PROPOSAL FOR AN USE CASE POINT ANALYSIS

Claudio G. Bernardo, Stephanie L. Nascimento, Adrilles R. Mariano

Universidade Paulista – Barueri – SP. E-mail: adrilles@uol.com.br

ABSTRACT
The aim of this paper is to present Analysis by Use Case Point that is used for specifying requirements in different systems. This tool is important for software development, cost versus time for states prepared to help in planning any activity. A proposal to solve a case of calculations in a lawyers' association, which has the priority map all your processes and create systems that can improve customer service while remaining competitive in your market.
Keywords: Use case Point - System Requirements - Software Development

PROPOSTA DE ANÁLISE DE PONTOS DE CASO DE USO

RESUMO
O objetivo deste trabalho é apresentar a Análise por Ponto de Caso de Uso, que é usada para especificar os requisitos em diferentes sistemas. Esta ferramenta é importante para o custo de desenvolvimento de software. Outro objetivo é propor a resolução um caso de cálculos em uma associação de advogados, que tem no mapa de prioridade todos os seus processos e criar sistemas que possam melhorar o serviço ao cliente para se manter competitivo no seu mercado.
Palavras-chave: Ponto de caso de uso – requisitos de sistema – desenvolvimento de software

INTRODUTION

The task of system drive and analysis of requirements can be regarded as the cornerstone of the process of software development, given its relevance, because it is an allowance for other phases related to the specific tasks of development.

The process of requirements gathering and analysis has, according to Jair (2000) and Leffingwell and Widrig (2003) as main objectives: To provide system developers with a better understanding of stakeholder needs, which are the professionals directly involved with the project from the end user to responsible for approving the project development; Define the limits of the system (system scope); provide a basis for planning the technical content of the stages of development; provide a basis for estimating cost and development time of the system; define a user interface for the system, focusing on the needs and goals of the users; Establish and maintain compliance with customers and other stakeholders about what the system should do.

To achieve these goals, it is important first of all understand the definition and scope of the problem to solve. Stakeholders must be identified and their requirements must be identified, collected and analyzed. The requirements of a system can be classified as functional and nonfunctional [Kotonya and Sommerville 1998]. Functional requirements are those that describe the behavior of the system and how it interacts with users or other systems. The Non-Functional Requirements are those that describe other constraints of the system being developed. In general, we can classify as functional requirement system functionality and nonfunctional requirements like usability, reliability, performance and scalability of the system.

One of the tools used to specify the functional requirements of a system is the Use Case Model. A Use Case Model consists of two types of documents: The Use Case Diagram and Use Case Specification [Camargo 2010] [Fan, Xiaohu, Xiaochun, and Lu 2009]. The main purpose of this paper is to present the main concepts of specifying requirements with use cases and the use of metrics for estimating projects through Use Case Points Analysis applied in an institution.

FUNCTION POINT

It was created by ISO/IEC 20926, International Standard formalized the IFPUG Function Point Analysis method [Dias 2003], the vast collection stored for comparison between different organizations and certifications offered by non-governmental entity IFPUG - International Function Point Users Group, responsible for the method of Function Point Analysis, many design deficiencies are found during development [BFPUG 2011].

The technique of Function Point Analysis helps managers to measure productivity related to the efforts in process development, optimization and system maintenance. At first the Function Point Analysis looks very promising as a method to aid the task of managing the development of computer systems. According to Dias (2003) "measures the functionality of the system based on the user's view, having the following characteristics: Independence of technology used; production based on the view of the user; Significance for the end user usage estimates; Prone to automation.

One can understand that this type of method falls in the background or put questions to those who will not perform basic system specifications. For the IT manager the 'how' will appear to have less weight than the 'what' should be implemented. This fact is quite comprehensive as this type of professional, most often, is directly

linked to planning activities. This sidelined the technical aspects related to the implementation by IT managers can lead to inefficiency or even a quality final product less than planned.

It is interesting to go deeper in the methodologies that make up the Function Point Analysis since its spin-off can provide more elements that have this disagreement between the planning and execution process.

By analyzing the functions of the Function Points the sum of the contributions of all components results in the number of unadjusted function points [BFPUG 2011]; it means that the complexity depends not only on quantitative values such as numbers of logical records and data items referenced. Often both the programming language and development platform (hardware) may have two equivalent solutions in a matter of performance and fulfillment of needs, and yet they have different numbers in relation to the total number of logical records and data items referenced.

USE CASE POINTS

The UCP - Use Case Points were created by Gustav Karner in 1993 as a specific adaptation of Function Points. UCP is a technique for modeling software that helps developers determine which features should be implemented and how to resolve errors by means of metrics [Bittner and Spence 2002]. This metric allows estimating early in the project. His technique is based on the definition of Function Point Analysis (FPA), in which the functionality seen by the user is the basis for estimating the size of the software [Vazques 2008] [Aguiar 2011]. According to Medeiros (2004) the counting process this metric consists of the following steps: a. Relate the actors, ranking them according to their level of complexity (simple, average or complex) respectively assigning the weights 1, 2 or 3 as shown in table 1.

b. Calculate TNAWA (Total of Not Adjusted Weight of Actors) adding the products of the amount of players by their weight; ***c.*** Counting the use cases and assign the degree of complexity where complexity classes based on number of transactions; ***d.*** Calculate TNAWUC (Total of Not Adjusted Weight of Use Cases) adding the products of the amount of use cases by their weight as shown in table 2; ***e.*** Calculate NAUCP (Not Adjusted Use Case Points) according to the formula: NAUCP = TNAWA + TNWAUC; ***f.*** Determine the Technical Complexity Factor (TCF). The technical complexity factor varies on a scale of 0 to 5, according to the degree of difficulty of the system to be built.

The value 0 indicates that the degree is absent or is not influential, and 3 indicate the medium influence and 5 indicates a significant influence through the whole process.

Table 1. Specification Weight Actors

Actor Complexity	Description	Weight
Simple	Very few entities involved and Databases without business rules	1
Medium	Few entities involved and databases with some complex business rules	2
Complex	Complex business rules and many entities in this database	3

Font: (Medeiros 2004)

After determining the value of the factors, multiply by its respective weight shown in table 3, add up the total and apply the formula: Technical Complexity Factor (TCF) = 0.6 + (0.01 * Sum of Technical Factor). ***g.*** Determine the Environmental Complexity Factor. The environmental complexity factor indicates the effectiveness of the project and are related to the level of professional expertise. These factors described in table 4 are determined using a scale of 0 to 5, where 0 indicates low experience, 3 indicates medium experience and 5 indicates high experience. After determining the value of each factor, multiply the weight and add the total values. Then apply the formula:

Environmental Complexity Factor (ECF) = 1,4 + (-0,03 * Sum of Environmental Factor).

h. Calculate AUCP (Adjusted Use Case Point). This calculation is based on the multiplication of NAUCP (Not Adjusted Use Case Points) by the technical complexity and by the environmental complexity, as illustrated in table 4, using the following formula: AUCP = NAUCP * Technical Complexity Factor * Environmental Complexity Factor.

i. Finally calculate the estimatation of programming hours. Karner (1993) suggests the use of 20 man/hours per unit UCP. In [Schneider and Winters 1998] is suggested the following refinement:

X = total items from F1 to F6 with scores below;

Y = total items from F7 to F8 with scores above 3;

If X + Y <= 2, use 20 like unit of man/hours;

If X + Y = 3 or X + Y = 4, use 28 like unit of man/hours;

If X + Y >= 5, should try to modify the project in order to reduce the number, because the risk of failure is relatively high;

Estimated hours = AAUCP * hours per man per UUCP (unit UCP);

Table 2. Classification of Use Cases

Use Case Type	Description	Weight
Simple	Consider up to 3 transactions with less than 5 classes of analysis	5
Medium	Consider from 4 to 7 transactions with 5 to 10 classes of analyses	10
Complex	Consider up 7 transactions with at least 10 classes of analyses	15

Font: (Medeiros 2004)

To calculate the UUCP must follow the following basic rule: UUCP = Total weight of WA + Total weight of WUC. To calculate the WA (Weight Actors) must find the sum of the weights assigned to different following the specification of Table 2; WUC (Weight of Use Cases) = Sum of weights assigned to different use cases, following the specification of Table 1.

To calculate TCF is used the following rule:

TCF = 0.6 + (0.01 x TFator), where TFator = Σ (Weight x Factor) and FA = 1.4 + (-0.03 x Efator).

Technical factors to be considered are presented in Table 3. For this it is important to define the technical factor of each item, a factor that will vary from 0 to 5, where 0 ↔ means Irrelevant and 5 ↔ Very Relevant.

To calculate the FA must be followed the following basic rule: Efator = Σ (Weight x Factor), where FA = 1.4 + (-0.03 x Efator).

The environmental items to be considered are presented in Table 4.

For this it is important to define the technical factor of each item, a factor that will vary

from 0 to 5, where 0 ↔ Irrelevant and 5 ↔ Very Relevant.

To calculate the Use Case Points (UCP) of the proposed application must then perform the following calculation: UCP = UUCP x TCF x FA.

Actors and use cases are found using the requirements of customers and potential users as vital information. As they are discovered, the use cases and actors should be briefly described. Before describing the use cases in detail, the model of use cases should be reviewed by the client to verify that all use cases and actors are found, and that together they can provide what the customer wants.

Table 3. Technical Complexity Factors x Weigth

Description	Weight
Distributed Systems	2,0
Aplication Performance	1,0
Final User Efficiency (on line)	1,0
Complex Internal Processing	1,0
Code Reusabillity in another applications	1,0
Instalation Facility	0,5
Usability (operational facility)	0,5
Portability	2,0
Maintenance Facility	1,0
Concurrence	1,0
Security Specials Features	1,0
Direct Acess by outsourceds professionals	1,0
Special training facilities	1,0

Font: (Medeiros 2004)

In an iterative development environment, you select a subset of use cases to be detailed in each iteration.

When the actors and use cases are found, the flow of events of each use case is described in detail. These descriptions show how the system interacts with the actors and the system runs in each individual case.

Finally, the model of use cases complete (including descriptions of use case) is reviewed, and the developers and customers use to agree on what the system should do.

Use Case Points Diagram

Diagram of Use Case Points aims to aid communication between analysts and the client. A Use Case Diagram describes a scenario that shows the features of the system from the viewpoint of the user. The customer should see the Use Case Diagram the main features of your system [UFCG 2011].

The Use Case diagram is represented by:

Actor - An actor is represented by a doll and a label with the name of the actor. An actor is a system user, which can be a human user or another computer system;

Use case - A use case is represented by an ellipse and a label with the name of the use case. A use case defines a major function of the system. The implication is that a function can be structured in other functions and thus, a use case can be structured;

Relationships - help to describe use cases; occurs between an actor and a use case;

Association - Defines system functionality in terms of user;

Among Actors - Generalization (Use cases of B are also instances of use of A), A has its own use cases.

Between use cases - Include (A relationship Include of an use case A to use case B indicates that B is essential for the behavior of A. It can be said also that B is_part_of A), Extend (A relationship Extend of a use case B to a use case indicates that the use case B may be added to describe the behavior of A (not essential). The extension is inserted into an extension point of use case A.

Table 4. Factors of Environmental Complexity

Factor	Description	Weight
F1	Familiarity with the process of software development	1,5
F2	Experience in application	0,5
F3	Experience with OO, in language and technique development	1,0
F4	Leader's ability to analyze	0,5
F5	Motivation	1,0
F6	Stable requirements	2,0
F7	Workers with partial dedication	-1,0
F8	Difficulty of programming language	-1,0

Font: (Medeiros 2004)

Extension Point in a use case is an indication that other use cases may be added to it. When the use case is invoked, it checks whether or not their extensions should be invoked.

These relationships can be:

- associations between actors and use cases;
- generalizations between the actors;
- generalizations, extends, and includes among the use cases.
-

PROBLEMS TO MAKE AN ANALYSIS OF REQUIREMENTS

This paper presents a proposal to use the Use Case Points in an institution located in São Paulo. The Lawyers Association of São Paulo is an example in defense of Class Counsel. Over the years the institution has followed the evolution of times and always responded to the demands of law practitioners. It carries a large amount of courses, events and activities and requires a system to monitor these events and issue certificates to participants and associates. Currently this organization does not affect control over their processes. Want to map all your processes and create a system that can improve their services and improve quality to better serve its customers [Rossier 2011].

A system is composed of tasks to be developed and tasks that will meet the needs of a client [Heimberg 2005]. How to identify such tasks? How to estimate how long one, if not all, assignments are due? How to measure each task considering the level of complexity? And how to make this measurement regarding who will perform the tasks, but considering the efficiency of project and level of experience of professionals?

PROPOSAL FOR A SOLUTION USING CASE POINTS

For solving this problem was proposed a Use Class Diagram, presented at a figure 1. This diagram was prepared after an interview hold with

stakeholders, made by requirements analysts and that have defined the following needs:

- Control of all areas of the courses and events available to conduct activities;
- Registration of all available resources in the association to carry out activities;
- Registrations of the events and courses by the organizer, allocating space main event and several members of support staff;
- Registration of activities by the organizer, scheduling for space and resources needed to achieve the same;
- Registration of the participants in various activities and control the payment of fees receptive. The forms of payment are cash with an identified deposit, debit card, credit card or check. If by check, to verify the status of the participant system will automatically check the Serasa consultation, which is the Brazilian institution of centralized payment information at rates of Brazilian institutions;
- Control input of participants in activities;
- Issuance of Digital Certificates, which will be automatically sent to the student by e-mail;

This system will help control the courses are given by the Association which are separated by three methods:

- Classroom: Course performed in the Association where students attend classes in person.
- Remote Classroom: Course is also performed in the Association but is transmitted to the other states of Brazil by satellite.
- Via the Internet: The student can attend the course in real time on the internet site of the Association.

To register for any of class is necessary to effect the registration site or in person by the Association. The modality Remote Classroom is done in partnership with the association from other states where it has a place for students to attend the course. The system will track student records and registration, registration of teachers, provision of venues for events, control of costs and receipts, course material and registration of partners. With this control all that employees will spend less time on repetitive tasks and can help develop new work in the quality and growth of new courses in addition to more reliable results to be presented.

It is important to note two functionalities suggested by stakeholders that were not included in Use Case: Register Available Areas and Register Available Resources. This is because they are functionalities who update databases and who do not have specific business rules.

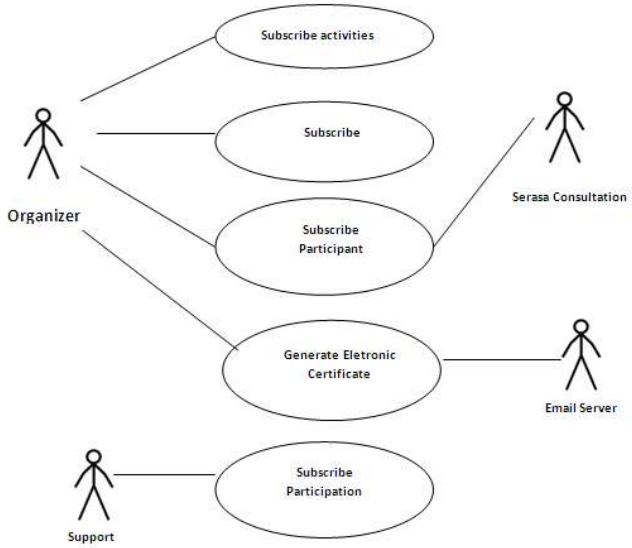


Figure 1. Use Class Diagram suggested

Based on the requirements and the exposed content was possible calculate the following Use Case:

UUCP = WA + WUC = 18+25 = 43 points

TCF = 0,6 + (0,01 x 28) = 0,88

FA = 1,4 + (-0,03 x 22) = 0,74

UCP = UUCP x TCF x FA = 43 x 0,88 x 0,74 = 28 points

For a considered FH = 20 was obtained:

Estimation of hours = 28 x 20 = 560 man/hours.

Estimation of days = 560 / 8 hours = 70 days;

In a team of 5 professionals: 70 / 5 = 14 days as prescribed period. It can be considered an error of about 10% in the period stipulated.

FINALS CONSIDERATIONS

With the Use Case Point Analysis can troubleshoot the Lawyer Association of São Paulo, as well as raising all their requirements and map all your processes also allows greater control on the cost estimate and development schedule software. The analysis helped to define the limits of the system searching for what was really needed, always focusing on customer needs. Upon completion of the development was documentation step by step, if there is a need to consult it for maintenance. Thus the institution has a system that meets almost all its needs and was able to implement it in more precise time.

Acknowledgements

The authors wish to thank the Lawyers Association of São Paulo for the provision of informational material.

REFERENCES

AGUIAR M. (2011) “Function Points or Use Case Points ? How to Estimate Object-Oriented Design”(in portuguese). Published by Developer’s Magazine. Available in BFPUG – Brazilian Function Point Users Group - http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos_de_Funcao_ou_Pontos_por_Caso_de_Uso.pdf. Access in 08/04/2011.

BFPUG – Brazilian Function Point Users Group (2011). “How Functin Point are count”(in portuguese). Available in http://www.bfpug.com.br/Como Contar PF.htm. Access in 29/04/2011.

BITTNER, K. and SPENCE, I. (2002) “Use-Case Modeling”. Addison-Wesley, Boston, MA, USA.

Camargo, L. (2010) “Estimation by Use Case Points”(in portuguese). Available in: www.luizcamargo.com.br/arquivos/EstimativaPorPontosDeCasosDeUso.pdf. Acess on Apr 1st, 2011.

DIAS, R. (2003) “Function Point Analysis: A Technique for Design of Information Systems”(in Portuguese). System Information Eletronic Newspaper (in portuguese). Vol. 2, Nº 2: 2003. Available in:

http://revistas.facecla.com.br/index.php/reinfo/article/view/134/28. Access in 07/04/2011.

FAN, W.; XIAOHU, Y.; XIAOCHUN, Z.; LU C. (2009) “Extended Use Case Points Method for Software Cost Estimation”. Proceedings of 2009 International Conference on Computational Intelligence and Software Engineering (CiSE 2009). Wuhan, China, IEEE Computer Society.

HEIMBERG, V. (2005) “Case Study of Metrics Application of Use Case Points in a Software Company” (in portuguese). Furb, Santa Catarina, Brazil. Available in: <www.inf.furb.br/seminco/2005/artigos/130-vf.pdf> . Acess on Apr 1st, 2011.

JAIR, C. L. (2000) “Analyses and Requirements Specification” (in portuguese). Rio Grande do Norte, Brazil. Available in: <http://www.dimap.ufrn.br/ ~jair/ES/c4.html> , Acess in Apr 1st, 2011.

KARNER, G. (1993) “Use Case Points: resource estimations for Objectory projects”. Objective System SF AB (copyright owned by Rational/IBM).

KOTONYA, G. and SOMMERVILLE, I. (1998) “Requirements Engineering: Processes and Techniques”. John Willey & Sons, England.

LEFFINGWELL, D. and WIDRIG, D. (2003) “Managing Software Requirements”. 2nd Edition, Addison-Wesley, Boston, MA, USA.

MEDEIROS, E. (2004) “Developing Software with UML 2.0” (in Portuguese), Ed. Makron Books, São Paulo, Brazil.

ROSSIER, E. (2011) “Lawyer Association of São Paulo”; (Associação dos Advogados de São Paulo, in portuguese). AASP, São Paulo, Brazil.
Schneider, G. and Winters, J.P, (1998) “Applying Use Cases: A Practical”. Addison Wesley, California, USA.

UFCG – Use Cases. (2011). “ Use Cases Diagrams” . Available in: http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm. Access in 15/10/2011.

VAZQUES, C. E. et.al (2008) “Analyses of Function Points: Measurement, Estimation and Projects Management” (in Portuguese). 9ª Edition, Erica Editor, São Paulo, Brazil.