Software projects effort estimation using “use case points” method in the context of Project Management Body of Knowledge

R. Uzal (*) (**), G. Montejano (*), D. Riesco (*)
(*) Universidad Nacional de San Luis
Ejército de los Andes 950
5700 San Luis - Argentina
gmonte@unsl.edu.ar
(**) Universidad de Buenos Aires

N. C. Debnath
Winona State University
Winona, MN 55987 - USA
ndebnath@winona.edu

Abstract

As a contribution to consider software projects to be risk investments, it is important the full standardization of either the planning methodology (as a subset of project management methodology) and the effort estimation method. We think there is important and abundant evidence backing the convenience of the join use of Project Management Body of Knowledge and “Use case points” method in software project planning and control. Software development teams should share information, as it is important to compare the “ex post” analysis of programming/budgeting estimations against project execution data with “real world” software projects. This comparative activity would include teams into a permanent improvement process.

Keywords: Software engineering economics, PMBOK, use case points method

1. Introduction

Software development projects are investment projects [1]. Every investment has three key characteristics: Expected Return, Risk and Marketability.

Expected return: It refers to the amount of interest, dividends or capital gains that you expect to receive from your investment. (Actual returns may, of course, be quite different.). There is a direct correlation between expected return and risk. We can say: “The higher the expected return, the greater the risk”.

Risk: It is the possibility of losing some, all or more than your initial investment, or the possibility of receiving less return than you expected. Lower risk investments include, for example, government treasury bills. In the case of software development projects, they are usually high risk investments [2] so, investors will expect high returns.

Marketability (or 'liquidity'): This characteristic refers to the possibility of selling or redeeming your investment quickly at or near the current market price. Term deposits are an example of an illiquid investment, since you are not generally allowed to withdraw your money before the end of the term without paying a significant penalty. Several other investments, such as mutual funds or listed securities, are very marketable as they can be quickly sold or redeemed on short notice and at low cost. Marketability is an important factor to be considered when an investor is selecting his/her investments. Software development projects are generally illiquid investments.

Important investors are needed to face important software development projects. An investor will use a variety of figures to evaluate the financial attractiveness of a software development project. He/she (investor) needs to estimate the project’s capital cost, projected earning, annual revenues, expenses, and tax impact. The three primary figures to be used by investors are:

Net Present Value: Net Present Value (NPV) is the sum of all years’ discounted after-tax cash flows. The NPV method is a valuable indicator because it recognizes the time value of money. Projects whose returns show

positive NPVs are considered to be attractive.

Internal Rate of Return: Internal rate of return (IRR) is defined as the discount rate at which the after-tax NPV is zero. The calculated IRR is examined to determine if it exceeds a minimum acceptable return, often called the hurdle rate. The advantage of IRR is that, unlike NPV, its percentage results allow projects of vastly different sizes to be easily compared.

Payback Period: A payback calculation compares revenues with costs and determines the length of time required to recoup the initial investment. A Simple Payback Period is often calculated regardless the time value of money. This figure of merit is frequently used to analyze retrofit opportunities.

As a preliminary conclusion we can say that a software development project needs a well supported "projected cash flow" in order to begin conversations with potential investors. "No investors, no software project"

In order to obtain the required well supported "projected cash flow", high quality effort estimation method and a consistent planning methodology are needed. In this paper we strongly suggest the join use of Project Management Body of Knowledge (PMBOK) by Project Management Institute (PMI) [3] and "Use Case Point" method [4] [5] [6] [7] in the context of software development planning and control.

2. The "use case points" method

A brief presentation and any reference of "use case points" method as described by Schneider, G. and Winters, J. [4] is developed through this paper. This estimation procedure requires the possibility to count the number of transactions in each use case. A transaction is an event occurring between an actor and the system, the event being performed entirely or not at all.

There are four steps included in "use case points" method:

1. The actors in the use case model are grouped as *simple*, *average* or *complex*. A simple actor represents another system with a defined API; an average actor is another system interacting through a protocol such as TCP/IP; and a complex actor may be, for example, a user (human being) interacting through a graphical user interface or a web-page.
A weighting factor is assigned to each actor group:

• Simple: Weighting factor 1
• Average: Weighting factor 2
• Complex: Weighting factor 3

The total *unadjusted actor weight* (*UAW*) is calculated counting the number of actors in each group, multiplying each subtotal by its specified weighting factor, and then adding subtotals.

2. Each use cases is also categorized as *simple*, *average* or *complex*, depending on the amount of transaction included in a specific use case. We must also consider the transactions in alternative flows. Included or extending use cases are not considered. In general terms, a simple use case has 3 or fewer transactions; an average use case has 4 to 7 transactions; and a complex use case has more than 7 transactions. A weighting factor is assigned to each use case category:

• Simple: Weighting factor 5
• Average: Weighting factor 10
• Complex: Weighting factor 15

The *unadjusted use case weights (uUCW)* is calculated counting the number of use cases in each category, multiplying each category of use case with its weight and adding the products. The UAW is added to the UUCW to get the *unadjusted use case points* (*UUPC*).

3. The use case points are adjusted based on the values assigned to a number of technical and environmental factors.

Each factor is assigned a value between 0 and 5 depending on its assumed influence. A rating of 0 means the factor is irrelevant for the project; 5 means it is "essential".

The Technical Factor (TCF) is calculated multiplying the value of each factor by its weight and then adding all these numbers to get the sum called the *TFactor*. Finally, the following formula is applied:

TCF = 0.6 + (.01 * TFactor)

The Environmental Factor (EF) is calculated accordingly by multiplying the value of each factor by its weight and adding all the products to get the sum called the *Efactor*. The formula below is applied:

EF = 1.4 + (-0.03 * EFactor)

The *adjusted use case points (UCP*) are calculated as follows:

UCP = uUCP * TCF * EF

Karner [7] proposed a factor of 20 staff hours per use case point for a project estimate, while Sparks states that field experience has shown that effort can range from 15 to 30 hours per use case point [6]. Schneider and Winters recommend that the environmental factors should determine the number of staff hours per use case point [4].

According to our own studies and experience [1] [2] [10] [11] [12] [13] we agree with Sparks' point of view.

Example of use case points method application

Here we mention a project at COMPSIS Company located in Brazil [8][9]. The project's aim was to construct a software system used to register the data about customers of electronic toll collection. It was called TURS (Toll User Registration System).

The list of actors and identified use cases for the system are presented in Tables 1 and 2 with their respective complexities.

Table 1: Example of "real world" application of use case points (actors)

Actors	Complexity
Lane System	Average
Operational Manager	Complex
Vendor	Complex

Table 2: Example of "real world" application of use case points (use cases)

Use cases	Complexity
Pass registration	Average
Customer registration	Average
Order registration	Average
Order payment	Simple
Pass delivery	Simple
Interface with lanes	Simple
Customer list report	Simple
Financial report	Simple

The complexity for each actor and use case identified during this phase was determined following Karner's specification [7] and its criteria are synthesized in Tables 3 and 4.

Table 3: Criteria for actor complexity determination

Complexity	Definition	Weight
Simple	Represents a system with API available	1
Average	Represents an interaction with another system through a protocol There is a human interaction with a line terminal	2
Complex	Represents an interaction with a graphical user interface	3

Table 4: Criteria for use case complexity determination

Complexity	Definition	Weight
Simple	3 or less transactions	5
Average	Between 4 and 7 transactions	10
Complex	More than 7 transactions	15

Using the following formula:

$$uUCP = \sum_{i=1}^{6} n_i * W_i$$

The unadjusted use case weights (UUCW) were calculated, where:

uUCP => total of unadjusted UCP
i =1..6 => the list of complexity levels for actors and use cases.
n_i => total of actors and use cases identified and organized by complexity level
W_i => influence factor according to the complexity of factor and use case

In the case of TURS project, the calculation was:

uUCPActors = 1 * 2 + 2 * 3 = 8
uUCPUse Case = 5 * 5 + 4 * 10 + 2 * 15 = 95
uUCP = uUCPActors + uUCPUse Case = 103

The application of the Technical Factor (TCF) and Environmental Factor (EF) was not reported in this example by authors [8][9]

According to our experience (several e-government systems development [10] [11] [12]) “use case points” method has clear advantages compared to “line of code”, “function points”, “COCOMO” and other “traditional methods” in the effort evaluation field.

Currently we (the authors of this paper) are facing the effort evaluation for a Health Care Provider System including Technical Factor (TCF) and Environmental Factor (EF) whose results will be reported in the near future.

3. The Project Management Body of Knowledge use in software projects

The PMBOK presents Project Management practices in logical dimensional groups. One dimension describes "knowledge areas" while the other dimension describes project management processes split into five process groups

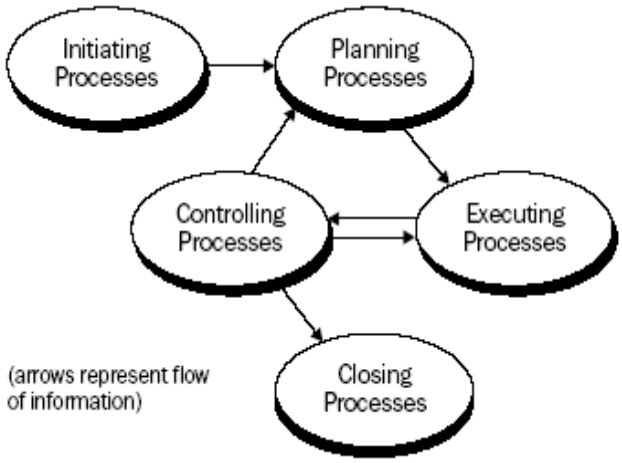
The PMBOK knowledge areas are:

i. Project Integration Management
ii. Project Scope Management
iii. Project Time Management
iv. Project Cost Management
v. Project Quality Management
vi. Project Human Resource Management
vii. Project Communications Management
viii. Project Risk Management
ix. Project Procurement Management

The 39 processes are organized into five process groups:

- Initiating Processes
- Planning Processes
- Executing Processes
- Controlling Processes
- Closing Processes

We can see theses five process groups:



We can mention, as PMBOK output (Some of the main PMBOK deliverables):

- Project Plan (with supporting details)
- Work Results and Change Requests
- Corrective Actions and Lessons Learned
- Project Charter (with constraints and assumptions)

The Project Plan implies to be:

- Created in Project Plan Development process in the early phases of a project lifecycle
- Updated throughout the project lifecycle
- Outlined as follows:
    - Project Charter
    - Project Management Approach or Strategy
    - Scope Statement
        - Project objectives
        - Project deliverables
    - Work Breakdown Structure (WBS)
    - Cost Estimates, Schedule and Responsibility Assignments for Deliverables
    - Measurement Baselines for Scope, Schedule and Cost
    - Major Milestones and Target Dates
    - Required Staff
    - Other Plans

Throughout the development of this paper we have strongly recommended [13] the use of PMBOK instead of "commercial" or "proprietary" methodologies for project management. We are suggesting the use of "use case points" method to support some of PMBOK processes/ knowledge areas.

4. "Use case points" method supporting PMBOK areas of knowledge

As pointed previously, we are applying "use case points" method in the context of PMBOK methodology (Health Care Provider System development).

The following table shows the use of "use case points" method considering PMBOK areas of knowledge

PMBOK areas of knowledge	Use case points method application
Project Integration Management	
Project Scope Management	Scope changes are "the rule" in software projects. Scope change quantification is an important application of use case points
Project Time Management	Programming (project calendar) is a very important area of use case points application.
Project Cost Management	Budgeting is a critical area of use case points application to effort estimation
Project Quality Management	Quantitative Software Quality Assurance planning topics could be an application of use case points
Project Human Resource Management	Human Resource allocation is a another critical area of use case points application to effort estimation
Project Communications Management	
Project Risk Management	Quantitative Risk Management planning topics could be an application of use case points
Project Procurement Management	Quantitative Project Procurement Management planning topics is an important area of application of use case points method

5. Conclusions

a. The use case points method is, perhaps, the best available tool for effort estimation in the context of software projects. It has been proposed to estimate the software development in early phases of software project and used by a lot of software organizations. Intuitively, UCP is measured by counting the number of actors and transactions included in use case models. Several tools to support calculating UCP have been developed. The acceptance of UCP as a generalized standard will be an important contribution to software industry.
b. PMBOK, used in software projects, is considered to have clear advantages regarding commercial methodologies. An aggressive commercial pressure, has evidently well positioned some of these commercial methodologies in the market. Therefore, the increasing adoption of PMBOK, in software development teams, will also be an important contribution to software industry.
c. Several PMBOK processes / knowledge areas can be supported applying “use case points” method.
d. Software development teams must use their own statistical data comparing programming / budgeting estimations against project execution data. This comparative activity will include teams into a permanent improvement process. If software development teams share these statistical data, in the future, software industry will use very consistent and useful tables when programming and budgeting tasks.

6. References

[1] R. Uzal, G. Montejano, D. Riesco, et al, “Software Engineers First Duty: the Preliminary Financial Feasibility Analysis of Real World Software Projects”, 5th IEEE International Symposium on Signal Processing and Information Technology (IEEE ISSPIT-2005), December 18-21, 2005, Athens, Greece

[2] N. C. Debnath, R. Uzal, G. Montejano, D. Riesco, “Software Projects Leadership: Elements to Redefine Risk Management" Scope and Meaning”, IEEE-EIT06. 6th IEEE International Conference on Electro/Information Technology (IEEE EIT-2006) May 7-10, 2006 Michigan State University East Lansing – Michigan – USA.

[3] www.pmi.org Project Management Institute, A guide to the project management body of knowledge (PMBoK® Guide) (Newtown Square, PA: Project Management Institute, 2004).

[4] Schneider, G. and Winters, J. *Applying Use Cases – A Practical Guide*. Addison-Wesley. 1998.

[5] Smith, J. The Estimation of Effort Based on Use Cases. Rational Software, White paper. 1999.

[6] Sparks, S. and Kaspcynski, K. The Art of Sizing Projects, Sun World. 1999.

[7] (http://www.sunworld.com/sunworldonline/swol-12-1999/swol-12-itarchitect.html).

[8] G. Karner, Resource estimation for Objectory projects, *Objective Systems SF AB*, 1993.

[9] T. C. Chiossi, and F. S. R. Germano, Software project management (Campinas, Brazil: Computing Institute – UNICAMP, 2001).

[10] T. C. Chiossi et al, ”The integration of use case points, COCOMO II and earned value analysis within PMBOK guide, International Conference on Software Engineering and Applications, IAESTED, 2006.

[11] R. Uzal, G. Montejano, D. Riesco, et al, "Designing a particular Balanced Scorecard to Manage the last and very difficult stage of a very important hydroelectric project", Proceeding of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications, The International Society for Computers and Their Applications, Rio de Janeiro, Brazil, pg 23-26, June 2003, ISBN 0-9742059-0-7.

[12] R. Uzal, G. Montejano, D. Riesco, M. Peralta, C. Salgado, "Improving the Management of the last stage of the Yacyreta Hydroelectric Project trhough the use of Kaplan and Norton's Balanced Scorecard Scheme", Proceedings of the 15th International Conference, Computer Applications in Industry and Engineering, the International Society for Computers and their Applications - ISCA, pg 252-255, San Diego, California, USA, November 7-9, 2002. ISBN: 1-880843-45-5. This publication is abstracted and indexed in INSPEC.

[13] J. Dujmovic, E. Petrolo, R. Uzal, A. Dasso, A. Funes, D. Riesco, G. Montejano, "Software Quality Assurance in a Project Based on Rapid Evolutionary Prototyping Methodology", 17th Anual International Conference. Proceedings of Computer Science. The Association of Management and the International Association of Management, Volume 17, number 3. Part C. August 6-8, 1999, San Diego, California, USA. ISGN:0-9668650-6-5. Pg 147-152. http://cs.sfsu.edu/People/jozo/respub.html

[14] Roberto Uzal, Germán Montejano, Daniel Riesco, “Pautas para optimizar la gestión de proyectos de software”, CACIC06, San Luis, del 24 al 27 de Octubre de 2006. ISBN CD: 950-609-050-5 ISBN impreso: 950-609-049-1