El Modelo del Análisis de Precios Unitarios Industrial adaptado a un Proyecto de Desarrollo de Software para un Producto de Informática

The Industrial analysis model of unitary prices adapted to a Software Development Project for a Software Product

Cirilo Vega Magaña
Universidad Tecnológica del Valle de Toluca, México
cirilo_vega@hotmail.com

Ivan Steve Vega Herrera
Consultor Senior en Crucialsoft SC, México
ivancatarina@gmail.com

Resumen

En el presente trabajo se analiza la factibilidad de realizar el cálculo y estimación de costo de un proyecto de personalización de un producto de software empleando técnicas de costeo del Análisis de Precios Unitarios utilizados en Ingeniería Industrial.

Se contemplan los ajustes que se tendrían que hacer para poder utilizar el modelo de Análisis de Precios Unitarios, buscando las equivalencias de los conceptos usados en Ingeniería Industrial: maquinaria y equipo, materiales, mano de obra, jornada, etcétera, y aplicados a la ingeniería de software específicamente en proyectos de desarrollo y mantenimiento de software.

Mediante un modelo simplificado obtenido a partir del análisis se verifica la viabilidad de usar el modelo de costeo comparándolo con cifras obtenidas a partir de las estimaciones generales que se hacen en informática en el mercado mexicano.

Palabras Clave: análisis precios unitarios, estimación tiempo y costo desarrollo de software

Abstract

This paper discusses the feasibility of performing the calculation and estimation of cost of a project of a software product customization using costing techniques of the analysis model of unitary prices used in Industrial Engineering.

Contemplated the adjustments that will have to do in order to use the model of analysis of unit prices, seeking equivalences of concepts used in Industrial Engineering: machinery and equipment, materials, labor, day, etc., and applied specifically to software engineering in development and maintenance of software projects.

Using a simplified model obtained from the analysis verifies the feasibility of using the costing model compared with figures obtained from General estimates that are made in computer science in the Mexican market.

Key words: unit price analysis, estimate time and cost, software development.

Fecha recepción: Diciembre 2014 **Fecha aceptación:** Mayo 2015

Introduction

Most of the techniques of estimation of cost and software development times are based on the list of requirements of functionality that the software must meet and are made based on an estimated theoretical development time, most of the time from a description of the functionality that, in a few cases, is calculated as sufficiently detailed as to arrive at a calculation not to mention accurate, but at least suitable, functional, and that it will lead to the development team have a positive result economically, that is, that the project will generate dividends, especially when the budget is tight.

Another drawback that has shaped this estimate is that normally does the personnel of sales or in the best of cases, with one technical level sufficient to carry out the analysis of the elements of

operating with the required technical specifications; however, anyone who sells or carries out the initial analysis in general is not the person or the team that ends up running the job. Sales staff or which technically supporting sales lacks enough detail to make a survey of requirements sufficiently clear as to the given calculation.

Also influences the technical knowledge of the part of the person concerned on the product; since this is focused on fiscal issues, stakeholders are usually accounting or fiscal staff, and very rarely have support systems staff. In SMEs, with "General" operations, the standard configuration of the tool is sufficient to meet the reporting requirements, but in large or corporate enterprises, operations can be so complex that the Organization has a team of people in computer processing these data. In the latter case, sometimes the staff attorney/accountant does not have good communication with the computer area, and this drift that the computer part not collaborate enough and do your part required to make definitions in technology the user area can not by lack of technical knowledge.

From the computer point of view, an estimate of cost and effort implies to consider the time of one or more profiles that perform the following tasks:

- Requirements Analysis
- Component Design
- Component Building
- Component Testing
    - Component Installing and Configuring

If the above list of tasks is observed, and looking at it from the standpoint of Unit Prices as industrial engineering exercises, these activities only provide for the labor of a product which, although not physically tangible product, software ends being a product manufactured with a process similar to industrial manufacturing process.
Taking into account the aspects mentioned by the Industrial Engineering, for the generation of a product needs to consider:

1. Labor
2. Machinery
3. Materials

Converging these points with the Systems Engineering, labor is the time that each profile software development should invest, including staff running the activities and who directs and coordinates the development team.

The equipment includes computer equipment and other peripherals that are needed to generate the / software products, which includes computer equipment each developer uses, printers, communication devices, scanners, and so on.

The materials in computer area include: paper for printing documents that formalize the development (if applicable), consumables for printing equipment, storage devices such as flash drives or compact disks or DVD.

The hardest part is the estimated time of labor, specifically the development team. One advantage you have with the case study is that development tasks are of the same nature. For the analysis of the feasibility of using the technique of analysis of unit prices, it is taken as a case study customizing the application module consisting generate invoices from the information that is read from a text file issued by another system.

The case study

For reasons of confidentiality and in order not to expose crucial information about the operation of the application, since it is software that is sold commercially, specific software names will not be revealed and will mention them with a generic name.

The software in question is a commercial solution offered in the Mexican market to issue tax documents based on technological SAT defined definition for electronic generation of electronic accounting and tax documents.

The case study is the functionality available to the software to enable the capture of billing information and make a connection to an external ringing service to register and then generate an invoice in XML format and in PDF format.

In the standard version of the program, ie, the factory offered, the user is presented with a screen where you can capture the data to generate an invoice. Once the user has completed his capture, the software validates all the information needed to generate the bill is present and then sends that information to a stamped service of electronic invoicing which will return the invoice in XML format by an authorized stamp SAT. After receiving ringing service response, it generates the printed document or PDF. After this process, the system internally registers the new bill and this is presented to the user when the latter enters the billing system to verify.

The bill generated is commercially available as a standard invoice that includes only the fields that SAT requests, and these:

- the issuer RFC
- RFC receptor
- Residence tax receiver
- Folio
- List of products
- Rode
- Percentage of tax
- Tax
- Total

When a client requires the bill in print include more information, such as a serial number, a specific legend or any other aspect not included in the standard structure, the development team must make internal changes to system programming for make it happen.

On the surface, add the field shown simple, but depending on the conditions that are around this simple fact, the task seems easy can be converted to work more than 40 hours of programming.

The conditions that determine the complexity of the task are:

• ***The origin of the data.*** *Where does it come from? Does the user to capture or be recovered from any other source?*

• ***The persistence of data.*** *Does the data must remain available once this was added to the printed format or not? Have you must save in a repository and be associated with some other information?*

• ***The presence of data.*** *Where the data must be present? Onscreen? Printed?*

• ***Using the data. Is it informative?*** *Is it used in calculations? You can be used as a search term?*

Conduct a thorough analysis of these different possibilities combinations can give, it is beyond the scope of this paper. What is sought is a method of estimating time and effort that can be used to make the calculation of the cost to the company to make changes to software programming that can respond to the needs of a particular customer, and this cost in turn can be used as a basis for calculating the selling price to the consumer.

As described in the introduction of writing, usually the work of software development are listed based on how long a multidisciplinary development team needs to perform the work, leaving out the costs they represent for the organization to use the different equipment computer, peripheral equipment, communication, printing and all kinds of consumables associated with them.

What is observed in the technique of Unit Prices of Industrial Engineering, is that any quote considers these three elements; making the analogy described in the introduction, it is seen entirely feasible to adapt this technique to software development projects.

Adapting materials

It starts by this section to be the simplest. The materials in the computer part covers:

- printing paper
- Ink cartridges
- Folders
- Storage devices (external hard drives, CD-R / RW, DVD-R / RW drives)

A unit value should be given to each of these elements, including, if the organization so defined, a percentage of profit or extra costs for internal management.

Adaptation of machinery

Machinery costs applied to Computer Engineering are to:
Depreciation cost of the computer equipment used. Normally a computer desktop computer / laptop has a shelf life of 3-5 years. 5 will be taken in this case. The team lost its resale cost in 3 years and in five years will fall to an estimated one thousand pesos. After the 5th year is considered not resalable and its cost is amortized to 1.

For purposes of the simplified model generating the next number will be used:
Average cost of computer equipment: 13 000 pesos
3-year monthly depreciation: 13 000 pesos / 36 months = 362 pesos / month
Depreciation per hour of use (20 days 8 hours) = 362 pesos / 160 hours = 3 pesos / hr
Communication equipment. When used and are provided by the company depreciation cost network equipment (router, switch), telephone equipment (mobile, fixed, voice ip) was contemplated.
Teams printing, scanning and copying. A period of 5 years of depreciation laser and inkjet printers, scanners and copiers is contemplated.
External storage devices. A period of 5 years of depreciation to external storage devices such as external hard drives are contemplated.

Adaptation of the cost of labor

In this case three ranges are only considered Developer:

1. beginner
2. Intermediate
3. Advanced.

Average monthly salaries for each range used:

- Beginner: 15 mil
- Intermediate: 20 mil
- Advanced: 25 mil

The average cost per hour of each resource with the following formula is calculated:

Average cost per hour = (monthly cost x 12) / (total number of working days per year x 8).

Simplified figures:

12 months on average 20 working days per month

Beginner: 180 000/1920 = 93.74 pesos / hour - is rounded to 94 pesos / hour

Intermediate: 240 000/1920 = 125 pesos / hour

It advanced: 300 000/1920 = 156.25 - 157 rounds pesos / hour

Average time of execution of each task

In order to gauge the cost of a given development, activities that are commonly needed in the implementation of the application and customization of a screen or printed format is categorized. The tasks are:

- Add a field on the screen
- Add a field in print
- Get a database data
- Insert an entry in database

• Updating a Database data

Each task is valued at a standard time, which can vary depending on the complexity. To all intents and purposes the industry standard, tasks are categorized into simple, average and complex. It is necessary that any development is expressed in tasks that can be categorized as well to be able to calculate in turn.

According to the type of task and their categorization, we have the following standard times:

Tarea	Complejidad	Tiempo
Agregar/eliminar/modificar un campo en pantalla.	Baja Media Alta	1hr 2hr 4hr
Agregar/eliminar/modificar un campo en formato impreso.	Baja Media Alta	2hr 4hr 8hr
Obtener un dato de base de datos.	Baja Media Alta	2hr 6hr 12hr
Insertar un dato en base de datos.	Baja Media Alta	1hr 3hr 6hr
Actualizar un dato en base de datos.	Baja Media Alta	1hr 2hr 4hr

These tasks must be added an estimated time of testing. In general, it is customary to calculate 30% of the development time as the time required to verify the functionality of a component. Taking the estimated times, times of testing result (closing half hours):

Tarea	Complejidad	Tiempo
Agregar/eliminar/modificar un campo en pantalla.	Baja Media Alta	0.5hr 1hr 1.5hr
Agregar/eliminar/modificar un campo en formato impreso.	Baja Media Alta	0.5hr 1.5hr 2.5hr
Obtener un dato de base de datos.	Baja Media Alta	0.5hr 2hr 4hr
Insertar un dato en base de datos.	Baja Media Alta	0.5hr 1hr 2hr
Actualizar un dato en base de datos.	Baja Media Alta	0.5hr 1hr 1.5hr

Test model

To test the accuracy of the calculation model with unit prices, compared with the estimation model based solely on development time, the following case will be taken, making the cost calculation with the parameters described and then compared with the estimated cost for development and finally against the actual cost that meant the same.

Test case:

When a user wishes to issue an invoice, the system checks whether existing billing information receiver. Where they exist, the user can change if there has been an error in the catch. It broadcasts a message on the screen when the user wants to change the data and when the same message has already changed and proceeds to update the data in database. Additionally it is necessary that the button lets you change the largest data is in blue background and change the text that displays for "change" instead of "change." As product development results of executed tests are given printed.

The division of tasks with respect to the definition of the model is performed to obtain the estimated total number of hours:

Delete the message when you click on the button change
Standard task: remove display field
Complexity: Low
Estimated time: 1 hr
Test time: 30 min

Change button to modify the data, to put it with blue background and change the legend showing the same:
Standard task: changing screen field
Complexity: Low
Estimated time: 1 hr
Test time: 30 min

The work would involve three hours of development calculated with the tables defined for the model unit price.
The same work was estimated at 4 hours of development over 1.5 hours of testing with the general model estimation, for a total of 5.5 hours.

Cost analysis:

Modelo precios unitarios	**Costo**	**Modelo general**	**Costo**
Tiempo de desarrollo con desarrollador principiante	3hr x 94 = 282 pesos	Tiempo de desarrollo con desarrollador principiante	5.5 x 94 = 517
Maquinaria Depreciación del equipo de cómputo	3hr x 3 = 9 pesos		
Materiales: 3 hojas	3 x 1peso = 3 pesos		
Total	**$294 pesos**		**$517 pesos**

The actual execution time was 2 hours development. The actual cost was:

2 hours Developer: $ 198

3 sheets: $ 3 pesos

Comparing costs we have:

Costo/Tiempo PU	Costo/tiempo estimación	Costo/tiempo real
$294	$517	$201
3 hr	4.5hr	2hr

The time and cost of model-calculated unit prices were much closer to the data obtained in the actual execution of the project.

Deviations in the estimation model

Projects and development activities in the industry is normal to have deviations of up to 30% of the time / cost resulting from the disadvantages that occur during the execution of the work itself, sometimes motivated by lack of deep knowledge of the subject you are working (eg, accounting systems programming without having basic knowledge of accounting developer) makes

impossible to anticipate needs or barriers, sometimes due to ignorance of the universe of information that an organization owns (eg, lack of data where as the rule should not apply).

It is unusual for a development team maintained solely by a single development, ie, usually have more than one project at a time and assume that the team will be 100% of the time focused on the planned tasks is unrealistic.

Information systems are designed and built to solve real-life problems of organizations and individuals. These problems and information needs change over time and information systems must evolve at the same time to continue to respond to the needs of its users. A common factor for the deviation of the estimated times is that during the construction of a software component is a change in the scope of the original needs to be had, either by the appearance of a new requirement of functionality, the need to add more components to an agreed, new requirements deriving from legal definitions of character that authority in turn determines, among other development.

Conclusions

Model Unit Prices of Industrial Engineering project fits perfectly to a Software Development Project with the adjustments presented in this article.

Regardless of how it is used to estimate costs and development time, either by unit or by assessment staff time prices, both models need to keep a historical record by activity, allowing to have statistical actual data from the time it takes a developer of the three types of experience (beginner, intermediate, advanced) and obtain an overall average time per type of activity and complexity.

The model can be extended and made more representative if further average times are obtained much experience and type of complexity, ie average time a novice developer, average time of an intermediate developer and average time an advanced developer. In such a way that an estimate

of time and cost can be made with different combinations of job templates, which can have a mixture of experience and profiles number of different resources.

The computer field as an expert in information could devote part of their time and resources in creating a software solution that allows keeping track of these statistics and in turn get the average time and costs as well as the generation of calculation time and cost estimates for the project, based on information from the breakdown of tasks that feeds the system.

Bibliography

Software estimation techniques. Recuperado de https://en.wikipedia.org/wiki/Software_development_effort_estimation

Estándar IEEE 830-1998 – Recomended Practice For SoftwareRequirements Specifications Software Engineering Standards Committee of the IEEE Computer Society June 1998 ISBN 0-7381-0332-2

Estándar IEEE 1016-2009 Standard for InformationTechnology—Systems Design—Software Design Descriptions Software Engineering Standards Committee of the IEEE Computer Society March 1999 ISBN ISBN 978-0-7381-5926-3

Metodologías Ágiles de Desarrollo. Recuperado de http://agilemethodology.org/

Manifiesto de las metodologías ágiles de desarrollo. Recuperado de http://www.agilemanifesto.org/

Modelado y documentación de sistemas ágil Agile Model Driven Development (AMDD) Recuperador de http://agilemodeling.com/essays/amdd.htm

2003-2012 Scott W. Ambler Ciclo de vida de desarrollo de software http://www.tutorialspoint.com/sdlc/sdlc_overview.htm

Software Test Estimation: How to estimate testing efforts Recuperado de http://inderpsingh.blogspot.mx/2010/03/how-to-estimate-testing-efforts-6.html Marzo 2010