

Journal of Agricultural Sciences Research

GOAL-BASED SOFTWARE BUDGETING METHODOLOGY

Luis Alberto Derico

Computer engineer. Universidad Católica
“Nuestra Señora de la Asunción” Campus
Itapúa.

All content in this magazine is licensed under a Creative Commons Attribution License. Attribution-Non-Commercial-Non-Derivatives 4.0 International (CC BY-NC-ND 4.0).



Abstract: In the southern area of Itapúa and surrounding areas, there is an accelerated growth of MSMEs, which, being affected by formal electronic information and taxation systems, in addition to the growing volume of information that they must handle, are a niche market. ideal for custom software development where development teams typically consist of one or two people.

The main characteristics of these developments is that they have well-defined objectives, and are restrictive in terms of time and budget, so a methodology that covers budgeting at the same time as development is necessary to avoid misunderstandings between clients and developers, as well such as budget losses, underestimation of work, or lack of development and implementation time.

In this work I propose a goal-oriented methodology, which focuses on an initial budget that is as exact as possible and accompanies each phase with specific products, that allow the validation of the development results and the quality control of the resulting software throughout the process.

Keywords: Software-Development-Budget-Methodology-Engineering.

INTRODUCTION

Nowadays, Micro, Small and Medium Enterprises constitute the majority portion of the Paraguayan market. According to data from the Vice Ministry of MYPYMES, by the end of 2017, MIPYMES make up 93% of the productive units in Paraguay.

With the new provisions of the Sub-Secretary of State for Taxation (SET), which is working on a process of total digitization of its operations, it is necessary for MSMEs to have the correct digital support that provides them with the information required by tax authorities control.

Currently, the SET manages electronically the information systems for taxable operations

(HECHAUKA), the settlement systems for operations taxed by VAT (MARANGATU), the VAT withholding systems (TESAKA) and more recently the settlements of the Income Tax. Personal Income (ARANDUKA) and more recently the upcoming implementation of the e-invoice system.

In this scenario, independent software developers must adapt to the characteristics of this majority segment of the market, to generate a product that meets the expectations of customers so that the system helps them manage their companies organizationally; and on the other hand that the valuation of the work is carried out correctly to ensure the continuity of the development service provider in the market, seeking a balance between a competitive price for clients and the necessary profitability for the developer.

The biggest problem for independent developers is that there is no concrete methodology that is adapted to this context and that meets the quality assurance metrics.

This paper defines the main characteristics that must be taken into account and explains the methodological proposal designed to combine all these aspects.

DEFINITION OF MSMES

According to Law Number 4457 sanctioned by the Congress of the Paraguayan Nation on May 16, 2012, Micro, Small and Medium Enterprises (MIPYMES) “are all economic units, which according to the dimension in which the work and the capital, are within the categories established in Article 5 of this law and deal with artisanal, industrial, agro-industrial, agricultural, forestry, commercial or service work”.

In Article 4 of this same law, it categorizes MSMEs by two elements:

- The number of employed workers; and,
- The annual gross billing amount,

made in the previous fiscal year.

In Article 5 of the same law, it classifies the parameters of each category and delimits the scope for each one:

Microenterprises: It is made up of a maximum of 10 workers, in which the owner or members of his family work personally in it and invoice a maximum of Gs annually. 500,000,000 (five hundred million guaraníes).

Small company: It is made up of a maximum of 30 workers and annually bills a maximum of Gs. 2,500,000,000 (two thousand five hundred million guaraníes).

Medium Company: It is made up of a maximum of 50 workers and invoices a maximum of Gs annually. 6,000,000,000 (six billion guaraníes).

IDENTIFICATION OF THE RESTRICTIONS OF MSMES

The two main restrictions regarding the development of a software project are:

- **Limited Time:** When a micro-enterprise or small business decides to implement administrative software, it is generally due to the fact that the current administrative and fiscal system no longer covers the current operational demand, either by internal accounting and administrative controls or by outsourced professionals.

The volume of operations of the company grew more than what its current structure supports and is forced to automate processes to save time.

As these changes are subject to the tax regime, most likely the expected development and implementation time will be less than one year.

- **Budget:** MSMEs do not see software development projects as investment projects or as services, but in the form of products, which makes them expect a

fixed price for certain features.

The company neither expects nor agrees that in certain cases the budget may increase due to last-minute changes or due to characteristics that had not been considered at the beginning of the software development. What the company expects is that a total value is agreed at the beginning of the project and that the functions are delivered as established, without exceptions.

SOFTWARE METRICS

Metrics are systematized standards that are applied to software or any process to verify that it has some specific property, both for measuring productivity and for software quality assurance.

There are many metric standards, but not all of them are useful for budgeting, since they measure the result, and not the future software development and implementation effort.

Some of the best-known metrics that we can cite are:

- ABC metric
- Degree of cohesion
- Construction cost model (COCOMO)
- Function points
- Number of lines of code
- Binary program size

CALCULATION OBJECT PER METRIC

ABC metric: measures the complexity of a software based on three main variables, assignments, branches and conditionals. The measurement object of the metric is the developed software, which makes it useless for budgeting, since the system must be finished to measure it.

Degree of cohesion: ordinal measurement

system that classifies software between desirable and undesirable traits. The object of measurement is the software developed and the objective is to measure the quality of the software, not the effort invested in its creation.

Construction cost model (COCOMO): is an estimative mathematical model where the cost of the final product is calculated based on the final size and number of generated files and other characteristics, but it does not take productivity into account and provides unrealistic values when using object orientation, due to inheritance properties and reuse, which affects the final amount of code written. Your measurement object is incomplete, because it anticipates the generated code but not the resources or productivity needed to create it.

Function points: metric that measures the functionality delivered to the user regardless of the technology used for the construction and exploitation of the software. The way of measurement can be somewhat empirical since it depends on the adjustment parameters applied by the person who performs the measurement. Its object of measurement is the characteristics of the software to be developed, but it is very subjective in its application and can give incorrect values in its application.

Number of lines of code: It tries to calculate the cost of software directly by the amount of code generated. The object of measurement is the software developed preventing it from being used for budgeting before developing the product.

Binary program size: Like the previous one, but instead of calculating by lines of code, it measures the weight of the binary files generated by the software. The object of measurement is the software developed preventing it from being used for budgeting before developing the product.

METRIC SELECTION

As seen in the list above, the most recommended options are those that allow calculating the effort before developing the software, leaving the metric by Function Points as the most viable option, adjusting the variables in the most objective way possible to adapt to the current environment.

BUDGET PROPOSAL IN THE REQUIREMENTS ANALYSIS STAGE

The phases into which the requirements analysis and budgeting process is divided are the following:

1. Book of Requirements
 - a. Definition of the main and secondary objectives of a Project
 - b. Definition of functional and non-functional requirements necessary for the system to meet the objectives
2. Function Point Rating
 - a. Function point calculation for each functional and non-functional requirement
 - b. Determination of the weight in hours of the function points
 - c. Determination of the value per hour in relation to the work team
 - d. Calculation of financial cost and project work time
3. Budget
 - a. Determination of the necessary investment
 - b. Clauses of services included and not included in the budgets
 - c. Scope
 - d. Development and implementation time

- e. Guarantee
- f. Developer and client protection clause
- g. Agreement to start activities

BOOK OF REQUIREMENTS

The first step to determine the effort required to develop software is to identify the functions that it must fulfill. This step must be executed after an initial data collection and the necessary interviews with the future users and managers of the software to clarify the expected results of the same.

Definition of the main and secondary objectives of a Project

They must define exactly what the main function of a system will be, and once achieved, list the activities necessary for that central objective to be fulfilled.

Definition of functional and non-functional requirements necessary for the system to meet the objectives

Once the objectives have been defined, the functions that need to be developed to meet them must be identified, and they are separated into two groups depending on their impact on the objectives.

The functional requirements are the functions that are directly involved in the fulfillment of the proposed objectives.

The non-functional requirements are those that do not intervene for the fulfillment of the objectives, but that are necessary for the operation of the software.

FUNCTION POINT RATING

When the identification of the requirements is finished, we can move on to their assessment metrics. Using function points, using the function point analysis method, we assign each requirement a number of "points" which we then convert to a more quantifiable scale.

The first step is to count the data

functions, from the user's point of view. The complexity indices are identified and later their contribution in points. The easiest way is to identify a system entity and assign a point for each operation on the data, namely; registration, cancellation, modification and consultation.

Transactional functions are then counted, such as additional calculations, application of formulas or reports.

The melting points are then counted and an adjustment factor is applied to them, which depends on 14 software features:

1. Data communication
2. Distributed processing
3. Performance
4. Equipment configuration
5. Volume of transactions
6. Online data entry
7. User interface
8. Online update
9. Complex processing
10. Reusability
11. Ease of implementation
12. Ease of operation
13. Multiple locations
14. Ease of changes

For this proposal, it is considered that these adjustment factors are not relevant in this budgeting process and an adjustment of 1 will be taken, since the fulfillment of the objectives must be total and it involves small or one-person work teams.

Function point calculation for each functional and non-functional requirement

A list of functional and non-functional requirements is made, the corresponding function points are assigned and the final

count is made.

Determination of the weight in hours of the function points

At this point, a direct assessment of a functional requirement is carried out, actually carrying out the execution of the requirement by timing the time invested, dividing it by the assigned function points to obtain the time invested in one function point.

Determination of the value per hour in relation to the work team

To value the estimated labor hours to complete the software requirements, the labor cost of each invested hour must be calculated. To do this, it is proposed to make an estimate of the value based on a normal working year, compensating for days off, downtime and operating expenses (calculated on the premise that the development is carried out by a team that does not work in a dependency relationship with a entity).

Calculation of financial cost and project work time

At this point, the work hours are multiplied by the value per hour calculated, and the approximate cost of the effort necessary to develop the system and the amount of time necessary to carry it out will be obtained.

BUDGET

With the previous data already calculated, the budget can be drawn up and presented to the person in charge of the software. The sections that a budget must have are the following:

Functionality and determination of the necessary investment

A detailed list of the functional and non-functional requirements of the system, plus its global or detailed assessment.

Clauses of services included and not included in the budgets

Delimitation of responsibilities of the development team and enumeration of

planned activities and activities that are not the responsibility of the development project.

Scope

What is the framework of the project, and how far will the budgeted activities go.

Development and implementation time

Total work time estimate. When implementation is included, the functional requirements must be added to those of the budget development.

Guarantee

Description of guarantees on planned development activities.

Developer and client protection clause

General forecast clauses for termination of activities, both on the side of the development team and those responsible for the software.

Agreement to start activities

A section to mark the official start of the works, also necessary for the evaluation of compliance and application of guarantees.

EXAMPLE OF APPLICATION OF THE PROPOSED METHODOLOGY

CASE STUDY

Evaluate the development of a system for a law firm:

Main goal: manage current accounts and administrative operations of clients of a law firm assigned to associates.

Secondary Objectives:

- Keep track of customers
- List the services provided by the associates
- Record direct costs
- Record indirect costs
- Maintain the checking account of clients
- Bill the services

IDENTIFICATION OF FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

From the proposed objectives, we can identify the list of functional and non-functional requirements of the project:

The ABMC can be considered with 6 total points, 4 correspond to data processes of registration, cancellation, modification and consultation; one point for the interface and one point for the entity creation.

Functions such as direct costs imply the same 4 points of ABMC, but additionally you need to consult the entities of associates, clients, and services; in the transactional category a possible calculation of subtotals is added to the interface.

Reports aggregate data points per query, plus transactional filtering and print formatting.

Non-functional requirements are evaluated in the same way as functional ones, although they do not directly impact the objectives.

Additional work such as installation of services or development of the base system must be evaluated from the perspective of the activities necessary for its development, for example, the installation of the database, in this case only one point, as it is a single service. On the other hand, the application server would be composed of an HTTP server, and a model manager, plus the basic template of the system.

For simplicity, the function points adjustment value will not be applied, leaving the adjustment index at 1.

In total these functions accumulate 70 points.

DETERMINATION OF WEIGHT IN HOURS BY PF

The determination of the weight in hours will depend on the number of people in the team and their experience with the

technological tools to be used.

As an example, we will start from the premise that it takes the developer 12 hours to develop a complete and functional ABMC from the creation of the database to the implementation of the interface, which would give us a base value of 72 minutes of work per point. of function. (1 database installation point, 3 necessary software installation points, 6 points for full ABMC).

Determination of value per hour

For the second part of the application of the methodology, we need the applicable monetary value for each hour of work. The determination of this value can be determined with the following procedure:

1. The maximum number of hours that can be worked in a one-year period must be determined: Hours per day x Business days x 4 weeks x 12 months = $8 \times 6 \times 4 \times 12 = 2,304$ hours per year.
2. The value of each hour is calculated based on a current minimum wage: minimum wage x 12 months / Total annual hours = $2,550,307 * 12 / 2,304 = 13,283$ G. per hour.
3. The days off per year are calculated in a weighting: annual holidays + days off + emergencies = $11 + 5 + 5 = 21$ days per year.
4. An adjustment must also be made for "dead time", which corresponds to work hours that are not effective, either due to mobility, preparations, or any work time that is not actually worked. This index can vary but initially 10% of the total hours can be applied: $2,304 * 10\% = 230$ hours.
5. A sum of the basic operating costs of the development team is necessary, which must be paid during the year, which varies according to each case. A basic example can be: (rent + cell phone +

N°	Requirements	Type
1	Associated ABMCs	functional
2	ABMC customers	functional
3	ABMC services	functional
4	Direct costs	functional
5	Indirect costs	functional
6	Report - customer account account	functional
7	ABMC invoice	functional
8	ABMC users	not functional
9	Functions	not functional
10	ABMC permissions	not functional
11	DB installation	not functional
12	Application server installation	not functional

Function Point Calculation

Number	Requirements	Data FAQ	Transactional faqs
1	Associated ABMCs	5	1
2	ABMC customers	5	1
3	ABMC services	5	1
4	direct costs	8	2
5	indirect costs	6	2
6	report - customer account account	4	2
7	ABMC invoice	7	2
8	ABMC users	5	1
9	functions	2	0
10	ABMC permissions	6	1
11	DB installation	0	1
12	Application server installation	0	3

internet service + supplies) * 12 months =
(800,000 + 120,000 + 130,000 + 100,000)
* 12 = 13,800,000 G. per year.

6. The cost of inactivity and the cost of downtime plus annual operating expenses must be evaluated, and weighted to the cost per hour per year: ((Days off per year * 8 * Value per hour) + (downtime * Value per hour) + annual operating expenses) / maximum annual hours = ((21 * 8 * 13,283) + (230 * 13,283) + 13,800,000) / 2,304 = 8,284 additional G. per hour.

7. For the final calculation of the work hour, the base hourly cost + the weighting of the annual costs + a percentage of profitability + taxes = 13,283 + 8,284 + 50% + 10% = 35,585 G. per work hour are added.

Calculation of financial cost and working time

With the variables already calculated, it can be determined that this project will have a cost of ((70 PF * 72 min) / 60 min) * 35,585 G., which gives a value of 2,989,140 G.

Total working time is 84 hours and 11 business days.

CONCLUSIONS

The proposed model seeks to balance the complexity of evaluating the development effort with the fairest valuation of the work effort in software development, where a realistic value is paid for the services received and where it is feasible for the development team to receive a decent compensation for your work.

The Paraguayan market in general and the southern zone of Paraguay in particular are taken as a parameter, orienting towards the SMEs and MSMEs market.

For the calculation of effort, it is assumed that the development will be carried out by a team of independent professionals or a single professional who will carry out all the work of the project.