Reliability of function point counts

P. Kampstra and C. Verhoef

VU University Amsterdam, Department of Computer Science,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

{pkampst,x}@cs.vu.nl

Abstract

An important characteristic of any software is its size. Frequently used metrics for measuring the size of software are source lines of code (SLOC) and function points. Lines of code are easy to count and seem unambiguous (but different definitions can cause variations of 500%). Function points are normally counted by certified professionals, which may introduce differences between counters. In this article we survey existing literature on inter-rater reliability that classically involves recounts, but recounts are hardly possible in practice. We propose multiple methods to test for differences between raters that do not involve recounts. In a case study of 311 projects and 58143 function points from a large institution we determined that function point counts are a reliable base. Using our proposed method, we did not find statistical evidence for systematic differences between counters, and recounts were not necessary for that. So in this organization, the function point counts are a reliable data source for IT management.

Keywords and Phrases: Empirical software engineering, quantitative software engineering, software metrics, function points, inter-rater reliability, function point analysis

1 Introduction

An important characteristic of any software is its size. Frequently used metrics for software size are source lines of code (SLOC) and function points [2, 1, 13, 8, 10]. Lines of code are easy to count, but for function points certified professionals are a good practice. Lines of code seem unambiguous, but for some programming languages that allow multiple statements per physical line, different definitions still can cause differences of 500%. More importantly, when used as a normalizing metric, LOC has been proven to penalize modern programming languages [15]. Function points may introduce differences due to different counters and methods.

One of the perceived problems with function point analysis [19] is that it does not produce exact results. Different measurements sometimes produce different results. This is a reason for some executives to abandon this metric altogether. However, in fact also measuring length is not exact, and suffers from the same effects as function point analysis. But we still use meters. In this article we will focus on differences between function point counters, or in other words the inter-rater reliability. We will survey the existing literature on the reliability of function point counting. This is classically measured by repeated function point counts. We propose various techniques to assess the systematic differences between counters where recounts are not necessary.

In a case study of 311 projects with in total 58146 function points, counted by professional function point analysts from a large institution, we determined the suitability of function point counts. We did not find statistical evidence that counters counted differently, so the function point counts are in this organization a reliable metric for decision and control.

The method we used relies on statistical principles and is applicable to virtually any metric. For example, consider the case where we are manually counting the number of words per book. For books that are randomly distributed to word counters, systematic differences between counters are detectable without doing recounts. Because the books are randomly assigned to counters, the word count distribution we find for each counter should be not statistically different from the total distribution. If the means per counter show a statistical difference, this is a strong indication for statistical differences between counters or for non-random assignments. Note that this method only tests for systematic differences: if you recount a manually counted book, it is not uncommon to find a difference of a few words. While a rough method already spots systematic differences, a refinement is possible by adding more information to the equation. For instance, given that thicker books usually have more words, testing for systematic differences between counters is improved by compensating for the thickness. In this paper, we use these observations to investigate potential differences between function point counters by using two approaches. First, we examine function points counted per counter. Second, we also take the project cost into account.

Organization The remainder of this paper is organized as follows. In Section 2 we survey the existing literature on inter-rater reliability and compare the literature with the methodology that we propose. In Section 3 we will describe the data from our case study. In Section 4 we will test for differences between individual counters, and in Section 5 differences between groups of counters are tested. In Section 6 a model is constructed between cost, function points and the influence of counters, to further investigate the presence or absence of systematic differences between counters. Section 7 discusses results on a more limited data set, where differences between counters where detected. This section also reviews potential limitations of our proposed methodology. Finally, Section 8 provides a summary and conclusions.

2 Related work

In 1992 [17, 18], an extensive field experiment was conducted to address the questions of inter-rater reliability and inter-method reliability. The experiment set out was to have different counters analyze the same system, and test statistically whether the outcomes were the same. In the experiment 27 systems were counted twice using standard IFPUG-methodology, and 21 systems were counted twice using a different methodology, the so-called Entity-Relationship method. In the experiment an inter-rater reliability, defined in terms of differences between two counts, was found of about 12% in median between people in the same organization using the same method. A test for systematic statistical significant differences between those counts showed no evidence for differences.

However, we doubt that the research design justifies the latter conclusion on the statistical significance of the results. Let us explain. When doing a paired two sample t-test, normally the two samples should not be randomly assigned to from the same pool. Otherwise, one is testing whether this pool is equal to itself or not, and the

probability of rejecting the null hypothesis is theoretically exactly equal to the p-value used. That is, if other assumptions to the paired t-test are not violated that impact the probability of false positives, such as the assumption of a normal distribution [3, 7]. As we understand in the research design of Kemerer [17], the two counters for each methodology were not kept the same during the experiment. In fact they were randomly chosen as far as the researchers could oversee. Statistical tests confirmed that: they showed no evidence for non-random assignment between the groups. So the assessment of systematic differences between counters should not be based on a paired t-test. However, in that paper we read:

> The results of a paired t-test of the null hypothesis that the difference between the means is equal to 0 was only -0.61 (p = 0.55), indicating no support for rejecting the null hypothesis. The power of this test for revealing the presence of a large difference, assuming it was to exist, is approximately 90% [8, Table 2.3.6]. Therefore, based on these results, there is clearly no statistical support for assuming the counts are significantly different.

Therefore, we must conclude that they did use a paired t-test on two randomly assigned samples with a p-value of 0.10 (see the cited table [5]). Note that for *small* differences like those between function point counters, in the same table the power listed is only 19%, even if we accept a 10% false positive rate. The false positive rate of 10% is likely to be higher due to the expected non-normal distribution of the function points, as shown later in this paper. So we can only conclude that their claims in regard to statistical significant differences between counts using the same method were not supported by the described tests. Therefore, it is unknown which amount, if any, of the 12% median difference found between all 27 pairs of measurements using the standard method is explained by *systematic* differences between counters. Therefore, we advise to use the 12% with caution.

In 1990 [23], different figures were reported: a variance about the mean within an organization of within 30 percent, along the lines of a previous study inside IBM. Across organizations the variance was possibly higher, which could not be statistically verified. Two relatively small systems of about 58 and 40 function points were counted by 22 experienced analysts divided among 7 organizations. The system of about 58 function points was also counted by 20 inexperienced analysts, who estimated significantly more function points. Systematic differences between experienced counters were not tested for, so it remains unclear whether some counters systematically counted more function points, or that there was only random variation in individual measurements.

In a 1998 Inter-counter Consistency Checking Experiment by the UK Software Metrics Association (UKSMA), differences of up to 50%, based on documented functionality, were observed for novice estimators rating logical files [26]. Estimates for this category could be rapidly improved with some environment-specific training.

We have not found any peer-reviewed studies on inter-rater reliability that were conducted after 1992, which is more than 15 years ago at the time of writing this paper. In fact, we also have not found any study that tested for *systematic* differences between experienced counters, perhaps except for the work on counting rules clarification by Kemerer in 1992 [18]. A likely reason for the rarity of such research, is that the recounts classically involved are expensive for the organizations doing the function point analysis. Therefore, we propose a methodology that does not impose new budget

requirements on the organization doing the function count analysis, and still assesses the inter-rater reliability.

Our research design is different: we have not asked counters to repeatedly carry out a function point analysis (FPA) of the same system, but we collected function point totals of finalized IT projects. The function point analyses were carried out by multiple counters. Some of the criticism on function points is that when two counters estimate the function point total of a system, the answers may be different. In our opinion, function point counting is a stochastic process, inevitably leading to differences. Rather than trying to ban the differences, we propose to recognize the stochastic nature of function point analysis, and take that as the fundamental viewpoint on function points. In other engineering disciplines the stochastic nature of certain processes is not only recognized, but is used as a basic tool to construct systems. Let us explain. When van Doorne's transmissions invented the Variomatic (an automated kludge for automobiles), this consisted of a number of metal bands that fit very closely together. While in the lab situation the researchers could produce small numbers of the kludge, it turned out to be a problem to industrialize it. Apart from oven heating problems that disturbed production start-up, it was also close to impossible to deliver the various metal bands with ultra high precision. The solution was not to improve the reliability of the production process, but the other way around. You just make a large number of metal bands, and after production you collect them, measure them, categorize them, and construct a perfect fitting transmission system. So high tolerance, low cost, and high precision can go hand in hand in other engineering disciplines. Therefore, we should not abandon imprecise metrics as useless. We better recognize stochastic effects and if possible exploit their properties. We think that there are such opportunities in software engineering, of which this paper testifies.

To statistically test whether or not different counters produce different results due to differences in counting habits one would ideally set up a controlled and appropriate experiment to test this. For example, each counter counts a number of systems of different sizes and the measurements are compared. There will then be a number of observations for each of the systems; each observation being the number of function points produced by one of the counters. The question of interest is whether or not the counters are different with respect to the number of function points counted. If the counters produce comparable numbers of function points in counting the same system, then the conclusion is justified that there are no differences in counting behavior between the various counters. In statistics this way of working is known as experimental design modeling. The interest is not in predicting the value of one variable by using the values of related factors, but the interest is mainly in comparing the effects of two or more factors.

Of course the sample size should be chosen not too small to be able to powerfully test the hypothesis that there is no effect caused by counter behavior. With powerful we mean that the chance of accepting the null hypothesis when the alternative hypothesis is true is not unacceptably high. However, in practice it is often not possible to design special experiments for the purpose of credibility checks. A number of different counters have counted different systems of different sizes and no more information is available. There are no two counters who have counted the same system, so a direct comparison between counter behavior is impossible. This often is the case in practice. However, it remains a debatable point whether a specific effect caused by counter behavior plays a role in counting the number of function points.

Let us explain. When we investigate the data set of function points we want to know whether the set is homogeneous or heterogeneous from a counter point of view.

Suppose that we have good reasons to postulate a relationship between the size of a system and the costs of building the system. Or, we expect that there is a relationship between the chance of IT project failure and the system size. Note that the relationships do not have to be linear, but can take every conceivable form. We then want to use the observational in-house data to test statistically whether our theory is supported or not by the data and, if so, to obtain an estimation of the parameters of the functional relationship. However, if the data is biased because of individual counter behavior then the set is not usable to test our hypothesis. In this paper we show that a test on homogeneity of the data set is also doable when no direct comparison between the measurements of different counters is feasable.

For all function points, our counters used the same method: an approved adaptation of the IFPUG-standard. Backfiring, where SLOC are converted into the language independent function points using conversion ratios [14], was not used. Different methods are known to produce extremely different results [12], but as all counts were done using the same method we did not address the inter-method reliability question. Our focus here concerns the inter-rater reliability.

Our research design to test whether the function point counters are equal is done in a stochastic manner. We give another example to illustrate this. Suppose we would want to find out whether two dice are the same, or, in other words reliable as in fair and unbiased. Then by throwing the dice, we should not conclude that the dice are unreliable if the outcome of the dice is different. We would say that the dice are different if the *probability* of a certain outcome differs between the two dice. Exactly the same, we investigate the function point counters. Some stochastic process produces a string of function point totals out of a universe of IT projects that need to be counted. We are not interested in the question whether the exact outcome will be given if we provide one system to all counters, but we want to know whether the probability that they give the *same* outcome is not different. And by recognizing the stochastic nature of function point counting, this becomes in fact feasible. In the remainder of this paper we use formal statistical tests to do this. These tests give us insight in whether different "FPA-dice" produce different results.

3 Our data set

From a large institution, we received data on a portfolio of IT projects to perform an audit on the effects of a software improvement project in the organization. Before doing the audit, we assessed the accuracy and plausibility of the function point data. Our data set consists of 311 IT projects with a total size of 58146 function points. The function point counts are done by 17 function point counters, of which 14 are internal, and three are hired externally from a specialized company doing function point counting only. All the counts measured actually delivered functionality, not estimates of proposals. For the methodology this does not matter, but one should be careful when mixing both type of counts into one group, because the counts are usually different on average, because the scope of projects tends to increase [22].

When you are going to audit whether targets are met or not, you have to exclude the variant that more function points are reported than are delivered, since this would boost IT productivity. We will therefore test whether or not this was the case.

outlier In the data set of 311 projects, there is one project with a size of just 3 function points, while all other projects are 15 function points or more. Because function points

were never intended for small projects, and this project would be spotted as an outlier on every graph, we left this project out of the analysis presented in the rest of this paper. We ran all analyses in parallel to ascertain that the removal of this project did not influence the results. To be precise, the largest difference would be in Table 6 presented later on in this paper. A p-value for the Shapiro test for the distribution of external counter number 2, who counted this project, would drop to 0.0228 from 0.12, but that does not alter the conclusions. Hence, without loss of generality we discarded the outlier.

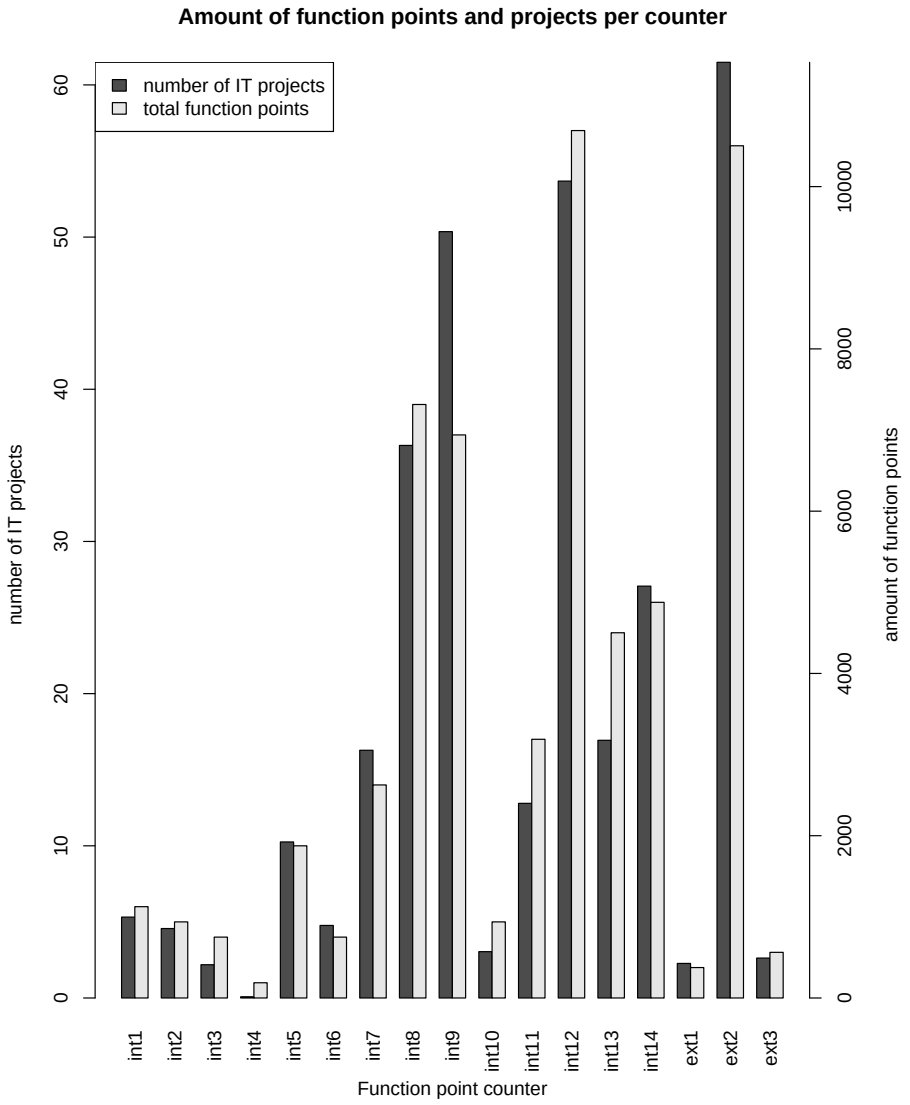

Figure 1: Visualizing the total amount of function points and projects counted per function point counter.

4 Individual counters

The organization employs 17 function point counters, of which 14 are internal, and three are hired externally from a specialized company doing function point counting only. In Figure 1, we plot the total amount of function points measured by each counter. The abbreviation int stands for an internal function point counter, and ext is short for an external one. It is immediately visible from Figure 1 that some counters do a lot of counting and some do very little counting. For instance, external counter 2 is the star counter: more than 11000 function points. Internal counter 12 is second in row with about 10000 function points. Furthermore, there are a few counters who counted almost no function points.

Now that we have an initial idea of the totals per function point counter in the research set, we are also interested to know how many IT projects each counting specialist took care of. We also plotted this in Figure 1, as the dark-grey bar. There is a clearly visible relation between the two bars. Indeed, a (superfluous) test using Pearson's product moment correlation coefficient [30] gives a p-value of $<$1e-10, proving the visually obvious correlation. So, it is unlikely that for example a counter always counts the large projects, and thus counts not too many projects, but does count a large amount of function points, or vice versa. These views combined already give us evidence that incoming IT projects are indeed randomly assigned. If there is no random assignment, our methodology will usually spot differences caused by biased assignment, which we indeed did not find as is presented further in this section.

To obtain a more in-dept view we constructed another view in which we visualize the size-range per counter. We do this via a so-called box and whiskers plot, or box-plot for short [29, 24]. A box-plot is just a visual form to summarize the data with as few points as possible. One well-known point is the median, dividing the ordered data in two equally sized groups. The other points we use are of the family of quantiles. In general, a *quantile* is any of several ways of dividing your ordered observations into equally sized groups. An example of a quantile is the *percentile*: this divides your data into 100 equally sized groups. Likewise, *quintiles* divide into 5 equally sized groups, and *quartiles* divide data into 4 equally sized groups. You can obtain a fairly good idea of the distribution of your data by dividing it into quartiles [29, 24]. The boxes in Figure 2 are limited by the first and third quartile, and the white line inside the box is the median so that skewness of the data is immediately visible. The shaded box encloses the middle 50% of the observed values. Its length is also called the inter-quartile range, which is an important measure that is less influenced by extreme values. The whiskers are some standard span away from the quartiles, we used as standard span 1.5 times the inter-quartile range. Points that go beyond the whiskers are potential outliers and they are drawn individually.

The distribution of function points tends to have a large right-tail, which means that very large values can occur at the high-end of the distribution. The box-plots indeed show that there are few values that are smaller than the rest, while there are some values that are shown as outliers in the box-plot at the high-end. This indicates that we are dealing with a function point distribution that shows signs of control; the outliers shown are as expected. Furthermore, the box-plots do not show very strange deviations, except that external counter 1 appears to deliver higher function point totals than others. Clearly this counter just counted somewhat larger projects, or there is a deviation that too high function point totals are reported. All in all, an external counter usually has no interest in counting too many function points, since it is their profession to count correctly. Another indication is that external counters count more function points per

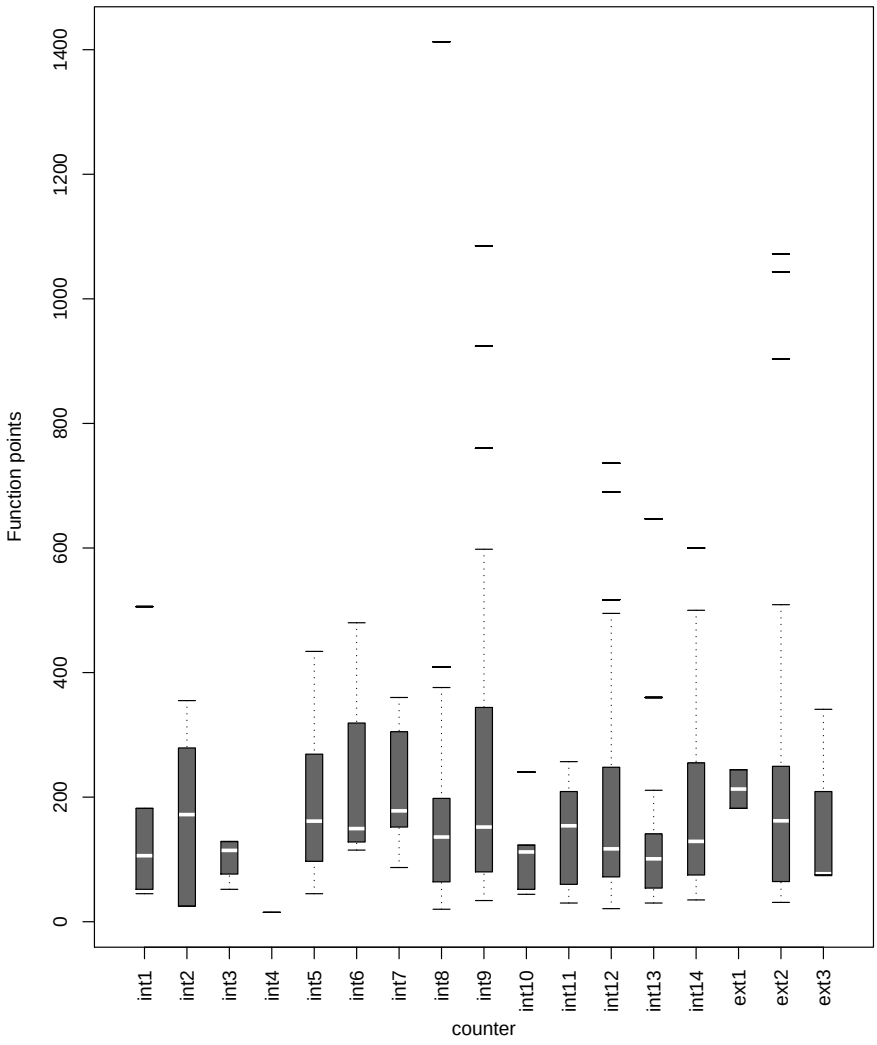

Figure 2: Box-plots of the different function point totals per function point counter.

project than internal counters, providing initial evidence that there is no boosting of function point totals in place, by the internal metrics people.

After this initial visualization of the function point distribution per counter, we want to know this distribution for real. Therefore, we further zoom in on the distribution of the function point totals per counter. As a first quantitative indication, we summarize for each counter a ten-point summary statistic in Table 1. The abbreviations in Table 1 are self-explanatory, except sd, which is short for standard deviation, and geom which stands for geometric average. From Table 1, we find that function point counters int4, ext1 and ext3 only rarely counted function points. We quantified now much more con-

cnt	#	min	1st qu	med	mean	3rd qu	max	sd	sum	geom
int1	6	45	61	106	166	167	506	174	997	115
int2	5	25	25	172	171	279	355	148	856	101
int3	4	52	89	114	102	128	129	36	410	97
int4	1	15	15	15	15	15	15	NA	15	15
int5	10	45	98	162	192	262	434	127	1923	155
int6	4	115	134	150	224	238	480	172	894	187
int7	14	87	156	178	218	297	360	89	3053	201
int8	39	20	64	136	175	198	1413	225	6810	118
int9	37	34	80	152	255	344	1085	262	9445	159
int10	5	44	52	112	114	123	240	79	571	95
int11	17	30	60	154	141	209	257	78	2399	115
int12	57	21	72	117	177	248	736	159	10068	127
int13	24	30	55	101	132	138	647	131	3176	99
int14	26	35	79	129	195	254	600	156	5076	145
ext1	2	182	198	213	213	228	244	44	426	211
ext2	56	31	65	162	206	249	1072	222	11532	138
ext3	3	74	76	77	164	209	341	153	492	125

Table 1: Ten-point summaries for the various function counters. For each counter (cnt), minimum (min), first quartile (1st qu.), median (med), mean, third quartile (3rd qu), maximum (max), standard deviation (sd), summation and geometric average (geom) are reported. NA stands for not applicable.

crete how the data is skewed, for instance by the sometimes large differences between the median and the mean. But we gain the most insight in a plot of the various density functions. So we calculated the empirical probability density functions [32, 27] and show them in a beanplot [16]. In Figure 3 all the counters are shown. The small white lines reflect individual projects that are counted by a counter, and the density estimation is shown in black. The prominent black line reflects the geometric mean. A log-scale is chosen, so that individual points do not overlap and the distributions obtain a Gaussian shape. The plot shows that the distributions do not have entirely different locations. Only internal counter number 4 appears to be off, but this counter has done only one measurement, so this is not statistically relevant.

Next we want to understand whether the density functions we depicted in Figure 3 are truly different from each other. Of course, when we look at the density functions we spot differences, but how different need these differences be before we would say that the function point counters are actually counting with systematic differences? In order to find out we used a formal test to asses this hypothesis.

The performance of each counter is described by the cumulative distribution function (CDF) of the number of function points measured by that counter. As discussed before the number of function points counted by counter i is considered as a random variable. Let $F_i(x)$ denote the cumulative distribution function of the number of function points counted by counter i. So $F_i(x)$ gives the probability that the number of function points counted by counter i is less than or equal to x. We test the hypothesis that $F_i(x)$ and $F_j(x)$ are equal for all i and j, $i \neq j$. We therefore used the Kolmogorov-Smirnov goodness of fit test, or KS-test in short [20, 28, 6]. If the hypothesis is true we conclude that the counting skills of the counters are not different from each other. Unfortunately, we do not know the exact theoretical CDFs of the individual counters, but we are able to approximate them via their empirical CDFs. We have to assess $F_i(x)$ for each counter i on the basis of the number of systems measured by this counter. So for each counter i we derived the empirical CDF $EF_i(x)$ of the

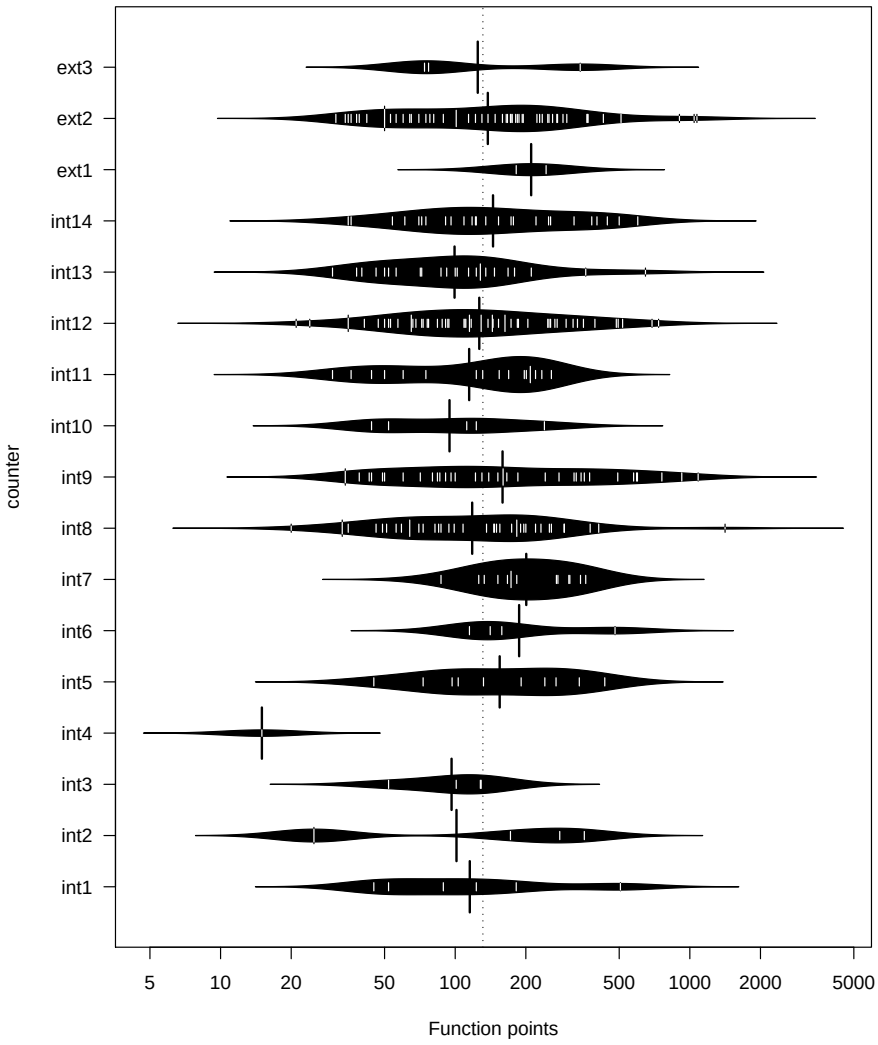

Figure 3: A beanplot for the function points counts per counter. Note the log-scale.

counted number of function points and applied the KS-test to statistically test whether $EF_i(x) = EF_j(x)$, for all i and j, $i \neq j$ holds.

The KS-test is a so-called distribution-free test, which means that it is applicable to any distribution of the data. It measures the maximal vertical distance between the cumulative distributions (CDF) of two data sets. The beanplots in Figure 3 are (empirical) density functions and not cumulative distributions, needed for the KS-test. However, this is not a problem as it is possible to obtain the cumulative distribution from the density function and vice versa. The maximal vertical distance between two CDFs measured by the test is known as the KS-test statistic. If this value is very small,

$p > 0.10$		No evidence against H_0: data seems consistent with H_0
$0.05 < p \leq 0.10$	.	Weak evidence against H_0 in favor of the alternative
$0.01 < p \leq 0.05$	*	Moderate evidence against H_0 in favor of the alternative
$0.001 < p \leq 0.01$	**	Strong evidence against H_0 in favor of the alternative
$p \leq 0.001$	***	Very strong evidence against H_0 in favor of the alternative

Table 2: Symbolic notation for various ranges of p-values with qualitative explanations of their meaning.

counter	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	e1	e2	e3
i1	-																
i2		-															
i3			-				*										
i4				-										.			
i5					-												
i6						-											
i7			*				-	.		*		*	**	.			
i8							.	-									
i9									-								
i10							*			-							
i11											-						
i12							*					-					
i13							**						-			.	
i14				.			.							-			
e1															-		
e2													.			-	
e3																	-

Table 3: Comparisons of individual function point counters with each other with a KS-test.

this is an indication that both CDFs are not fundamentally different. A related metric to indicate how strong the evidence supporting this hypothesis is the p-value. In Table 2, we introduce symbolic notation indicating in qualitative terms which evidence range is meant by which p-value. For instance, if a p-value is smaller than 0.001, there is very strong evidence that the hypothesis is not true, in favor of the alternative.

In Table 3, we summarize the results of carrying out 136 KS-tests, comparing the CDFs of all individual function point counters against each other, except themselves, hence 136 KS-tests. The null hypothesis, denoted H_0, is that both CDFs based on the data coincide. For instance the first row of Table 3, shows that all the KS-tests are in the nothing or no-stars category. This means according to Table 2 that there is no evidence to reject the null hypothesis. In other words, there is no evidence, based on this formal test, that the counting practice of counter 1 differs from the counting practices of any of the other function point counters. Note that we made Table 3 symmetric (so the first column is the same as the first row). Internal counter 7 differs in 6 cases from other counters. The more stars, the stronger the evidence that the differences are due to different counting practices. Internal counters 4, 13 and 14 and external counter 2 show a dot, indicating weak evidence. All in all, these differences give us input for a more in-dept analysis to investigate why both counters seem to differ from other counters.

Power of the test In formal testing of an hypothesis two types of errors are distinguished: rejecting the hypothesis when it is true (error Type I) and not rejecting the

counter	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	e1	e2	e3
i1	-																
i2		-															
i3			-				*								*		
i4				-													
i5					-												
i6						-											
i7			*				-	**		.	*	**	***			*	
i8							**	-							.		
i9									-				*				
i10							.			-					.		
i11							*				-				*		
i12							**					-			.		
i13							***		*				-	.	*	.	
i14													.	-			
e1			*					.		.	*	.	*		-		
e2							*						.			-	
e3																	-

Table 4: Comparisons of individual function point counters with each other with a t-test.

p-value category	notation	meaning	amount	% of total
nothing		$p > 0.10$	120	88.24
dot	.	$0.05 < p \leq 0.10$	6	4.41
single star	*	$0.01 < p \leq 0.05$	7	5.15
double star	**	$0.001 < p \leq 0.01$	2	1.47
triple star	***	$p \leq 0.001$	1	0.74
Total number of combinations			136	100.00

Table 5: Summary of the various ranges of p-values found in the individual t-test comparisons between function point counters.

hypothesis when it is false (error Type II). Our method of testing controls the chance of making the Type 1 error. Ideally, the test also controls the probability of making error Type II satisfactory. Let us denote the probability of making the Type II error with β. The power of a test is defined as $1 - \beta$. In the case of the distribution-free KS-test the power of the test is unknown. However, it is clear that the power increases when the sample size increases. Especially, when the number of systems counted by some counter i is rather small and the systems counted are also of about the same size the approximation of $F_i(x)$ by $EF_i(x)$ will not be very accurate. The probability that the null hypothesis will be excepted when it is in fact false (error Type II) will in that case increase as a consequence.

If we assume that the distribution of the function points per counter is in fact a log-normal distribution, we are able to use a stronger, more powerful, statistical test, namely the t-test [30]. Later in this article in section 5 (Table 6) it is shown that, after correction for multiple tests, there is no statistical evidence for non-log-normality of the distributions. In Table 4 we show the results for the t-test. The results are similar to the KS-test, but the number of stars is higher. As discussed the power of the t-test is much stronger, so this test produces more interesting results. We are not too surprised that external counter 1 differs, since we already noticed from the box-plots in Figure 2 that this counter is at the high end in function point totals. In Table 5, we

summarized the various p-value categories for the 136 t-tests. The table shows that 120 of the 136 combinations do not show evidence of differences, which is about 88% of the combinations. More interesting is that there is one combination that scores a triple star. There appears to be a huge difference between internal counter 7 and internal counter 13. The actual p-value of this particular pair is 0.0006114. This is quite low, so there could be something strange going on. We however have to take into account that we did $16 \cdot 17/2 = 136$ tests. In such cases, one should use a correction on the p-value. The most well known method is the Bonferroni correction [9, p. 339], which amounts to simply multiplying the p-value by the number of tests. Note that the Bonferroni correction is not unquestioned [25], but in this article we combine it with single tests, and also show all uncorrected p-values (for example, many low p-values indicate a problem). By multiplication of the lowest p-value with the number of tests we obtain a corrected p-value of 0.0813, which is acceptable and gives only a weak indication of possible differences.

A single test for multiple comparisons

We also tested for differences between counters in one single test, so that compensation for multiple tests is not necessary. Normally, this is the preferred way to start an analysis. However, when differences are found, or if a closer look is appreciated, one has to resort to the previously mentioned methods anyway to know where there are differences. To keep the ordering of increasingly sophisticated methods in this article, we present these single tests now and not earlier.

If we do not use assumptions on the underlying distribution, we may still use a Kruskal-Wallis rank sum test [6] to test for differences between groups; in this case between the groups that are formed by the projects counted by different counters. The test gives a p-value of 0.3460, which gives no indication for differences between groups. The Kruskal-Wallis test tests for differences in location, not in distribution like the distribution-free KS-test we used earlier.

Assuming a log-normal distribution with equal variances for the distribution per counter allows us to utilize an ANOVA-model [30]. The F-test [30] for the inclusion of counters in the model gives a p-value of 0.2092, indicating no differences between counters. Note that the t-test we used makes less assumptions than its combined ANOVA-variant, namely it assumes no equality of variance of the compared function count distributions. These single test methods provide reassurance for the lack of statistical evidence for differences we found earlier.

5 Comparing counter groups

In our first analysis we have shown that there are not many indications for differences between individual counters, with a few exceptions. In order to come up with an answer to the question whether a software process improvement project resorted the desired effect, we have to exclude the following possibility that we already alluded at. Suppose the internal counters have a vested interest in counting more than is produced. In this way, productivity can turn out to be higher, while in effect this is not the case. To that end, we analyzed the data of the function point counters in two groups. The group of internal counters, who might have that interest, and the group of external counters, whose profession it is to count the correct amounts. Delivering erroneous

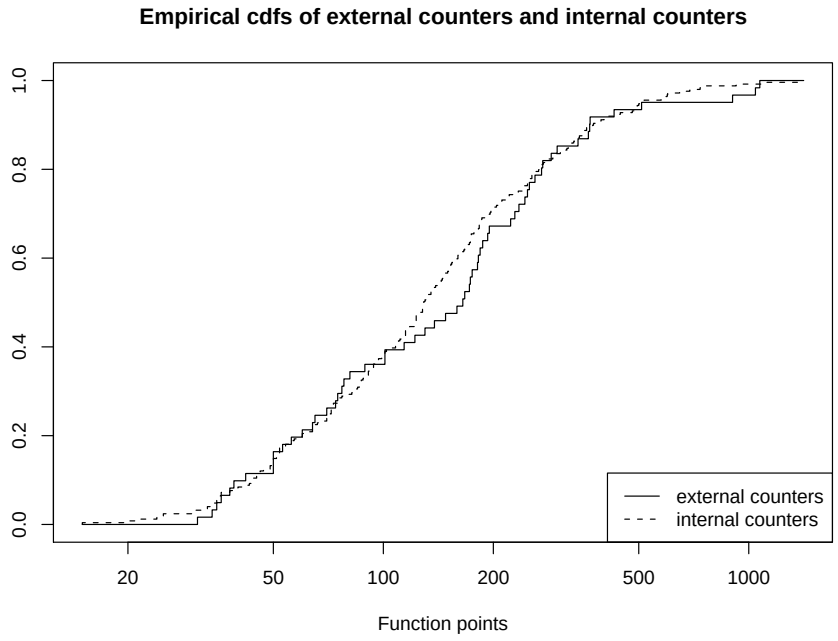

Figure 4: Comparison of CDFs of internal and external function point counters.

results exposes them and their firms to litigation risks due to contracts that are based on their results.

In Figure 4, we plot the cumulative distribution functions of the group of internal counters and the group of external counters. The plot shows no sign of systematic differences between the two groups. We also carried out a formal test, the Kolmogorov-Smirnov goodness of fit test, which gives a p-value of 0.449, giving no indication that the distributions are different. A t-test gives a p-value of 0.5647, also indicating no difference.

For this t-test we give a 95%-confidence interval of the difference (95% of such intervals contain the true value of the difference). In this case the interval on a log-scale is $[-0.316, 0.174]$. In linear representation this is between 27% lower for internal counters to 19% higher for internal counters. A difference of up to 27% sounds at the high side, but given that differences in IT metrics, for example between estimates and finals, can easily be a factor 2 or more, this is entirely reasonable. Seen from the mean value of function points, the largest project is 977% larger, and the smallest project is 89% smaller (this is after filtering one outlier). The largest project is 9320% larger than the smallest one, so the scale is much larger than the confidence interval. Next to that, 27% is on the outer bound of the interval, and the real value is likely to be much more in the center of the interval. We conclude that even if there is a difference between those groups, the difference will not have a significant impact.

As an extra assurance to the results in this section and the previous one, we also tested whether individual counters give different results than all the other counters grouped together and whether the distributions are log-normal. In Table 6 we show

counter	count	mean	median	shapiro		KS-test		U-test		t-test	
int1	6	166.17	106.0	0.6621		0.9639		0.6324		0.7349	
int2	5	171.20	172.0	0.0971	.	0.4531		0.8131		0.6752	
int3	4	102.50	114.5	0.1087		0.2368		0.3888		0.2420	
int4	1	15.00	15.0	1.0000		0.2718		0.0853	.	1.0000	
int5	10	192.30	161.5	0.9126		0.8759		0.4644		0.4844	
int6	4	223.50	149.5	0.1265		0.4643		0.4302		0.3445	
int7	14	218.07	178.0	0.3695		0.0215	*	0.0226	*	0.0022	**
int8	39	174.62	136.0	0.5973		0.8554		0.4245		0.4094	
int9	37	255.27	152.0	0.2138		0.1630		0.2939		0.2077	
int10	5	114.20	112.0	0.6269		0.6166		0.3367		0.3440	
int11	17	141.12	154.0	0.0187	*	0.4077		0.6773		0.4468	
int12	57	176.63	117.0	0.9095		0.7819		0.6952		0.7195	
int13	24	132.33	101.0	0.5376		0.0847	.	0.0588	.	0.0630	.
int14	26	195.23	129.0	0.6942		0.9367		0.5485		0.5159	
ext1	2	213.00	213.0	1.0000		0.3658		0.3186		0.1610	
ext2	56	205.93	162.0	0.1216		0.6087		0.6691		0.6518	
ext3	3	164.00	77.0	1.0000		0.7929		0.8893		0.9286	

Table 6: Various values for the different counter groups. The test for a log-normal distribution used is the Shapiro test. The three other tests are for difference in location between the counter and all other counters. Under our assumptions, little stars are to be expected.

the p-values for three different tests, a KS-test, a Wilcoxon Mann-Whitney U-test [6] and a t-test. There is little evidence for rejecting the hypotheses that there are no differences. The most evidence is found for internal counter number 7, but after the Bonferroni correction for 17 counters (multiplication by 17) to its p-value of 0.0022 there is little evidence that something unexpected is happening. In this table, we also show the outcomes of a Shapiro test for normality [30], to test for a log-normal distribution. Internal counter number 11 gives most evidence for rejecting log-normality, but after compensating the p-value of 0.0187 by 17 tests with the Bonferroni correction, it is clear that nothing unexpected is happening.

6 Relating costs and function points

We have now established that there are no differences between the distributions of the number of function points counted by different counters. In further analyses, those function point counts are usually used in different statistical models. One of the interesting models is the relation between costs and function points, or in other words, the productivity.

It is known from research that there is a functional relationship between the size of a system and the costs involved in building the system. Normally, there is a log-log relation between function points and costs [31]. If the counting practices of the individual counters play an important role then these qualitative effects must be incorporated in the estimation process as dummy variables. We statistically tested whether the improvement in fit achieved by including dummy variables in the model is sufficiently large to conclude that the inclusion of the dummy variables is profitable. If not, the parsimonious model without dummy variables must be favored, which indicates differences between counters.

no	model
1	$\log(\mathit{costs}_i) = \alpha + \beta \log(\mathit{FP}_i) + \epsilon_i$
2	$\log(\mathit{costs}_{ei}) = \alpha_e + \beta \log(\mathit{FP}_{ei}) + \epsilon_{ei}$
3	$\log(\mathit{costs}_{ei}) = \alpha + \beta_e \log(\mathit{FP}_{ei}) + \epsilon_{ei}$
4	$\log(\mathit{costs}_{ei}) = \alpha_e + \beta_e \log(\mathit{FP}_{ei}) + \epsilon_{ei}$
5	$\log(\mathit{costs}_{ai}) = \alpha_a + \beta \log(\mathit{FP}_{ai}) + \epsilon_{ai}$
6	$\log(\mathit{costs}_{ai}) = \alpha + \beta_a \log(\mathit{FP}_{ai}) + \epsilon_{ai}$
7	$\log(\mathit{costs}_{ai}) = \alpha_a + \beta_a \log(\mathit{FP}_{ai}) + \epsilon_{ai}$

no	AIC	BIC	RSS	F-test	Shapiro	kurtosis
1	490.14	501.35	86.53		0.0290	0.3752
2	489.40	504.34	85.76	0.0993	0.0071	0.4764
3	490.84	505.79	86.16	0.2562	0.0114	0.4442
4	483.10	501.78	83.50	0.0043	0.0083	0.4726
5	466.46	481.41	79.65	0.0000	0.0088	0.5754
6	470.80	485.75	80.77	0.0000	0.0141	0.5864
7	463.78	482.46	78.45	0.0000	0.0042	0.5111

Table 7: Various models for the relation between costs and function points. The ANOVA F-test was done with regard to the first model. Models 2–4 vary for being counted by an external counter or not. Models 5–7 vary for being after project 149 or not.

Formally, the relation between function points and costs is expressed as

$$\log(\mathit{costs}_i) = \alpha + \beta \log(\mathit{FP}_i) + \epsilon_i$$

with costs_i the cost of project i, FP_i the number of function points of project i, ϵ_i a normally distributed error per project and α and β as model parameters. We tested whether the impact of function point counters should be added to this model. If it needs to be added this shows that there are differences between counters.

For testing we used both AIC (Akaike's Information Criterion) and BIC (Bayesian Information Criterion) as model selection criteria. Both criteria penalize model complexity while in the penalty term of BIC the sample size is included. With small sample sizes AIC favors models with fewer parameters compared to BIC. The latter is more commonly used in sociology, while AIC is very popular in econometrics. Both criteria will usually identify good models for observed data but sometimes fail in this respect, for example by selecting a model with too many or too few parameters. It therefore is advocated to use the two criteria together. When both criteria agree on the best model, this provides reassurance of the choice. Given the underlying assumptions, and simply stated, AIC seeks a model that minimizes the error with the true model, while BIC seeks for the true model, the model on which the data is actually based. A more detailed analysis of AIC and BIC is given in Kuha [21] and Burnham and Anderson [4]. Because we want to find a true model and know if functions point counters are part of it, we will prefer BIC here. Another reason in favor of BIC is the size of our data set. For larger data sets AIC penalizes less and allows more parameters with the danger of overfitting or uninterpretable models.

For all the models we will show the AIC, BIC and residual sum of squares (RSS) [30]. For these three criteria, a lower value indicates a better model, while a higher value indicates a model that is worse. Next to that, we will show the results of a

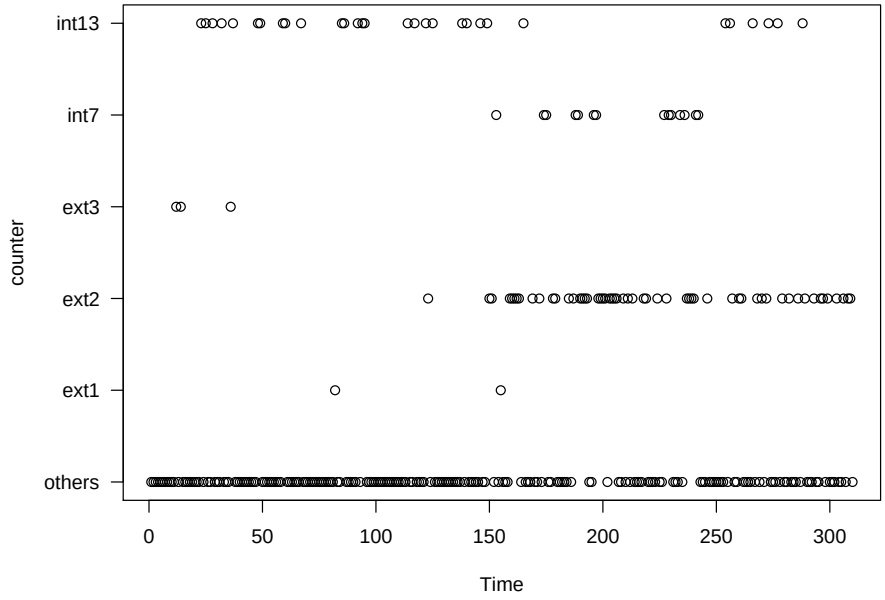

Figure 5: Numbered subsequent projects are counted by different counters.

more formal ANOVA F-test [30] that compares different models and gives a p-value. Because the residuals are not entirely normally distributed, as shown by a Shapiro test for normality, and a kurtosis unequal to 0, the p-values from the F-test are not exact and probably a bit too low. However, the F-test is reasonably robust [11], so the p-values still give an indication.

Adding all 17 function point counters to the model introduces lots of parameters, which does not give us a better model, but instead gives us a model that is over-fitted. We therefore focus on a distinction in two different groups, namely the internal and external counters.

In Table 7 we show the results of introducing different parameters for internal and external counters in the model. In this table, the following notation is used for each model between cost and function points. Variables α_e and β_e take two values, the one used depending on whether the project is counted by externals. Variables α_a and β_a take two values, the one used depending on whether the project is after project 149 or not. The project number is identified by i, and the combinations ei and ai are used to show the aspects contained in the model (ei for externally counted projects being treated different; ai for late projects being treated different). The normally distributed error term is denoted by ϵ.

Model 1 is the model with no impact from the function point counters. Model 2 has different parameters for the value of α, model 3 for the β and model 4 for both parameters. It appears that model 4 is the best model on all criteria, as it has a lower AIC, BIC and RSS, and also a low p-value for an F-test comparing it to model 1.

no	model
5	$\log(costs_{ai}) = \alpha_a + \beta \log(FP_{ai}) + \epsilon_{ai}$
8	$\log(costs_{aei}) = \alpha_a + \alpha_e + \beta \log(FP_{aei}) + \epsilon_{aei}$
9	$\log(costs_{aei}) = \alpha_a + \beta_e \log(FP_{aei}) + \epsilon_{aei}$
10	$\log(costs_{aei}) = \alpha_a + \alpha_e + \beta_e \log(FP_{aei}) + \epsilon_{aei}$
11	$\log(costs_{aei}) = \alpha_{ae} + \beta \log(FP_{aei}) + \epsilon_{aei}$
12	$\log(costs_{aei}) = \alpha_{ae} + \beta_e \log(FP_{aei}) + \epsilon_{aei}$
13	$\log(costs_{aei}) = \alpha_{ae} + \beta_{ae} \log(FP_{aei}) + \epsilon_{aei}$

no	AIC	BIC	RSS	F-test to 5	F-test to 7	Shapiro	kurtosis
5	466.46	481.41	79.65		-0.0317	0.0088	0.5754
8	468.35	487.03	79.62	0.7324		0.0106	0.5625
9	467.71	486.40	79.45	0.3896		0.0136	0.5441
10	462.17	484.59	77.55	0.0169	0.0597	0.0125	0.5618
11	466.91	489.33	78.74	0.1741		0.0121	0.5715
12	460.12	486.27	76.54	0.0070	0.0233	0.0161	0.5577
13	463.28	496.91	76.33	0.0243	0.0809	0.0146	0.5187

Table 8: Various models for the relation between costs and function points. The ANOVA F-test was done with regard to models 5 and 7 (with negative values indicating the first model was assumed to be better).

no	model
5	$\log(costs_{ai}) = \alpha_a + \beta \log(FP_{ai}) + \epsilon_{ai}$
14	$\log(costs_{aei}) = \alpha_a + \beta \log(FP_{ai}) + \epsilon_e + \epsilon_{afi}$
15	$\log(costs_{afi}) = \alpha_a + \beta \log(FP_{ai}) + \epsilon_f + \epsilon_{afi}$

no	AIC	BIC	RSS	F-test	Shapiro	kurtosis
5	480.48	495.39	472.84	0.0088		0.5754
14	482.48	501.12	472.84	0.9997	0.0088	0.5754
15	481.09	499.72	471.09	0.2378	0.0458	0.5002

Table 9: Models testing for random effects. Note that the base AIC and BIC of model 5 in this implementation is different, and the RSS is estimated as -2 times the log-likelihood.

At first sight, this would indicate that external counters count different from internal counters. That would be surprising given the outcomes of the tests in the previous sections.

An explanation is given by taking into account that our data set was gathered during a time of software process improvement, during which the relation between cost and duration may have changed. In Figure 5 we show the position in time of the projects that were counted for the three external counters, and two internal counters. It appears that external counter number 2, who counted most projects of all external counters, mainly counted projects after project 150. Other counters, such as internal counter 1 mainly counted projects during the beginning of our data set. Clearly, there is a relationship between project time and function point counter.

To address this property of the data, we distinguish between projects after project 149, and the ones before. We will introduce different parameters as with the internal

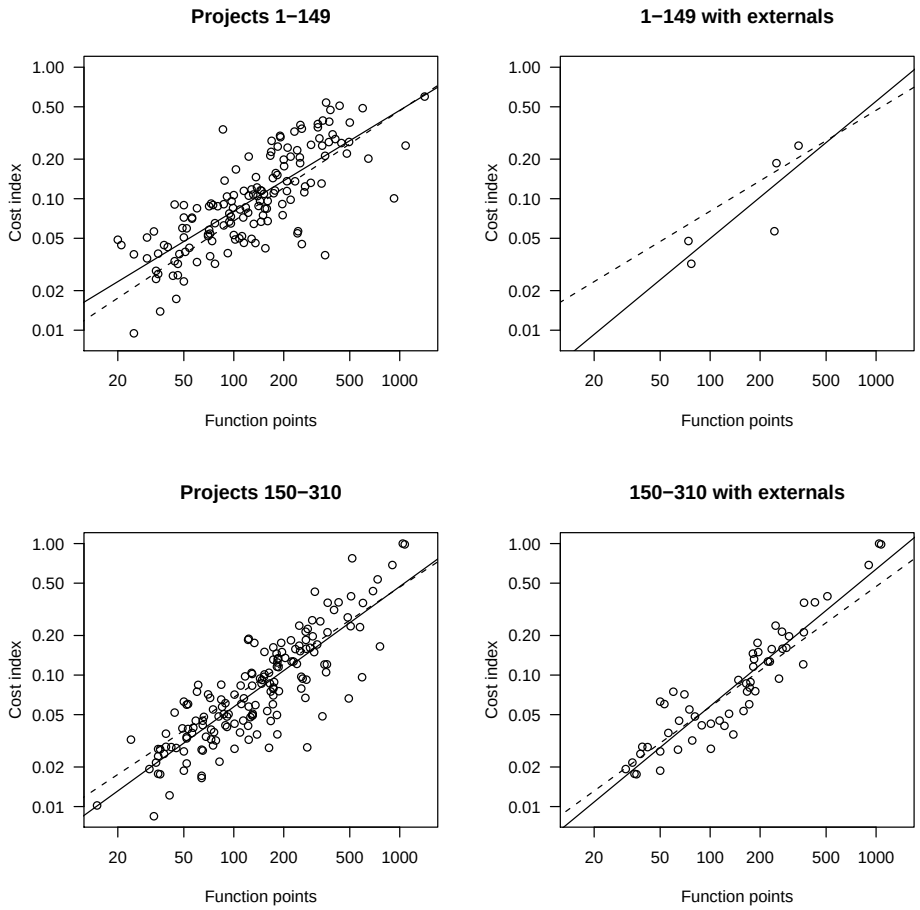

Figure 6: Visualization of the linear model for different groups. The dotted line shows the fitted line without the split taking place. For the plots on the left it is for the model without groups; for the plots on the right, it is the same as the solid line on the left of the plot.

and external counters, but now for early and late projects. In Table 7, these are shown as models 5 to 7. Now it appears that in terms of BIC, model 5 is the best model. This model has different values for the β model parameter for early and late projects. Model 7 is better in terms of AIC, and would be also a good choice, but as we are seeking a true model, we prefer model 5 for now.

After we have added the difference between early and late projects to our model, giving model number 5, we investigated the possibility of an even better model by taking potential differences between external and internal counters into account. For the models shown in Table 8 variables α_e and β_e are either zero or have a value, depending on whether the project is externally counted. Variables α_{ae} and β_{ae} denote four different parameters for the four situations (project among first 149 true/false and project counted by externals true/false). None of the models shown in Table 8 scores better on

our preferred BIC criterion. Some models do score better on the other criteria, notably model number 12. However, the p-values from the formal F-test are not very convincing, given that the test is not completely valid and we tested lots of models. So we conclude that model 5 is the best choice.

In Figure 6 we compared the different models visually. The earlier projects are shown in the upper-left corner, and the late ones in lower-left corner. The solid line is model number 1, which does not take time into account. The dotted lines are from model number 7, where time is taken into account. Visually, the fit of model number 7 is a bit better, but not much. On the right-hand side of Figure 6, only the projects counted by external counters are shown. On this side, the solid line is model number 7, and the dotted line is from model number 13, which has different parameters for the projects counted by externals and for early and late projects. While the lines do differ, given the number of projects and the non-universal difference (the line is not both lower and higher than model 7), it visually does not appear to be a better model than model 7. Therefore, also after visual comparison, there are no signs of differences between internal and external counters.

Instead of treating the impact of function point counters as fixed parameters, we also investigated modeling them as giving random systematic differences, which form a normal distribution. In this way, if we add all function point counters to the model, we only need to penalize for one extra parameter. Such models are called random-effect models [30]. In Table 9, we show the outcomes of adding random, systematic differences between counters to the model. In these models, variable ϵ_e denotes errors based on external/internal counters, and ϵ_f denotes systematic errors per counter. In model 14, we add systematic errors between internal and external counters, while in model 15 we add systematic errors between function point counters. The models score in terms of AIC and BIC worse than our current model number 5. Therefore, the random effect models do not appear to be better than model number 5. So, there are no signs of systematic, randomly distributed differences between counters.

In conclusion, the most likely true model we created is model 5. In this model, time is taken into account, but function point counters are not. We therefore conclude that there is no evidence for an impact of function point counters on the relation between cost and function points.

7 Discussion

In an earlier analysis we carried out on a more limited data set, we originally concluded that there were in fact differences between internal and external counters. Indeed, it seemed that internal function point analysts counted more function points than external function points analysts, thereby seemingly boosting productivity. This was exactly a problem we were looking for, and some suspicious high counts from internal counters, and low counts by external counters were inspected. After we had more data it turned out that our early conclusions were caused by either a statistical anomaly, or, more likely, that the external counters had not finished counting some of the larger projects.

Note that our research methodology is not a preferred design for researchers in general. In fact, there has been no adaption of the design towards the research at all; the data was just gathered in a real-world situation. For example, normally, one would prefer to have a more or less equal number of measurements per counter and per counter group. Next to that, one would prefer to have recounts of the same situation, instead of studying data gathered on different projects. Finally, one would prefer not to have time

#	test	statistic(s)	outcomes
1	Differences in distribution between pairs of individual counters	KS-test	Initially, 128 out of 136 tests (94%) showed no differences, after 1 Bonferroni correction 0 tests show statistical evidence.
2	Differences in location between pairs of individual counters (surpassed by #4)	t-test	Initially, 120 out of 136 tests (88%) showed no differences, after Bonferroni correction 1 test showed little statistical evidence.
3	Differences in location between all counters	Kruskal-Wallis rank sum test	no evidence ($p = 0.3460$)
4	Differences in location between all counters (assuming log-normal distributions)	ANOVA F-test	no evidence ($p = 0.2092$)
5	Differences in distribution between internal and external counters	KS-test	no evidence ($p = 0.449$)
6	Differences in location between internal and external counters	t-test	no evidence ($p = 0.5647$), 95% confidence interval between 27% lower for internal counters to 19% higher for internal counters
7	Differences in distribution between counter and all other counters	KS-test	Initially, 15 out of 17 tests (88%) showed no differences, after Bonferroni correction no test showed differences.
8	Differences in location between counter and all other counters (surpassed by #3 and #4)	U-test and t-test	Initially, 29 out of 34 tests (85%) showed no differences, after Bonferroni correction for 17 tests, only one test showed difference.
9	Log-normality of function point distribution per counter	Shapiro-Wilk test	Initially, 15 out of 17 tests (88%) showed no differences, after Bonferroni correction no test showed differences.
10	Function point counter and/or counter group is not included best cost-size model	AIC, BIC, RSS, F-test	After adjusting for time, function point counter behavior was not a factor that was included in the best model.

Table 10: A summary of important statistical tests that were performed.

effects that have to be compensated for.

It is a fact of life that data is imperfect. Using sometimes sophisticated means to compensate for that is in our view a powerful tool to bring empirical software engineering a step further. It is often not possible to construct experiments so that statistical analyses become standard or trivial. Instead when we gather data and reveal their imperfections we should account for them and filter out patterns that help in understanding software engineering in a quantified manner. In conclusion, we showed that also in a setting where recounts are out of the question, it is still possible to assess the reliability of function point counts.

8 Conclusions

We showed that in the literature, differences between counts of the same project by different function point counters give results that can differ more than 30%. For most statistical purposes, however, this is not a problem if the errors are not systematic, so that they should be compensated for during the analysis. Classically, differences between counters were measured using recounts. We proposed a method to test for systematic errors, without requiring any recounts. We used this method on a portfolio

of 311 projects and 58143 function points in total.

In our case study, the function point counting practice turned out to be state-of-the-art when it comes to inter-rater reliability: very little statistical evidence was found that there were differences between counters or groups of counters. Given the abundance of various tests we have presented, as summarized in Table 10, the sparse evidence that was found is not unexpected, as doing many statistical tests will generally give some false positives. For the difference between internal versus external counters, we showed that if a difference would exist, it is likely too small to influence analyses based on the data. We conclude that there is no real statistical evidence that there were systematic errors being made by the function point counters.

Acknowledgments This research has partially been sponsored by the Dutch Ministry of Economic Affairs via contract SENTER-TSIT3018 *CALCE: Computer-aided Life Cycle Enabling of Software Assets*. Furthermore, this research received partial support by the Dutch *Joint Academic and Commercial Quality Research & Development (Jacquard)* program on Software Engineering Research via contract 638.004.405 *EQUITY: Exploring Quantifiable Information Technology Yields* and contract 638.003.611 *Symbiosis: SYnergy of Managing Business-IT-alignment, IT-sourcing and Offshoring Success In Society*.

References

[1] A. J. Albrecht and J. E. Gaffney. Software function, source lines of code, and development effort prediction: A software science validation. *IEEE Trans. Softw. Eng.*, 9(6):639–648, 1983.

[2] A.J. Albrecht. Measuring application development productivity. In *Proceedings of the Joint SHARE/GUIDE/IBM ApplicationDevelopment Symposium*, pages 83–92, 1979.

[3] C. Alan Boneau. The effects of violations of assumptions underlying the t test. *Psychological bulletin*, 57(1):49–64, 1960.

[4] K.P. Burnham and D.R. Anderson. Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2):261–304, 2004.

[5] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences (2nd Edition)*. Lawrence Erlbaum, January 1988.

[6] W.J. Conover. *Practical Nonparametric Statistics*. Probability and Mathematical Statistics. John Wiley & sons, 3rd edition, 1980.

[7] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[8] J.B. Dreger. *Function Point Analysis*. Prentice Hall, 1989.

[9] Andy Field. *Discovering Statistics Using SPSS*. SAGE Publications, 2005.

[10] D. Garmus and D. Herron. *Function Point Analysis – Measurement Practices for Successful Software Projects*. Addison-Wesley, 2001.

[11] Gene V. Glass, Percy D. Peckham, and James R. Sanders. Consequences of failure to meet assumptions underlying the analysis of variance and covariance. *Review of Educational Research*, 42(3):237–288, 1972.

[12] D.R. Jeffery, G.C. Low, and M. Barnes. A comparison of function point counting techniques. *IEEE Trans. Softw. Eng.*, 19(5):529–532, 1993.

[13] C. Jones. *Applied Software Measurement: Assuring Productivity and Quality*. McGraw-Hill, second edition, 1996.

[14] C. Jones. *Estimating Software Costs*. McGraw-Hill, 1998.

[15] Capers Jones. Software metrics: Good, bad and missing. *Computer*, 27(9):98–100, 1994.

[16] Peter Kampstra. Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software, Code Snippets*, 28(1):1–9, 10 2008.

[17] C.F. Kemerer. Reliability of function points measurement – a field experiment. *Communications of the ACM*, 36(2):85–97, 1993.

[18] C.F. Kemerer and B.S. Porter. Improving the reliability of function point measurement: An empirical study. *IEEE Transactions on Software Engineering*, SE-18(11):1011–1024, 1992.

[19] Barbara Kitchenham. Counterpoint: The problem with function points. *IEEE Software*, 14(2):29,31, 1997.

[20] A.N. Kolmogorov. Sulla Determinazione Empirica di una Legge di Distribuzione. *Giornale dell' Istituto Italiano degli Attuari*, 4:83–91, 1933.

[21] J. Kuha. AIC and BIC: Comparisons of assumptions and performance. *Sociological methods & research*, 33(2):188–229, 2004.

[22] G.P. Kulk and C. Verhoef. Quantifying requirements volatility effects. *Sci. Comput. Program.*, 72(3):136–175, 2008.

[23] G.C. Low and D.R. Jeffery. Function points in the estimation and evaluation of the software process. *IEEE Transactions on Software Engineering*, 16(1):64–71, 1990.

[24] F. Mosteller and J.W. Tukey. *Data Reduction and Regression*. Addison-Wesley, 1977.

[25] Thomas V. Perneger. What's wrong with Bonferroni adjustments. *British Medical Journal*, 316:1236–1238, 1998.

[26] M.A. Rispens and F.W. Vogelezang. Application portfolio management, the basics – How much software do I have. In *Proceedings of the 4th Software Measurement European Forum (SMEF 2007)*, Rome, Italy, may 2007.

[27] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[28] N.V. Smirnov. Sur les écarts de la courbe de distribution empirique. *Matematicheskij Sbornik. (Novaya Seriya)*, 6:3–26, 1939. Russian/French summary.

[29] J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[30] W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S*. Springer Verlag, 4th edition, 2002.

[31] C. Verhoef. Quantitative IT portfolio management. *Science of Computer Programming*, 45(1):1–96, 2002. Available via: www.cs.vu.nl/~x/ipm/ipm.pdf.

[32] E.J. Wegman. Nonparametric probability density estimation. *Technometrics*, 14:533–546, 1972.