

Software Contracting and Management Using Function Points

by *Guilherme Siqueira Simões, CFPS*

Introduction

How can function points help improve both contracting and management of software projects? Some answers to this question will be presented in this paper based on my experience in Brazil, the country with the most function point users in the world.

First of all, we are going to talk about the outsourcing phenomena of software services development. Then, I will address the most common models for contracting this service. I will then present how to run a cost model using function points as a metric for software contracting. Finally, both disadvantages and advantages in the adoption of this model will be discussed alongside the possible benefits.

Software Development Outsourcing

The outsourcing of IT services has increased since 1990, including software development services. Before that, software development and maintenance were executed mainly by the company's internal teams, whose members were mostly systems analysts, programmers, and software developers.

Currently, many companies seek to keep IT team members focused on their core business. Therefore, they merged different IT job positions into just one: business analysts. This job was assigned to join both IT and business objectives while acting as a bridge between IT companies and IT services providers.

Common Models for Contracting

Nowadays, many of the software development projects are executed externally. The most common models for software services contracting are:

1. Man-Hour, also Known as “Body Shopping” or “Time and Material”

In this model, software development services, which are not always projects, are executed by professionals outsourced by the client. The payment for these services is based on the talent and effort of the professionals allocated on the contract.

In theory, this is a contract model of easy management by the client. This approach provides flexibility in response to changes during the project. Moreover, contractual renegotiations will not be necessary under this contract model.

However, the “agility” within this contract model can be illusory. Changes in requirements often are uncontrolled and require additional work which almost always is not visible to the client.

In addition, the software provider's payment is not related to the results obtained. In fact, it is a model that promotes the antithesis of productivity. That is, the longer the service lasts, the higher the compensation for the provider. There is no incentive for the provider to run the project in a more productive way. The project cost is calculated taking into account internal aspects, such as the effort taken and the level of expertise of the professionals. However, these aspects are mostly controlled only by the provider.

2. Fixed Price - It All Depends on the Scope

In this model the cost of the project is defined based on the scope presented by the client. In that case, the scope should be well-defined, which is something that is not always achieved. Therefore, more times than not, the bid includes a scope risk for the provider.

From the client's point of view, it is a comfortable model because there is a predictable cost; after all, the price is fixed!

But what happens if the initial price was poorly defined? What happens if there is a scope change in the project? In those situations, a new negotiation between the parties must take place, because the price will change. As the project is already underway, it is unlikely that the conditions for the renegotiation are the same from the original negotiation. Therefore, most of the time the client will succumb to a less favorable renegotiation instead of switching providers.

One advantage of this model is that the provider has an interest in being more productive, because it means more profitability.

The great challenge of working with a fixed price model is having a very well-defined scope for the software project, with low expectations of change. But how can we achieve this if the only certainty in software projects is that requirements will change? That's why another approach becomes necessary.

3. Unit Price – Sharing Responsibilities

This contracting model tries to balance the risks and responsibilities between both the client and the provider, combining the advantages and overcoming the shortcomings of the previous two models. Here, we can say that the scope

(continued on next page)

CHARISMATEK
SOFTWARE METRICS
www.charismatek.com

Function Point WORKBENCH
For the people who count.

Release 7.0L out now!

(Software Contracting and Management, continued from page 7)

management becomes the client's responsibility and productivity management becomes the provider's responsibility.

It is essential that the contract's unit of measurement represents a value delivery to the client. This means that not just any unit of measure can be used. For example, hours do not represent a value but a cost, and lines of code are difficult for the client to recognize their value.

Therefore, the challenge is to find a unit that is consistent, uniform, business-oriented and easily auditable. That's why the idea of using function points to measure software development contracts arises.

So, how can we use function points to manage software development contracts?

A simple approach is to use the function point size to estimate (or predict) the effort that the client will pay for the project.

To do that, we can use a model to estimate the effort using the size in function points as a primary input. There are several estimation models available in the market, but the simplest and most widely used in Brazil is to apply productivity to the software size project (in function points), as following:

$$\text{Effort (Hours)} = \text{size (FP)} \times \text{Delivery Rate (H/FP)}$$

The most commonly used productivity indicator is the rate of delivery, expressed in hours per function point, which is the average number of hours spent to produce a function point.

To use this cost model correctly, we need to do a productivity analysis before using this cost model. This study will examine the project's historical data, features and other attributes of the projects developed by the company.

You can find productivity numbers in publications and websites, but using any numbers without careful analysis, can lead to failure. Based on my experience as a consultant, it is a shortcut that does not work. In many organizations involved in process improvement initiatives, productivity analysis is a common practice.

Another useful indicator that can be derived is the ratio of the number of defects and the functional size, called defect level density. It is a useful tool for assessing a dimension of project quality, also useful in comparing software projects and analyzing the performance of the organization over time.

In project management, one of the most critical variables is scope. With function point analysis, it is possible to control the scope changes during project implementation. It allows for a direct and objective measurement of those changes.

Requirement changes are always present in software projects. If there is no way to quantify these changes, it becomes more difficult to assess the impact on the project.

Moreover, it's even more difficult to adequately communicate to the client any costs associated with the changes.

Now that we have talked about function points benefits in both project contracting and project management, we will discuss some of the challenges faced by organizations when deciding to make the transition to function points.

Difficulties While Transitioning

The first difficulty, and perhaps the most important one, is the adoption of the culture of planning. In many companies, software development and maintenance occur without proper planning. Without an accurate initial assessment of scope, requirements changes during the project end up being much more frequent. Therefore, additional work increases. Trying to adopt function point contracts without minimum planning maturity can be a scary situation because rework will be more visible and at higher levels.

Another common difficulty is to take the easy way of adopting FPA without making an accurate calibration of the estimation model (calibration is to adjust the estimation model with historical data from the organization). Some companies use published numbers, but at the end, they have to go back and make adjustments using their own historical data.

In this transition, we must be careful to use function points only for activities that are directly related to the software development and maintenance of software. I have seen companies that try to apply function points on issues that are not directly related to the software development activities (e.g.: support, training, hardware, etc.), which creates unrealistic expectations for the metric.

Measuring incorrectly and inconsistently is another difficulty, perhaps trivial, but still relevant. Some companies make the decision to use function points without proper team training. Moreover, many others do not have a strategy for quality assurance over its measurements. In that case, inaccurate measurements lead to wrong indicators and frustrating estimation results.

I've had the chance to hear some complaints from companies about the size of projects because the budget estimates were expensive. In those situations in which I could check the size, and the sizing measurement was right, I realized that the size of the project could be significantly reduced simply by streamlining requirements. That is, many requirements can be simplified and/or merged to form a leaner solution. FPA does not assess the quality of requirements; it only measures what has been specified.

Benefits

Now, let's talk about some benefits gained by organizations that have changed their software contracting model to one based on delivered results (and measured by function points).

(continued on next page)

(Software Contracting and Management, continued from page 8)

Increased capacity delivery is one perceived benefit. More projects are delivered in the same period of time due to the direct interest of the provider to achieve an increase in productivity and profitability.

Another positive effect is cost savings as the search for improved productivity and efficiency ends up reducing the number of defects in projects (corrective maintenance usually is not billed in FP contracts).

FPA also provides a very positive side effect: it induces verification and validation of the requirements specification. This leads to specifications improvements and minimizes the cost of repairing these defects in later stages of the project.

From the point of view of corporate governance, another benefit is that contracts using function points are easily auditable. For example, it is possible to track a payment to the provider at any time to quantify the direct result of the outcome. On the other hand, on body shopping contracts, a single payment is only related a number of hours, without assessing the results. In the Brazilian government, the man-hour contracting model was the focus of several public resources deviations.

Let's talk now about possible benefits in project management using function points.

The first point to take into account is the possibility of improving project planning. The size of a software project using function points can be used to produce estimations of effort, cost, time, and to promote a better assessment of the project scope.

In regards to monitoring and controlling projects, function point size reflects a quantification of the scope and permits resizing if any requirement were to change.

Since many indicators can be generated in conjunction with function points (like quality, productivity and scope indicators), it will help both in monitoring and controlling projects as well as software process improvements initiatives.

Last but not least, the communication with the client becomes much better when there is a metric that reflects something that the client recognizes and allows that person to perceive value: the software functions. The function point analysis concepts are the business user concepts, which is a huge advantage over technical metrics.

References

Questions and Answers about Function Points:

<http://www.fattocs.com/en/faq>

The Brazilian model for the procurement of software function points:

<http://fattocs.com/files/en/Articles/Quatic-2012-en-GuilhermeSimoes.pdf>

Software Outsourcing based on results:

<http://www.fattocs.com/en/services/software-outsourcing>

Counting Google3

By Julian Gomez, CFPS

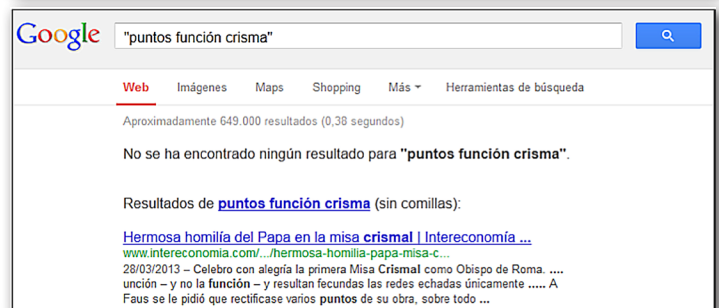
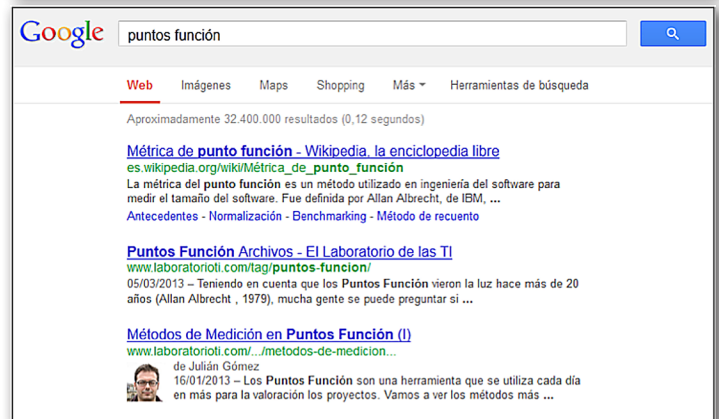
I've heard comments from certain people talking about how they can't measure their applications with function points.

"It's not possible, you know, my application is a web app and function points were not conceived to count the web."

"It's not possible; it's an iPhone / Android / Windows Phone app. We can't measure it with function points."

I'm happy to say: You CAN Measure Your App... with function points!

I'm going to show you a real example. I'm going to measure an application which is common to all of us: I'm going to measure Google.



(continued on next page)