Probabilistic Models of Software Function Point Elements

Masood Uzzafer
Amity university – Dubai
Dubai, U.A.E.
Email: muzzafer [AT] amityuniversity.ae

Abstract— **Probabilistic models of software function point elements are presented. Probabilistic models help to understand the random behavior of the function point elements under consideration; these models facilities use of function point elements to be used under different analysis techniques by using different probabilistic confidence bounds and expected values of the function point elements. Software function points are widely used in the software development industry. Analysis of function point elements using new modeling and analysis techniques has been a key research area for the researchers in the software industry.**

Keywords: Function Point, Software Projects, Software Size, Probability Models.

I. INTRODUCTION

Function Point elements have attracted much attention in the software research and development industry. Ever since function point were introduced by IBM in 70's their nature, behavior, impact, and correlation have been studied by the software researchers and the software developers. The idea behind function points is to standardize the measurement of the various software functions to estimate the software development effort which is independent of the computer language, development methodology, technology and the capabilities of the software development team. The International Function Point Users Group (IFPUG) was founded in the late eighties and is a membership governed, non-profit organization committed to promoting and supporting function point analysis and other software measurement techniques. There have been various releases of the Function Point by the International Function Point Users Group (IFPUG) with the latest 'Counting Practice Manual – 4.2' release.

Software projects developed using different software development tools, languages and different software development platforms experience different counts of function point elements. Probabilistic models of function point elements can help to model the function counts irrespective of the software development tools, and language and platform used. A probabilistic model is built by estimating the expected function points counts than constructing a probabilistic model using that expected function count.

To build the probabilistic models, function points counts is collected from different completed software projects and fitted with known probability models, the probability distribution fit is tested with the chi-square goodness of fit analysis.

The paper is organized in the following way: section 2 describes the data set used for the analysis section 3 focuses on the correlation between function point elements section 4 gives function point correlation with the software defects section 5 explains probability distributions of function elements section 6 focuses on the principle component analysis section 7 describes the multi-linear regression analysis and section 8 draws some conclusions.

II. FUNCTION POINTS DESCRIPTION

Function point describes the size of the software using five elements: Internal Logical Files (ILF), External Interface Files (EIF), External Inputs (EI), External Outputs (EO) and external Enquiries (EQ), function point calculation begins with counting these five elements. Each function point element is assigned a complexity level (Low, Average, High) based on its associated file number such as Data Element Type (DET), File Type Referenced (FTR) and Record Element Types (RET). The complexity metrics for five elements is shown in Table 1. Each function component is then assigned a weight according to it complexity shown in Table 2. Unadjusted Function Point (UFP) is the total number function points counted together and represents the size of the project. The unadjusted function point is computed from the following equation.

$$\text{UFP} = \sum_{i=i}^{5}\sum_{j=1}^{3} w_{ij}x_{ij} \qquad (1)$$

Where w_{ij} is the complexity weight and x_{ij} is the count for each function element. UFP is then multiplied by the Value Adjustment Factor (VAF) to get the function point (FP) count. The VAF is calculated from 14 General System Characteristics (GSC) using equation 2. These characteristics are 1) Data Communication 2) Distributed Functions 3) Performance 4) heavily used configuration 5) transaction rate 6) on-line data entry 7) end user efficiency 8) on-line update 9) complex processing 10) reusability 11) installation ease 12) operational ease 13) multiple sites and 14) facilities change. These values are summed and modified to calculate the VAF.

$$\text{VAF} = 0.65 + 0.01\sum_{i=1}^{14} c_i \qquad (2)$$

Where c_i are the GSC value. Finally the UFP and VAF are multiplied to the function point count.

$$\text{FP} = \text{UFP} \times \text{VAF} \qquad (3)$$

Table 1: Function Point element complexity metrics

ILF/EIF	DET		
RET	1-19	20-50	51+
1	Low	Low	Avg
2-5	Low	Avg	High
6+	Avg	High	High
EI	DET		
FTR	1-4	5-15	16+
0-1	Low	Low	Avg
2	Low	Avg	High
3+	Avg	High	High
EO/EQ	DET		
FTR	1-5	6-19	20
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

Table 2: Function Point complexity weights

Component	**Low**	**Average**	**High**
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

III. Understanding the Data Set

The data set for analysis is taken from the International Software Benchmarking Standards (ISBSG) repository [4]. ISBSG performs the data validation of the contributed data to make sure the data quality and consistency. The obtained repository contains data from 3024 different projects, where almost all the projects used IFPUG standard [5] for function points. Projects which used other methods then IFPUG were excluded from the study. Data sets with the missing function points and defects count values were also excluded.

In the selected projects largest projects were contributed from the financial industry (banking, financial services, and accounting) the rest of the projects were form engineering (software, hardware and telecommunication), insurance, public administration, government, manufacturing, consulting and education. The collected data set is not homogenous which would ensure linearity in statistical analysis. The variety in the data set ensures that the data sample represents different scenarios and possibilities in the software development industry.

Unadjusted function points represents the size of the projects, Figure 1 shows the histogram of unadjusted function points. The minimum project size is 13; the largest is 4943; mean is 579.33 and standard deviation 715.46. Majority of the projects size are in the range of 13 to 500 unadjusted function points, while there are few projects of size more than 2000.

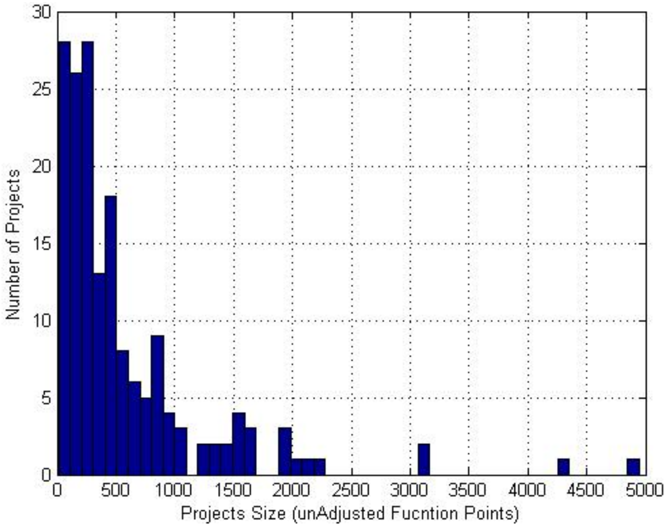

Figure 1: Projects Size (unadjusted FP)

Box plots helps to understand the measure of central tendency and dispersion. The box plot of function point elements is drawn in Figure 2. All the function point data elements which were 3 times the standard deviation away from the sample mean were classified as the outliers and removed. The line in the middle of the box represents the median if the line is not in the center of the box that is an indication of the skewness. Skewness is a measure of asymmetry of the data around the sample mean/median. The lower and upper lines of the box are 25th and 75th percentiles respectively. The distance between the upper and lower lines is the interquartile range. Whiskers, lines extending above and below the box, show the rest of the data. The length of the whiskers is set to 1.5 times the interquartile range. Plus sign shows the data point which the 1.5 times away from the interquartile range. Table 1 gives the median and percentile values of the function point elements.

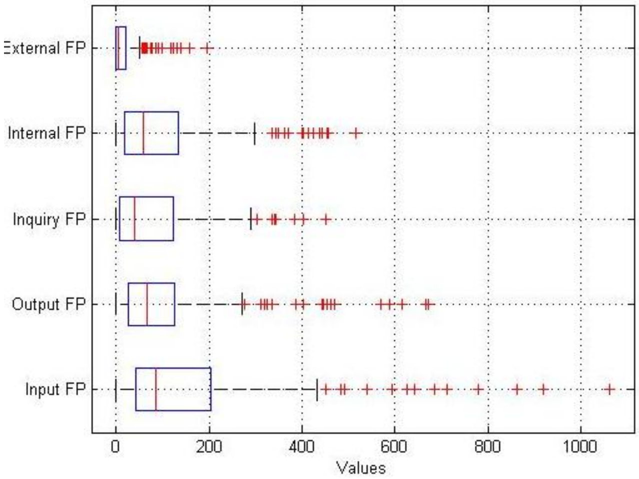

Figure 2: Box Plot of Function Point elements

Table 3: Median and percentiles of the function point elements

	Median	25th percentile	75th percentile	100th percentile
Input	86.5	41.5	204.5	1061
Output	67	26	125	673
Inquiry	39	7.25	122.5	450
Internal	58	17.75	133	516
External	5	0	20	195

It is obvious from the box plot that the input function point element has the longest tail above the upper whisker and the data values are more widely spread over the upper whisker then in any other function point element. The median line is not in the middle of the box representing having a large area after it in the box this represents the positive skewness meaning that the data values are more spread out after then median. This phenomenon is also observed with other function point elements and found to be common in all function point elements. The external function point has the smallest set of values with fairly small upper and none existing lower whisker.

IV. Probabilistic Models of Function Points

Probability distribution function assigns probability to a certain event and it helps to understand the chances to observe an event with a relevant probabilistic bound. Probabilistic model are not prediction models rather it determines the probability that a given event will occur inside a specific range of values. Probability models are more appropriate for function point modelling as appose to deterministic models. Probability distribution of function point elements will reveal information about what value of function points are most likely to be observed independent of the software developmental tools languages and platform. Different probability distribution models were tested the best probability model which found to be the best fit for function point elements was exponential distribution. It was observed with interest that all function point elements follow exponential distribution.

All the function point elements found to follow the exponential distribution, Figure 3 illustrate the distribution fitting for all the function point elements data.

Exponential distribution has the following form:

$$f\left(x\right)=\frac{1}{\mu}e^{-\frac{x}{\mu}} \tag{4}$$

Where μ is the mean of the distribution. Maximum likelihood estimation (MLE) analysis was used for parameter estimation of exponential distribution that fits the given data that gives the maximum likelihood for the given function points count data. First maximum likelihood of (mean μ) was calculated then this value of mean is used to estimate the exponential distribution as shown in figure 3. Figure 4 shows estimated and observed values of cumulative probability distribution of each function point element.

V. Chi-Square goodness of fit for exponential distribution fit

One fundamental issue in the probability and statistical analysis is whether an observed data fits a given distribution. The observed data would not fit the distribution exactly and some goodness of fit criteria is required. Chi-Square distribution provides such criteria. Let's assume is the null hypothesis that the observed data fits a given distribution. If denotes the observed data frequency and denotes the expected frequency obtained from the given distribution, then the chi-square value or chi-square statistic measures the weighted squares of the difference between observed and expected frequencies of the data:

$$q=\sum_{i=1}^{m}\frac{\left(O-E\right)^2}{E} \tag{5}$$

The equation 5 is known is Pearson test statistic. If there is large number of observations then q follows the distribution. This leads to the conclusion that the null hypothesis is accepted if the value of q is less then where m-1 is the degree of freedom, the value of m is usually the total number of observations. The value is determined by pre-assigning a significance level where:

$$\alpha = P\left\{\chi^2\left(m-1\right)\geq q\right\} \tag{6}$$

Frequently used values of are 0.1, 0.05, 0.01 and 0.005. Chi-Square statistic defined in equation (5) is widely used to find the goodness of fit of an observed data set for a given probability distribution.

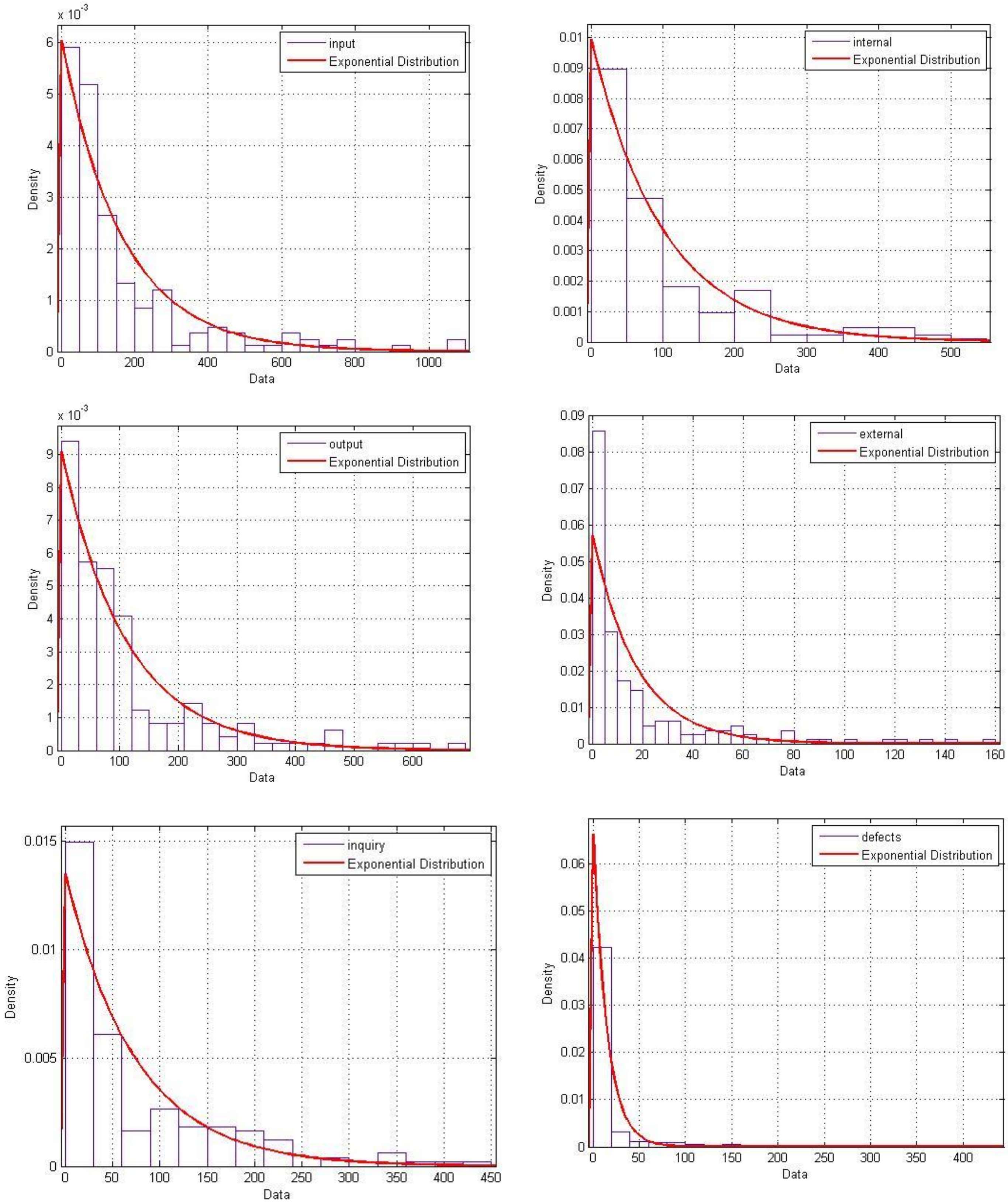

Figure 3: Function Point Distributions

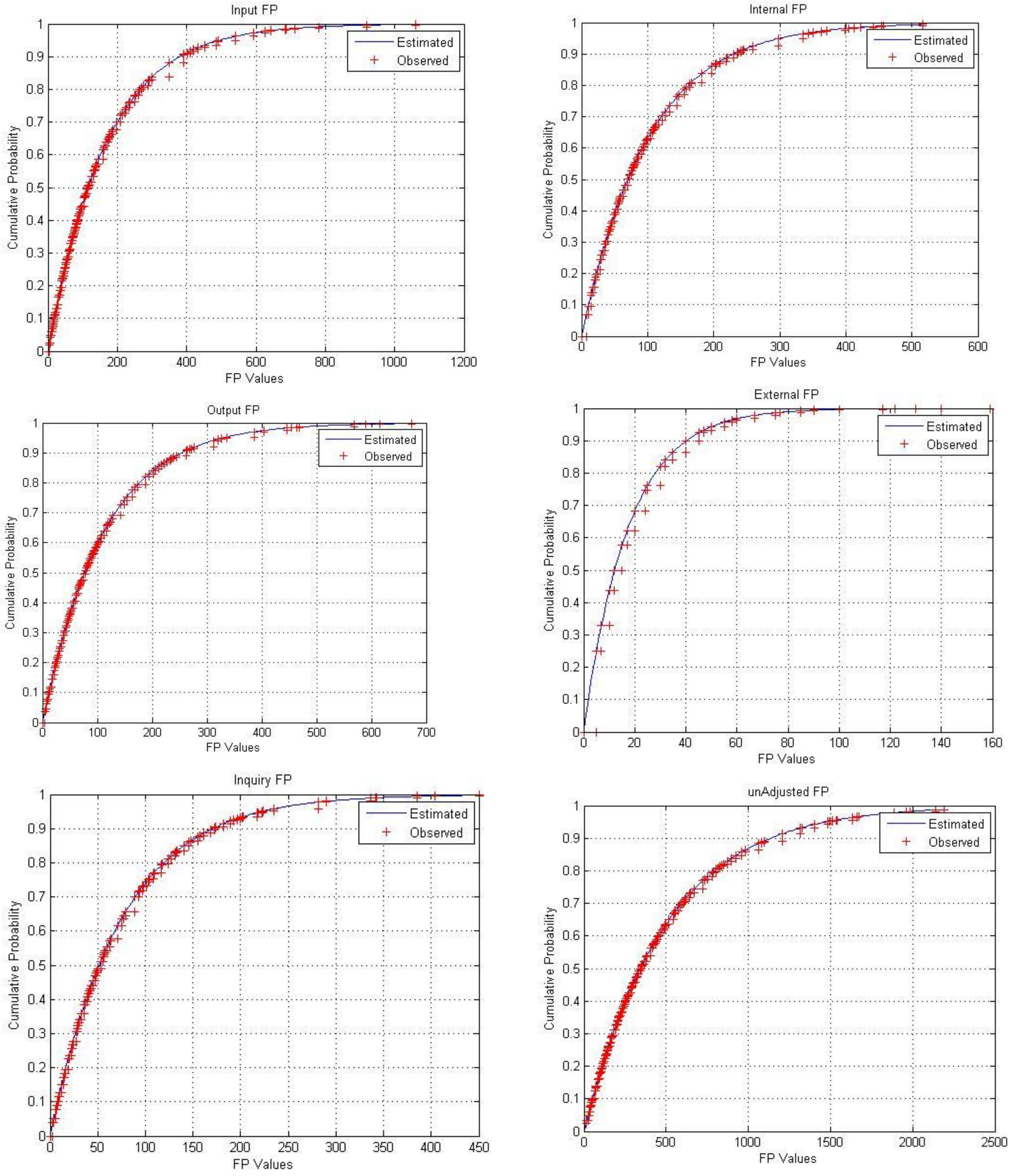

Figure 4: Observed and Estimated Cumulative Probability Function

First the expected frequency occurrences of the data are calculated then these values are plugged-in the equation (5) together with the observed frequencies that produces a chi-square value. If the chi-square value is less than the chi-square distribution $\chi^2(m-1)$ value with some degree of confidence then the null hypothesis is accepted. Table (4) shows the calculated chi-square values for function point elements also the chi-square distribution $\chi^2(m-1)$ value for α =0.005 is shown. All the calculated chi-square values for function point elements are much less than the chi-square distribution $\chi^2(m-1)$ value for α =0.005. It means that there is that there is 99.5% confidence that the data follows the exponential distribution or in other words the data set is taken from the set which is exponentially distributed.

Table 4: Critical and Calculated Chi-Square Statistics values

	$\chi^2(m-1)$=215.54 (Degree of freedom= 165) α =0.005				
Input	Output	Inquiry	Internal	External	Un-Adjusted
2.90	12.87	5.63	11.02	0	5.03

VI. Conclusions

Probabilistic models for function point elements counts are presented. Function point count data is fitted with the different known probability models and shown that the function point element follows exponential distribution. Chi-square goodness of fit test for exponential distribution is performed and shown that exponential distribution fits the function point data set taken from IFPUG. Given an expected number of count, these function point probabilistic models provides and understanding of the range and spread of possible values of function point elements regardless of the software development tools, language and developmental platform used for the development of the software project.

References

[1] D.R. Jeffery and J. Stathis, Function Point Sizing: Structures, validity and applications. Journal of Empirical Software Engineering, 11-30, 1996.

[2] B. Kitchenham and K. Kansala, Intr-item correlations among function points. In Proc. 15th International Conference on Software Engineering, IEEE, pages 477-480, May 1993.

[3] International Software Benchmarking Standards Group. Release 9.

[4] IFPUG. Function Point Counting Practices Manual release 4.2.