieczorek and G. Ruhe, "How valuable is company specific data compared to multi-company data for software cost estimation?" presented at *IEEE 9th Intl. Symp. on Softw. Metrics*, Ottawa, 2002.