

Causal analysis procedure focused on small software development organizations

Procedimiento de análisis causal enfocado en pequeñas organizaciones desarrolladoras de software

Procedimento de análise causal enfocado nas pequenas organizações desenvolvedoras de software

Fecha de recepción: 21 de octubre de 2016

Fecha de aceptación: 5 de junio de 2017

Kelly Yohanna Zúñiga-Silva*
Carlos Alberto Ardila-Albarracín**
Francisco José Pino-Correa***

Abstract

Very Small Entities (VSE) dedicated to software development lack of enough resources to adopt causal analysis practices, defined in models as CMMI, due to their complexity and costs. It is important to detect the generated defects in the development process, and to make a systematic analysis aimed at determining its causes. However, identifying those root causes is an arduous task, and failing to do so leads to wrong decisions that either fail to solve the problem or even make it worse. On this basis, this paper proposes a causal analysis procedure focused on small organizations PAC-DS (according to its initials in Spanish), which includes activities aimed at identifying the defects causes. After its evaluation in a preliminary case study, the utility of the procedure was evidenced.

Keywords: Causal Analysis; Software Engineering; Software Process Improvement; Small Software Development Organizations.

Resumen

Las pequeñas organizaciones desarrolladoras de software no cuentan con recursos suficientes para adoptar prácticas de análisis causal, definidas en modelos como CMMI, debido a la complejidad y el costo de estas. Es importante detectar los defectos generados en el proceso de desarrollo y hacer un análisis sistemático para determinar sus causas; pero identificar esas causas principales es una labor difícil, y no lograrlo lleva a tomar decisiones erróneas que no resuelven la problemática presentada o, incluso, la empeoran. A causa de lo anterior, este artículo propone un procedimiento de análisis causal enfocado en pequeñas organizaciones (PAC-DS) que incorpora actividades para identificar las causas de los defectos. Tras su evaluación en un caso de estudio preliminar, se evidenció la utilidad del procedimiento.

Palabras clave: Análisis causal; Ingeniería de Software; Mejora de procesos software; Pequeñas organizaciones de desarrollo software.

* Universidad del Cauca (Popayán-Cauca, Colombia). kyzuniga@unicauca.edu.co. ORCID: 0000-0002-5181-1327.

** M. Sc. Universidad del Cauca (Popayán-Cauca, Colombia). cardila@unicauca.edu.co.

*** Ph. D. Universidad del Cauca (Popayán-Cauca, Colombia). fjpino@unicauca.edu.co.

Resumo

As pequenas organizações desenvolvedoras de software não contam com recursos suficientes para adotar práticas de análise causal, definidas em modelos como CMMI, devido à complexidade e o custo que apresentam. É importante detectar os defeitos gerados no processo de desenvolvimento e fazer uma análise sistemática para determinar suas causas; porém identificar essas causas principais é um labor difícil, e não lográ-lo leva a tomar decisões errôneas que não resolvem a problemática apresentada ou, inclusive, pioram-na. Consequentemente, este artigo propõe um procedimento de análise causal focado em pequenas organizações (PAC-DS) que incorpora atividades para identificar as causas dos defeitos. Após sua avaliação em um caso de estudo preliminar, evidenciou-se a utilidade do procedimento.

Palavras chave: Análise causal; Engenharia de Software; Melhora de processos software; Pequenas organizações de desenvolvimento software.

I. INTRODUCTION

Czibula *et al.* [1] argued that software quality is of paramount importance in the field of software engineering, and also stated that the quality of the product is directly related to the absence of defects. When these defects are not detected or when they are detected late, there are consequences such as delays in delivery dates, inconvenience to the customer, and increased cost and effort; additionally, significant efforts may be required to correct or find those defects later in software development [2]. Once the defect is recognized, it is important to determine its causes by means of an analysis. Thus, software developers look for ways to identify the causes of problems, although they are not always identified [3].

In concordance, causal analysis helps improving processes in software development organizations, because it contributes to identify the causes that generate defects during the software life cycle. It is also possible to find opportunities for improvement, and implement actions to reduce the continuous manifestation of the same type of defect in future projects [4]. Also, causal analysis is a low-cost method [5], in fact, Kalinowski *et al.* [6] showed that the investment can vary between 0.5 % and 1.5 % of the total cost of the project, therefore, it is feasible to recover the invested money and decrease the defects rate by more than 50 %. It should be noted that the early detection of defects is beneficial, since timely treatment reduces the delay in the execution of the project [7].

Arreche and Matalonga [11] presented a set of tools for causal analysis. Among the used techniques are Cause-Effect Diagrams, Mind Maps, Systematic Thinking, Root Cause Analysis, and Radial Graphics Analysis. In addition, Lehtinen *et al.* [12] developed a tool to help conducting causal analysis in distributed software development groups; this tool is characterized by real-time feedback, as well as by the provision of functions for creating Ishikawa diagrams.

From another perspective, the paper presented by Card [5] showed a method of causal analysis that was evaluated in two organizations highlighting the effectiveness of the method, due to the fact that the lowest defect rate was presented among similar projects previously executed.

Following this line, Lehtinen *et al.* [13] proposed a light method of root cause analysis, which differs from other proposals in its approach, focusing on a group and thus simultaneously treating a greater number of problems, reducing the effort required in the initial stage of the causal analysis. In this context, Jalote *et al.* [14] presented a method focused on preventing defects in software development projects; this method manages an iterative and incremental model, whose purpose is to analyze the defects in iteration, in order to prevent these from continue happening.

On the other hand, Nelms [15] showed a method of causal analysis that considers people to be the main cause of problems, which is why he guided his method toward people, to identify their role related to things that are not going well. In addition, Jabrouni *et al.* [16] introduced a feedback experience approach using root cause analysis through the “Why Technique” because of its efficiency and ease use, allowing us to reach the source of the problem. Finally, Honda [17] showed a root-cause analysis technique that uses a tree-shaped graph with the causes that are generating defects, while five iterations are made from the “5 Why Technique”.

From the analyzed documents, we were able to establish that a causal analysis procedure was not available, according to the adaptation to the needs and characteristics of the VSEs. Therefore, we propose a causal analysis procedure that focuses on small software development organizations to incorporate practices in this area, and to execute them in a systematic way.

Considering the above, this article proposes a causal analysis procedure focused on small software development organizations (PAC-DS), to guide this type of organizations, also known as VSEs (Very Small Entities) in the execution of the causal analysis with templates, activities, and techniques. The procedure is focused on this kind of companies because they are the majority in the software industry [8]. The procedure was evaluated through a case study, which is recognized in terms of application and usefulness for the development of the project.

In addition to the introduction, the second section of this article explains the method used to create the procedure; the third section shows the activities, tasks, roles, and work products; the fourth section shows the case study, and finally, the conclusions are presented.

II. METHODOLOGY

To execute the proposed procedure, the Action Research (AR) method was implemented, according to the adaptations made by Pino *et al.* [9]. The proposal involves a research cycle and a problem-solving cycle (Fig. 1).

1) Research cycle. It starts from an initial investigation, in which conceptual, methodological, and technical problems are approached. The conceptual research

allows to identify the theoretical framework and the tasks related to the causal analysis. BPMN [10] was used in this cycle for modeling.

2) Problem solving cycle. A case study was carried out to evaluate and improve the PAC-DS procedure through three activities: i) Diagnosis, in which we designed the case study and prepared for data collection; ii) Action, in which we collected evidence; and iii) Reflection, in which we analyzed the collected data.

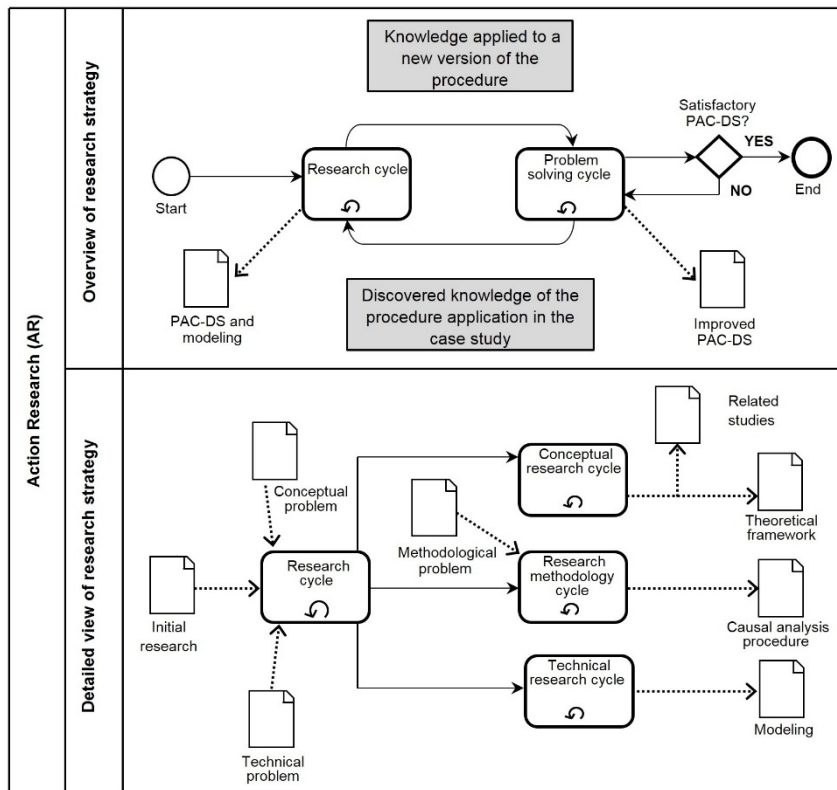


FIG. 1. Research Strategy.

III. RESULTS

A. PAC-DS overview

The activities suggested a set of techniques that support their implementation, and that have been reviewed in the literature. In addition, the templates were generated by collecting the information of each activity. Due to

space restrictions, here we only present the procedure, however, the other components of the PAC-DS can be consulted in the study by Zúñiga [18].

B. PAC-DS general description

The PAC-DS is described in terms of purposes, objectives, activities, roles, and work products (Fig. 2).

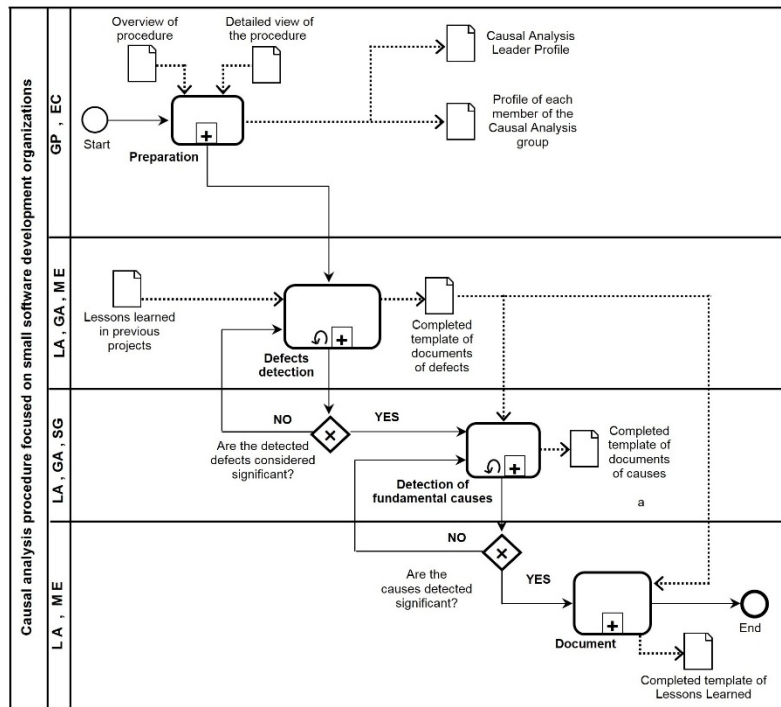


FIG. 2. Activities and products of PAC-DS.

1) Purpose: The purpose is to detect the causes of defects, in order to improve the quality of the product.

2) Goals: The objective is to propose a procedure of causal analysis able to guide the VSEs, in order to identify the causes of defects and establish the appropriate techniques to apply causal analysis, taking into account the reference of international norms and models.

3) Activities: The defined activities were preparation, detection of defects, detection of fundamental causes, and documentation. For each activity, a set of tasks was defined.

4) Roles: A person was selected as responsible for each task according to his or her competences. The roles involved in the procedure are Causal Analysis Leader (LA), Causal Analysis Group (GA), Project Manager (GP), all team members (ME), Training Officer (EC), and person asking for the modification (SC). The LA must be able to build reports and identify defects from interviews, and must have good communication skills, among others. The GA members need interpersonal skills, and must be able to express difficulties.

5) Work products: The GA reports the findings when executing the activities. The defined templates are defect collection, major impact defects, defects classification, document defects, found causes, fundamental causes, recommendations, changes requests, monitoring recommendations, document of causes, and lessons learned.

C. Description of activities

1) Preparation activity: Its purpose is to form the group responsible for executing the procedure. First, the leader (LA) and the Causal Analysis Group (maximum 6 members) should be selected, and a PAC-DS training should be conducted (when applied for the first time), in order to clarify the objective of the procedure and its structure. The tasks related to this activity are:

- a. Election of the Causal Analysis Leader
- b. Election of the members of the Causal Analysis Group
- c. Training on the structure and components of the procedure

d. Techniques training

2) Detection of defects activity: Its purpose is to discuss, analyze, and classify defects. The tasks related to this activity are:

- a. Defect identification. The GA discusses the defects arising in the development of the project by means of techniques such as Group Meetings and Affinity Diagrams.
- b. Sample defects determination. Defects with the greatest impact are selected. Defects are prioritized using techniques such as Pareto Diagrams or Modal Failure and Effects Analysis (FMEA).
- c. Defect classification. Defects directly related to the software product, and those more related to the development of the project are identified. For this, orthogonal classification scheme or Ishikawa cause classification scheme is used.

3) Detection of fundamental causes activity: The purpose of this activity is to identify, analyze, and classify the causes of the defects detected in the previous activity, and to provide guidelines to eliminate the causes of greater impact. The tasks related to this activity are:

- a. Identification of the causes. In order to perform this task, a meeting must be held with all members of the Causal Analysis Group, in order to classify each cause according to the categories proposed by Ishikawa.
- b. Analysis of the causes. The GA determines the causes of the greatest number of defects and those with the greatest impact on the development of the project. This should be determined in a group meeting.
- c. Development of recommendations. The LA writes a report with proposals to eliminate the causes of greatest impact.

d. Initiation of the requested change. The person responsible for making a change notifies the personnel who must performed it; they must have interacted with the causal analysis group, and have management or software engineering skills.

e. Disclose information to relevant members. The LA sends a communication to the members in charge of implementing the changes suggested in the recommendations.

f. Monitoring the recommendations. The LA verifies the recommendations to be implemented.

4) Documenting activity: The purpose of this task is to record the lessons learned during the project in a final report, and to store the documents obtained when applying the procedure in the organization's repository. These activities are described in detail in Zúñiga [18].

IV. CASE STUDY

A. Design

We designed the study based on Yin *et al.* [19], which is a simple holistic study that proposes a procedure applied to a working group of software development that has characteristics similar to the VSEs. Table 1 shows the main research question (PP), and the additional questions (PA) that guided the case study.

We used a development project called "Daily Activity Management Systems" as the unit of the analysis. The subjects for this research were part of a group of systems engineering students in ninth semester at the University of Cauca. The objective of the study was to propose the procedure. The measures used to answer the research questions were the following: 1) the effort made to implement the activities proposed in the procedure, 2) the number of found causes, 3) the people's perception involved in the project.

TABLE 1
CASE STUDY RESEARCH QUESTIONS

PP	Is the proposed procedure adequate for conducting causal analysis in small software development organizations?
PA1	Are the suggested techniques adequate for causal analysis in small software development organizations?
PA2	Does the procedure for conducting causal analysis allow identifying the causes of defects presented during the execution of software projects?

B. Field procedure and data collection

The field procedure, which consisted of four activities (Fig. 3), was fundamental to execute the PAC-DS, since it allows to know the group and achieve results through adapting activities and tasks according to the conditions and characteristics of the project. The “Preparation” activity was not taken into account because the leader was already in the group, and all the team members formed the Causal Analysis Group. It should be clarified that at that time, the training techniques task had not been defined, because it was included as an improvement to the procedure, once the case study was completed.

C. Intervention

The procedure was performed in four phases, in which the defects and their root causes were detected. For each activity, the time invested, the causes found, and the established recommendations were recorded, in order to determine the effort required in the proposed activities and the number of found causes. The intervention lasted 4 months, and the members of the project group played roles as analysts, developers, test managers, and manager within the process defined for the project. The efforts per phase are shown in Table 2.

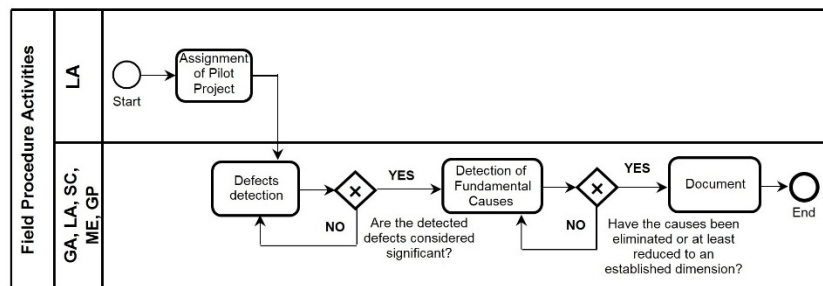


FIG. 3. Field procedure governing the case study activities.

TABLE 2
EFFORT INVOLVED IN THE PROCEDURE

Activity	Effort (person-hours)			
	Phase 1	Phase 2	Phase 3	Phase 4
Detection of defects	2.65	3.94	2.85	5.95
Detection of fundamental causes	13.42	8.76	6.17	10.54
Document (generation of the final report)	0.00	0.00	0.00	1.15
TOTAL	16.07	12.70	9.02	17.64
Procedure Total Effort: 55.43				

Among the causes found, we can mention recurrent lack of communication, and lack of motivation and responsibility in each phase; this may be due to implementing the project in an academic environment, and the fact that the team members had no experience in this type of projects. The latter cause represents a special condition that may limit the generalizability of the results; this is why we insist on the classification of this study as a preliminary scope. Table 3 lists some of the identified causes.

The perception of those involved was obtained through an interview, where the interviewee had the option to mark the answer Yes or No with a space for comments. The questions were the following: 1) Did the procedure permit the demonstration of defects? 2) Did the procedure permit to demonstrate the causes of defects? 3) Was the Ishikawa Technique easy to use? 4) Was the Diagrams Affinity Technique easy to use? 5) Are there any positive aspect and possible improvements related to the procedure? 6) Was the PAC-DS Causal Analysis Procedure useful for the project development? 7) Was the PAC-DS procedure easy to implement?

TABLE 3
LIST OF FOUND CAUSES ACCORDING TO PHASE

Phase	Cause
1	Absence of a means to agree on meeting dates Lack of motivation Lack of time Excessive academic load Nonexistent communication plans Deadlines for undefined deliveries Lack of punctuality when attending meetings Lack of responsibility
2	Lack of definition of deliverables at the end of meetings Lack of motivation Lack of responsibility Poor change management in the database Lack of knowledge in the framework Deficient software product version control Insufficient number of developers Poor planning in effort allocation over time Poor planning of software testing
3	Lack of knowledge in Ajax Absence of coding standards Poor change management in the database Health problems of one of the developers Document of requirements specified at very high level No detailed use cases were made Poor planning in effort allocation over time The Ajax tool was not used for all the functionalities Lack of knowledge related to the MoProSoft quality model
4	Poor planning of software testing Poor change management in the database Incorrect estimation of time required for each task Health problems of two of the programmers

D. Reflection

The effort required to implement the procedure activities was 55.43 person-hours (over a period of 4 months) in relation to 1092.11 person-hours employed for the whole project, representing 5.07 % of the total, indicating that the effort to apply the procedure can be considered low. Taking into account the close relationship between effort and costs, and according to the Project Management Institute (PMI) parameters, this result is evaluated and classified as “low” as long as it has a magnitude less than 10 % [20].

When applying the PAC-DS procedure, it was possible to reduce the defects; this was verified when counting them as registered in the “Template of Defects Collection”, and it was noticed that the defects diminished because it was possible to identify and to counteract a considerable quantity of the causes that generated them, improving the quality of the software product. The defect management process did not change over the four months, and was managed according to the tasks defined in the “Defect Detection” activity.

On the other hand, it is important to emphasize the benefits obtained through the meetings. As a result, it was possible to eliminate causes such as those related to communication, for which recommended mechanisms and clarifications were established, and made respectively on the valid means and how they should be used.

About the perception of the PAC-DS procedure by the people involved in the project (7 in total), the interview revealed the following:

- Was the PAC-DS procedure useful and easy to execute? Yes: 7, No: 0.
- Are Affinity Diagrams easy to use? Yes: 7, No: 0.
- Did the PAC-DS procedure allow noting defects? Yes: 5, No: 2. The procedure can help detecting some defects and that is why this question was included.
- Was Ishikawa’s Technique easy to use? Yes: 4, No: 3. The observations indicated that there was not further deepening in its construction and study. For

this reason, a task related to Technique Training was included in the “Preparation” activity.

The aspects to be improved suggested by the interviewees is another element taken into account. In the procedure (“Fundamental Causes Detection” activity), it was included the task Starting a Change Request (either in the definition of the process, tools, methods, work products, assigned roles, implemented methodology or any other element identified as a source of defects) to formalize the necessary changes to reduce or eliminate the causes of defects.

Based on the obtained results, in relation to the utility and ease of the procedure implementation and on the observations expressed by the participants, it can be said that the procedure is highly likely to be useful in small organizations.

V. CONCLUSIONS

The PAC-DS procedure was created to solve the need of small software development organizations to adopt and implement practices related to causal analysis. Hosting these practices would help identifying causes that negatively impact software development projects leading to product defects.

Therefore, the PAC-DS procedure was proposed because it provides a detailed description of each activity and task, and has diagrams that show the flow to follow, the roles involved, and the work products. It also includes templates to guide each activity in the required information registration.

The accomplished work helped us to verify the importance of performing the causal analysis within the organizations, which demonstrated the relevance of the PAC-DS procedure implementation in the reduction of defects, which positively impacts the product quality. In this sense, the procedure supports the identification of causes generating the defects, as well as the corrective execution to eliminate or reduce those defects.

As a future work, the procedure could undergo improvements as long as the international referents on which the research has been developed are updated. The procedure is subject to changes in activities, tasks, work products, and roles.

ACKNOWLEDGEMENTS

This article is the result of the degree work “Causal Analysis Procedure Focused on Small Software Development Organizations” carried out within the Systems Engineering Program at the University of Cauca. Carlos A. Ardila Albarracín and Francisco J. Pino are grateful to the University of Cauca, where they work as Associate and Titular Professors, respectively.

REFERENCES

- [1] G. Czibula, Z. Marian, and I. G. Czibula, “Software defect prediction using relational association rule mining,” *Inform. Sci. an Int. J.*, vol. 264, pp. 260-278, 2014.
- [2] C.-P. Chang, and C.-P. Chu, “Defect prevention in software processes: An action-based approach,” *J. of Syst. and Softw.*, vol. 80 (4), pp. 559-570, Apr. 2007. DOI: <http://doi.org/10.1016/j.jss.2006.09.009>.
- [3] M. G. S. Gonçalves, “MiniDMAIC: An Approach for Causal Analysis and Resolution in Software Development Projects,” in *Advances in Comput. and Inform. Sci. and Eng.*, 2008, pp. 166-171. DOI: http://doi.org/10.1007/978-1-4020-8741-7_30.
- [4] M. Kalinowski, G. H. Travassos, and D. N. Card, “Towards a Defect Prevention Based Process Improvement Approach,” in *34th Euromicro Conf. Softw. Eng. and Advanced Applicat.*, 2008, pp. 199-206. DOI: <http://doi.org/10.1109/SEAA.2008.47>.
- [5] D. N. Card, “Learning from Our Mistakes with Defect Causal Analysis,” *IEEE Softw.*, vol. 15 (1), pp. 56-63, 1998. DOI: <http://doi.org/10.1109/52.646883>.
- [6] M. Kalinowski, D. N. Card, and G. H. Travassos, “Evidence-Based Guidelines to Defect Causal Analysis,” *IEEE Softw.*, vol. 29 (4), pp. 16-18, Jul. 2012. DOI: <http://doi.org/10.1109/MS.2012.72>.
- [7] M. Leszak, D. E. Perry, and D. Stoll, “A case study in root cause defect analysis,” in *Proc. 2000 Int. Conf. on Softw. Eng. (ICSE)*, 2000, pp. 428-437. DOI: <http://doi.org/10.1145/337180.337232>.
- [8] F. J. Pino, F. García, and M. Piattini, “Software process improvement in small and medium software enterprises: a systematic review,” *Softw. Quality J.*, vol. 16 (2), pp. 237-261, 2008. DOI: <http://doi.org/10.1007/s11219-007-9038-z>.
- [9] F. J. Pino, M. Piattini, and G. H. Travassos, “Managing and developing distributed research projects in software engineering by means of action-research,” *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 68, pp. 61-74, 2013.
- [10] *Business Process Model and Notation (BPMN)*, Object Management Group, Inc. (OMG), 2011. Available: <http://www.omg.org/spec/BPMN/2.0>.
- [11] S. Arreche, and S. Matalonga, “Tools for defect causal analysis: A systematic literature review,” in *7th Iberian Conference on Information Systems and Technologies (CISTI)*, 2012, pp. 1-7.
- [12] T. O. A. Lehtinen, *et al.*, “A tool supporting root cause analysis for synchronous retrospectives in distributed software teams,” *Information and Software Technology*, vol. 56 (4), pp. 408-437, Apr. 2014. DOI: <http://doi.org/10.1016/j.infsof.2014.01.004>.
- [13] T. O. A. Lehtinen, M. V. Mäntylä, and J. Vanhanen, “Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies,” *Information and Software Technology*, vol. 53 (10), pp. 1045-1061, Oct. 2011. DOI: <http://doi.org/10.1016/j.infsof.2011.05.005>.
- [14] P. Jalote, and N. Agarwal, “Using defect analysis feedback for improving quality and productivity in iterative software development,” in *Proc. 3rd International Conference on Information and Communications Technology*, 2007, pp. 703-713.
- [15] C. R. Nelms, “The problem with root cause analysis,” in *8th IEEE Human Factors and Power Plants and HPRCT 13th Annual Meeting*, Aug. 2007, pp. 253-258. DOI: <http://doi.org/10.1109/HFPP.2007.4413215>.
- [16] H. Jabrouni, *et al.*, “Continuous improvement through knowledge-guided analysis in experience feedback,” *Engineering Applications of Artificial Intelligence*, vol. 24 (8), pp. 1419-1431, Dec. 2011. DOI: <http://doi.org/10.1016/j.engappai.2011.02.015>.
- [17] N. Honda, and S. Yamada, “‘Defect Root-Cause Analysis and 1+n Procedure’ technique to improve software quality,” *International Journal of System Assurance Engineering and Management*, vol. 3 (2), pp. 111-121, Jun. 2012. DOI: <http://doi.org/10.1007/s13198-012-0118-5>.
- [18] K. Y. Zúñiga, “Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software,” Thesis, Universidad del Cauca, Popayán, Colombia, 2015.
- [19] R. K. Yin, *Case study research: Design and methods*, 5th ed. Newbury Park CA: Sage publications, 2014.
- [20] Project Management Institute, “Project risk management,” in *A guide to the project management body of knowledge (PMBOK Guide)*, 4th ed. Newtown Square PA: Project Management Institute, Inc., 2008, pp. 281.