Feature Article

Function Points Leveraging Transparency and Enabling Control Over Software Procurement

By Carlos Eduardo Vazquez, CFPS, FATTO Consultoria

Methodology to Suggest Objects of Interest for Procurement Auditing Purposes

Introduction

The tragic scandal from Enron Corporation bankruptcy triggered in motion the events that culminated with the Sarbanes-Oxley Act. Now for over a decade, higher levels of transparency and chargeability are qualities pursued for every business, government and non-profit organization.

The C levels of Management have their attention drawn to corporate governance due to increased priorities derived from those goals. The goals set for overall business operations ripples its way to the IT organization. Outsourcing contracts and project procurement management play a major role as areas of interest to assure compliance to internal controls and auditing practices.

Management establishes software supplier agreements terms and conditions in different ways. Those agreements span over a series of issues, the two most relevant for the theme under discussion are:

• Pricing and compensation methodology that enables calculation of charges for the services provided to the acquirer.
• Pricing and compensation schedules that provide for charges for the products and services provided, including frequency, term, and pricing type (e.g., fixed price, lump sum, time and materials) as well as rate cards, and a skills matrix.

Regardless of the pricing type chosen, function points play a pivotal role when it comes to audit and control over software procurement.

Motivation

That is so because some half millennia ago Luca Pacioli introduced one of the most revolutionary measures towards control goals above mentioned: the double entry bookkeeping. This concept comes in handy when it comes to software procurement and software production, planning & control.

Function points accounts for assets as results from an investment of time and money. Without the “double entry” perspective, things get confused. Furthermore, function points measure the assets in a management understandable way since its foundation is the user view.

Whenever, there is an analysis based only on costs or investments, scenarios as the one described in Figure 01 arise. It is a meme (or idea) I have recently found when browsing my Facebook timeline. Read it carefully:

You’ve found a shirt for US$ 97.00.
But, you have no money, so you’ve borrowed US$ 50.00 from your mother and US$ 50.00 from your father. So:
US$ 50.00 + US$ 50.00 = US$ 100.00
You’ve bought the shirt, and there is change of US$ 3.00.
You pay US$ 1.00 back to your father, US$ 1.00 to your mother and keep the other US$ 1.00 to you.
Now you owe US$ 49.00 to you mother and US$ 49.00 to your father.
US$ 49.00 + US$ 49.00 = US$ 98.00 + your US$ 1.00 = US$ 99.00
What about the other US$ 1.00?

Figure 01

At the end of this article, there is the solution explaining the right approach to the meme presented.

Surprising as it may seem, the presence of this kind of mathematics is more frequent than you might imagine in the context of software development and maintenance contracts.

Money (as well as time) without a clear function attached to its use leads to confusion.

Scenarios as the above depicted conspire against transparency and that is so whenever there is not a clear product unit in place and there is no strong bound binding investments of time and/or money to deliveries measured in those units.

In those scenarios, if the CEO asks the CIO (really looking for a clear understanding of his/her answer) about how software production planning and control takes place or how the software development and maintenance contract agreements are drawn, then they would realize a universe not so far from the meme depicted previously.

If the only issue under discussion is money or time invested in a software project or operation without something to play a role as comparable product unit, then there is no way to associate a meaning to those numbers. As a result, you cannot assign importance to the information. Finally, you cannot make an informed decision or compare projected (or accomplished results) with prior results.

For instance in 2009, a law firm (working for a software contractor who had acted as a part in a contract where the other part had been a government agency) hired my company. They were seeking help to support their efforts to elaborate the defense thesis for a local Court of Accounts process in course.

The contract established function points as the only pricing and compensation methodology. My company has been keeping records about public software agreements like the one our

(continued on next page)
clients’ client had been engaged for over a decade. Therefore, we were able to certify some measurements and to point out flaws. Those flaws not necessarily were in the measurement but in the agreements’ terms and conditions themselves, so explaining some deeds otherwise easily perceive as acts of bad faith.

If there had been no function points as a product unit, had the contract pricing and compensation model defined for each demand an a posteriori (knowledge or justification is independent of experience) negotiated bunch of hours with no product unit to quantify the deliverables, then the discussion would rest only in expert opinion without whatever benchmark available regarding the process or its performance.

The Function Point Role in Transparency

The major merit of Function Point Analysis (FPA) is to introduce a management understandable, quantifiable and comparable, enterprise and market wide, product dimension in a world where there still prevails the perspective of investment and cost with no standard measure for the assets delivered. Function Points Analysis plays a role doing so and, therefore, establishes the means to plan and evaluate productivity. Without some metric like function points, there is no governance, no management: there is bargaining at most in software procurement.

Another merit of FPA is to identify deviations from a common behavior in order to point out objects of interest for auditing purposes. Sometimes, those exceptions do not necessarily correspond to actual deviations; sometimes, actual deviations do not show as an exception to a common behavior. However, there is a way (even though not the only way) to define a standard operational procedure to select contracts for a more detailed analysis.

Suppose an analysis of prior contracts reveals productivity (expressed by their delivery rates) distributed as depicted in Figure 02. According to the analysis of those 37 contracts, there is an 80% chance of the delivery rate to be about 09 Staff-Hours per Function points (SH/FP) or less.

Of course, FPA does not measure a series of relevant dimensions about the software process with impact on the productivity. That is why the selection presented includes only software development contracts for the Oracle platform. There is no contract addressing enhancement projects nor other platforms comprising different variables not measured by FPA that would cause unnecessary bias in the productivity data.

The information depicted by the analysis of productivity distribution enables software managers or client organizations to define criteria to select objects of interest in the future for auditing purposes.

Even if we focus in the present and consider those past contracts as the focus of our attention, the plot points out as outliers 04 cases, possibly items subject to further inspection.

Pricing Prescribed from Function Points

The procedures so far discussed, are valid in scenarios where the pricing and compensation methodology have as core measures time & material or a lump sum amount. Scenarios where the contract terms and conditions define pricing and

(continued on next page)
compensation methodology using function points as core measure require another procedure to identify objects of interest. Agreements like those have the functional size of the project measured in function points as the primary cost factor to prescribe how much the value of a contract or demand within an umbrella contract are. Since productivity is constant, there is no point for deviations from the prescribed productivity.

We can look for deviations in size for instance. If we plot the distributions of the contracts or individual demands size, then we can identify a pattern and look for exceptions for further evaluation. Figure 04 depicts the distribution of functional size of the very same dataset used so far.

According to my experience, distributions like the one in Figure 04 are common when it comes to software projects contracts. Since the range is too wide (ranging from 68 to 4,272 function points), a good practice is to split the data set into two. First, because those projects with over 1,000 FP are naturally objects of interest for auditing purposes due to its cost. Second, it is easier to analyze the data distribution with a shorter tail. Figure 05, depicts the same data considering only cases up to 2,000 FP.

Suppose the distribution above is set as criteria of comparison over a one-year period. The next year, the audit team will compare the actual distribution against the one set as criteria (in the same fashion the density distribution graph within Figure 05 compares the empirical distribution function derived from the data to the normal distribution function).

Cases within the interval bin with greater variations may have a higher priority for objects of interest selection for audit purposes.

Another strategy, complementary to the one using deviations from size distributions, is the one using staff-hours negotiated or amount paid as criteria to compare distributions. Figure 06 presents the same data set used in Figure 05 (both excludes cases with over 2,000 FP).
When we analyze the data, we realize 50% of the overall cases are up to 2,817 hours. Therefore, instead of using the size as criteria to define the cut off between two classes of projects to analyze, the number of staff-hours might be a better choice. The cut off criteria definition is in practice comprised of a series of iterations until the distribution best suited for the business needs rise.

**Conclusion**

When each contract or demand is negotiated in terms of an amount of hours in a context where the hourly fees are defined in a corporate agreement, management may be under the impression financial decisions are made by people with right authority to make them. However, it is often the case the sizing in staff-hours is just another currency, such as Euro, Pound or Brazilian Real and its exchange value is the hourly fee determined in the corporate agreement.

Function points allow establishing corporate productivity ranges and enabling standard market benchmarking more easily segregating those with responsibility to settle strategic and tactical corporate or departmental agreements from those responsible for technical decisions.

Even if your business IT organization does not use function point as a support tool for its procurement process, it is a great value for auditing purposes.

**The Solution to the Puzzle**

Our goal with this example was to establish the relationship between expenses and incomes (money, time) and its function (assets and liabilities).

The first step, in Figure 07, is the borrowing that creates a US$ 100.00 liability and, at the same time, an asset of the same amount of cash on hand.

(Nota: replace 'RS' with 'US' in all figures.)

There is balance and harmony! There is US$ 100.00 on one hand and, on the other, US$ 100.00. The second step, in Figure 08, is buying the shirt for US$ 97.00.

![Figure 08](image)

There is still US$ 100.00 on both sides. The next step, in Figure 09, is returning US$ 1.00 for each parent.

![Figure 09](image)

The balance of assets and liabilities end up with US$ 98.00. There is US$ 1.00 available for whatever use and a shirt US$ 97.00 worth. As liabilities, there is a balance totaling US$ 98.00. The balance is possible due to expenses and incomes allocation to its function... but if we forget this perspective there is madness.

Billions of dollars are spend every year by organizations worldwide in software development and maintenance without a standard product unit. People without the responsibility or authority by financial decisions bargain services on “too many hours” or “too few hours” basis.

Can you see enterprise governance without accounting? I cannot see software development and maintenance without a product unit measurement enabling software planning and control production.