Software development process that supports the industry applications 4.0: a proposal for improvement from the experimental software engineering approach

Proceso de desarrollo de software que soporta las aplicaciones de la industria 4.0: una propuesta de mejora desde el enfoque de la ingeniería de software experimental

DOI: https://dx.doi.org/10.17981/ingecuc.20.2.2024.11
Artículo de Investigación Científica. Fecha de Recepción: 04/03/2022, Fecha de Aceptación: 17/05/2023.

Luis Eduardo Peláez-Valencia https://orcid.org/0000-0002-4836-8336
Universidad Tecnológica de Pereira - Universidad Libre. Pereira, (Colombia)
luiseduardo.pelaez@gmail.com

Fabio Alberto Vargas-Agudelo https://orcid.org/0000-0002-6921-4918
Tecnológico de Antioquia. Medellín, (Colombia)
fvargas@tdea.edu.co

Miguel Ángel Cohuo-Ávila https://orcid.org/0000-0002-4365-187X
Tecnológico Nacional de México, Campeche, (México)
macohuo@itescam.edu.mx

Iván Andrés Delgado-González https://orcid.org/0000-0002-0417-2649
Fundación Universitaria Juan de Castellanos. Tunja, (Colombia)
idelgado@jdc.edu.co

Juan Luis Arias-Vargas https://orcid.org/0000-0002-7997-1891
Universidad Católica. Pereira, (Colombia)
Juan.arias@ucp.edu.co

Helmuth Trefftz https://orcid.org/0000-0003-2441-506X
Universidad EAFIT. Medellín, (Colombia)
htrefftz@eafit.edu.co

To cite this paper

L. Peláez-Valencia, F. Vargas-Agudelo, M. Cohuo-Ávila, I. Delgado-González, J. Arias-Vargas & H. Trefftz, "Software Development Process Supporting Industry 4.0 Applications: An Improvement Proposal From The Experimental Software Engineering Perspective," INGE CUC, vol. 20, no. 2, 2024. DOI: *https://dx.doi.org/10.17981/ingecuc.20.2.2024.11*

Abstract

Introduction: The purpose of Experimental Software Engineering is to ensure the quality of the development process in search of better technological products.
Objective: To design a model for the quality assurance of functional and non-functional requirements in software development projects. When applied by companies in Pereira, Colombia, the proposed model is expected to enhance the quality of software products.
Methodology: This study adopts an explanatory and incremental approach to define the properties and characteristics of process analysis related to the quality assurance of functional and non-functional requirements in Software Projects (SP). The research is grounded in the state of heart methods for software quality assurance
Results: The Quality Assurance model for requirements was designed by evaluating the experimental process presented in this study. The project facilitated the development of new alternatives for higher quality processes in the early phases of software development, which, in turn, leads to the creation of superior software products
Conclusions: It is possible to find alternatives to improve the software development process without requiring high financial investments in specialized human talent. This is achieved by experimentation in software engineering or, in other words, experimental software engineering, through processes that are fine-tuned considering both the size of the producing organizations and the capabilities of the human talent.

Key Words

Software Engineering Experimentation; Industry 4.0; Software Quality Assurance; SQA; Software Experimentation.

Resumen

Introducción: El propósito de la Ingeniería de Software Experimental es asegurar la calidad del proceso de desarrollo en la búsqueda de mejores productos tecnológicos.
Objetivo: Diseñar un modelo para el aseguramiento de la calidad de los requisitos funcionales y no funcionales en proyectos de software. Al ser aplicado por empresas de la ciudad de Pereira, Colombia, se espera que el modelo fomente la calidad de los productos de software.
Metodología: El estudio propuesto es de carácter explicativo e incremental, y busca precisar las propiedades y características del análisis de procesos relacionados con el aseguramiento de la calidad de requerimientos funcionales y no funcionales en Software de Proyecto (SP) a partir del estado del arte del aseguramiento de la calidad de software.
Resultados: Se diseñó el modelo de Aseguramiento de Calidad de los requerimientos y se puso a prueba el proceso experimental expuesto en este trabajo. El proyecto permitió generar nuevas alternativas para procesos de mayor calidad en las fases tempranas del desarrollo de software, lo que a su vez conduce a la construcción de mejores productos de software.
Conclusiones: Es posible encontrar alternativas para mejorar el proceso de desarrollo de software sin requerir de altas inversiones financieras en talento humano especializado. Esto se logra mediante la experimentación en ingeniería de software o, en otras palabras, ingeniería de software experimental, mediante procesos que se afinan teniendo en cuenta tanto el tamaño de las organizaciones productoras como las capacidades del talento humano.

Palabras Clave

Software Engineering Experimentation; Industry 4.0; Software Quality Assurance; SQA; Software Experimentation.

I. INTRODUCTION

Industry 4.0, structured in a broad dimension by software products, will impact society by improving the development process of new products and services. This article is the result of the Research Project "Automated Model for Quality Assurance of Requirements in Software Projects" formulated and developed by Higher Education Institutions in Colombia and Mexico according to the authors' affiliations, between the years 2017 and 2019. The objective of this project is to achieve a model that impacts quality assurance in the software development process, particularly, in the requirements management phase, so that time, cost and quality may be fostered in the local industry.

As a guiding hypothesis, it was considered that if the level of quality - quality assurance mentioned above - in the requirements of software development projects is increased, consequently the efficiency in the other phases of development and, consequently, the quality of the final software product should also be improved [1]. Additionally, in the case at hand, better inputs, represented in devices and applications, for the software industry, were achieved.

All this, framed in the improvement of software processes, as a broad field of study in the discipline of Software Engineering and, as a specific field, the quality assurance of requirements elicitation.

Currently, software becomes the heart of thousands of devices that are "part of the enabling components of the industry 4.0 [2]. Considering this important role of software, it becomes of special interest for those who are part of this industry to incorporate good practices that obey high quality criteria in the improvement of software processes that are embedded in these devices and networks [3].

This report is structured as follows: a brief introduction to contextualize the theoretical foundation that accompanied the research and, subsequently, everything related to the research methodology, the project's own experimentation and the conclusions by way of discussion.

II. THEORETICAL BACKGROUND

Producing good software is very difficult, very expensive and necessary for the development of all societies [4]. For this reason, in the 1960's the concept of Software Engineering was coined as a discipline and, since then, people and companies that are dedicated to the construction of software have the goal of applying Software Engineering [5] to achieve quality processes and products. Therefore, since a special purpose of Industry 4.0 is to incorporate digital technologies and enablers to the industry [6] and many of these are represented in software, it is necessary to pay particular attention to concepts such as: Industry 4.0, Software Engineering, Quality, Software Quality Assurance (SQA) and Requirements Quality Assurance (RQA).

A. Software Engineering

The term Industry 4.0, also known as i4.0, represents the fourth industrial revolution, defined as a new level of organization and control over the entire value chain of the product life cycle [7], [8]. New technologies in the field of data acquisition, sensors technology, the Industrial Internet of Things (IIoT), storage, cloud computing and machine learning allows to develop cyber-physical production systems (CPPS), which are often referred to as Industry 4.0 [9].

Therefore, the incorporation of technologies into the industry will provide great benefits, generating profound transformations in people's lives and impacting the economic and social sectors; but will also present enormous challenges for the industry and the government. The software industry, being responsible for the support of these technologies, should be the first to improve its products and processes [7]. In this sense, and in response to this call for process improvement, experimentation in the quest to improve the software development that is part of all these technologies mentioned above, is a key element in the development of the software.

B. Quality

There are multiple definitions of Quality, some of these definitions are included, Quality is "a predictable degree of uniformity and reliability at low cost, appropriate to market needs" [10]. "Quality is compliance with requirements" [11], "Quality is adequacy for customer use" [12]. “Set of properties and characteristics of a product, process or service that confer its suitability to satisfy stated or implied needs" [13], also presented as the "degree to which a set of inherent characteristics of a product, process or service meets requirements", understood as an "established need or expectation, usually implicit or mandatory " [13].

However, in more specific terms related to software development, definitions such as: "Degree to which a set of inherent characteristics of an object complies with the requirements” [14], [15] or “Extent to which a software product satisfies stated and implied needs when used under specific conditions” [16] have been more accepted.

C. *SQA*

Software Quality Assurance, SQA, as a sub-discipline of Software Engineering [17], provides assurance that the products and processes in the project life cycle meet their specified requirements by planning, enacting and carrying out a set of activities of sufficient confidence, and indicating the quality that lies in the software [5].

In this sense, the quality of the product, together with the quality of the process, is one of the most important aspects in software development today, so much that new standards have appeared recently, such as the ISO/IEC 25000 family of standards, which provides a guide for the use of the new series of international standards called Software Product Quality Requirements and Evaluation (SQuaRE) [18].

Taking the concept to the context of Software Engineering, SQA, is understood as the way to adopt good practices in the development process (internal quality variables) and for the product (external quality variables) [17].

D. Requirements Quality Assurance

The requirements for software development represent a fundamental part in the achievement of quality software, which are studied by recognized authors in the field as [19] and [20], among others, so much so that there is a discipline called requirements engineering (RE). Likewise, authors such as [21] consider that everything related to requirements is part of an entire domain called Requirements Engineering, which is divided into Development and Management.

On the other hand, Requirements Quality Assurance (RQA) is understood as the set of tools and best practices that provide the assurance that the products and processes related to the definition of a problem to be solved with new or modified software comply with what was previously specified in order to generate the necessary confidence that leads them to be considered as high-quality [22].

Accordingly, RQA can be considered as a sub-discipline of SQA that aims, through the application of good software engineering and requirements engineering practices, to ensure the quality of software projects from the early stages of the development life cycle through proper requirements development and management.

III. METHODOLOGICAL DEVELOPMENT

The work described in this paper was focused on the inductive research method and classified in two sections. The first one, to determine the key elements of software development that the existing quality models contemplate and the way they focus on requirements management. This is achieved by documenting the state of the art available in the world literature. It is also complemented with an instrument that inquires on quality in the development process and in the software product, to the Small and Medium-Sized Enterprises (SMEs) sector dedicated to software construction [23].

The second section, focused on the construction of the requirements quality assurance model in software projects and the evaluation and validation of the requirements through a primary method to obtain empirical evidence through an experiment of the application of the model to a group of students, professionals and companies in the software industry, allowing for the comparison, with the use of some data collection instruments, of the way in which the quality assurance of requirements is currently carried out and how it would result with the application of new models [24].

Regarding the research design, it is a mixed study with a quantitative and qualitative approach, quasi-experimental, among other things, as the sample was determined by convenience due to the difficulty of accessing some elements of the population, given the secrecy and confidentiality of the information that software development companies have.

With the explanatory purpose in the experimental order [24], it is sought to specify the characteristics of the software process analysis related to the quality assurance of functional and non-functional requirements.

IV. SOFTWARE ENGINEERING EXPERIMENTATION

When there is a need to test an idea, evaluate a new or existing model in software engineering through a rigorous scientific process, the following methods are typically used: scientific, engineering, analytical and empirical [25].

Scientists in software engineering research processes have made efforts to systematically analyze the results of empirical studies in recent years and have identified the lack of sound statistical methods [26] that would allow them to have greater validity in their experimentation processes or, in their case, incorrect interpretations due to the inadequate use of statistical assumptions with the results obtained.

Therefore, while some products allow for the identification and correction of anomalies during the manufacturing process without significantly altering the final result, for the software development process it is necessary to discover, analyze, document and verify the constraints, all in the context of the requirements, i.e. in the early stages of the development process [27].

V. THE EXPERIMENT IN THE RESEARCH PROJECT

Following is the formulation and deployment of the experiment for the evaluation of an automated requirements quality assurance model to support the enhancement of the digital enabler development process in the I4.0.

A. *Scope definition*

Fig. 1, which contains the Level 0 Context Diagram of the CHAMI model, it shows how project requirements are managed from the interaction that the system performs with users and companies. This is a basic form of operation that takes place at different levels.

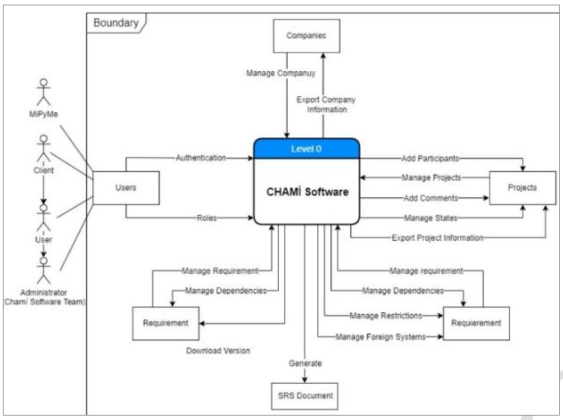


Fig. 1 Level 0 Context Diagram of the CHAMI model.

The purpose of the experiment is to verify and validate the impact generated in SMEs by having the support and use of a technological model called CHAMI [28], which aims to ensure the quality of the requirements in the early stages of software development.

For this reason, the following objective of the experiment has been proposed, using the GQM (Goal, Question, and Metric) methodology [29]: to analyze and evaluate the automated model of quality assurance of the requirements in the initial stage of elicitation, specification of requirements contemplating management models that allow to increase the quality in software development from the perspective of time and cost optimization, applying tests called Pretest (without the model), Posttest A (with the model and support instrument) and Posttest B (with the automated model).

B. *Planification*

This section describes the context of the experiment in order to explain the differences that occur when applying a requirements quality assurance model.

The purpose of the experiment was to evaluate the efficiency and effectiveness of the requirements quality assurance model in the initial phase of a software project in requirements education/discovery, specification and evaluation.

The defined perspective was related to the valuation of the requirements quality assurance model from the research point of view, i.e. to determine if there is a significant difference with its incorporation or not, in the initial phase of requirements.

Quality approach. The effect to be studied in the experiment comprises the following items:

• *Total efficiency of the requirement:* (Er) of the education/discovery of the requirement which consists of evaluating the time used to identify the requirement, by performing the total sum of time used in the process.

• *Negative effectiveness:* Consists in the identification of failures found for functional requirements (FRs) from the following classification: ambiguous, without behavior, inconsistent, or without identification of objects and actions.

• *Positive effectiveness:* consists of the correct identification for the functional requirements which contemplates 1) the problem domain contemplated when defining an object, a function or a state 2) limit or control the actions of the object, function or state and 3 the measurement of a is performed according to the formula proposed in [30], according to the following equation. Eq. (1).:

$$\text{Efectiveness} = (\text{Nconcepts} + \text{Nprocesses} + \text{Nrequirements}) / (\text{totalelements}). \quad (1)$$

For this purpose, a set of SMEs has been contemplated through a non-probabilistic sampling as mentioned in section 3.2.

Formulation of the hypothesis of the experiment. For the purpose of this article, only the hypothesis associated with the time variable will be addressed; the others are the object of study of other works resulting from the research project. In this sense, we will work as follows:

• *H0:* There is no significant difference in the decrease of time in the adaptation of a requirements quality assurance model in the initial phase of a software project

• *H1:* There is a significant difference in the decrease of time in the adaptation of a requirements quality assurance model in the initial phase of a software project.

Variable selection. For this experiment, the independent variable has been determined as the requirements quality assurance model (M) and three dependent variables which are the total time (T) required to manage a requirement in the processes of education, elicitation and specification of the requirement within a problem domain or project to be analyzed, the cost (C) of the resources needed to manage a valid or defective requirement and the quality (Q) of the resulting requirement [31].

Their values and behaviors will be attributes that depend on the resulting model. There is one qualitative independent variable and three dependent variables, one of which is qualitative and the other two quantitative, as shown in Table 1.

Subject selection. Three types of study subjects in three different roles were considered for the experiment: R1 for students of careers related to software engineering, R2 for professionals in the discipline and R3 for experts within the software industry, which were selected through the technique by convenience and randomly defined and grouped into teams of two people in a similar assignment project in the works of [32].

Table 1. Research variables

Type	Operationalization	Categories or operationalization attributes
Independent Qualitative	Quality assurance model for functional and non-functional requirements, called for practical purposes **CHAMI MODEL**.	A1: Education A2: Specification A3: Documentation A4: Evaluation A5: Automation A6: Output Models
Dependent Qualitative	**Quality Variable**. Understood as "the degree to which a system, component or process meets specified requirements and the needs or expectations of the customer or user" [32]. The variable will be assigned categories of attributes that will allow the assurance of the requirements at the beginning of the software projects.	A1: Education A2: Specification A3: Documentation A4: Evaluation A5: Automation A6: Output Models
Dependent Quantitative	**Time variable**. Understood as the time expressed in minutes that a person or a work team takes to manage a requirement from the moment it receives a written statement from the user or client until it is released for the next phase of software development to continue.	
Dependent Quantitative	**Cost Variable**. Understood as the quantitative valuation in Mexican pesos given to the work carried out to manage a requirement in the context of the software development process from the moment the written statement is received from the user or client until it is released to continue the next phase of software development.	

C. Experimental design

First, the projects or experimental units to be assigned to each subject or group of subjects were defined for random assignment. These experimental units were represented in six project versions (PR) in the form of case studies that simulated a small to medium sized project in the industry.

The experimental design is simple, with a single factor that contemplates two levels (with model and without model) and three treatments (Pretest, PosttestA and PosttestB) with participation of inter-subjects with a control block and unbalanced balancing due to the fact that treatments were applied for convenience according to the purpose of analysis intended to validate the impact of the model on the study variables and therefore a set of cases was designed, incorporating a control group (c1) as shown in Table 2.

Table 2. Cases applied to treatments

Cod_Treatment	Description
c1	No effect same experimental unit
c2	Same experimental unit pretest to posttest A
c3	Same experimental unit pretest to posttest B
c4	Change of experimental unit and of pretest to posttest A
c5	Change of experimental unit and of pretest to posttest B

The application of the treatments was constructed by randomly assigning the study subjects, the first one being a Pretest for all the subjects participating in the experiment process; an unbalanced design was considered in the Posttest A and Posttest B treatments, in each of these it is identified whether it is a student, professional or control group, the project assigned to each subject was randomized.

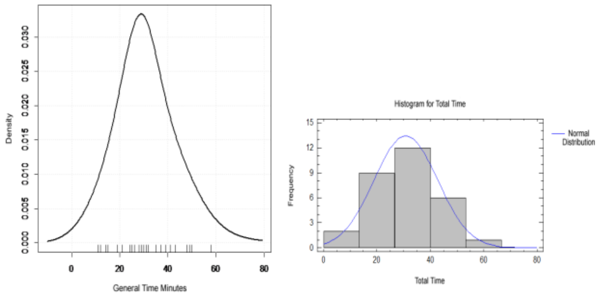
Instrumentation. Eight instruments were designed to support the entire experimentation process, collecting information related to: consent to participate, visual explanation of the experiment and its purpose, diagnosis of the knowledge of the participants in experimentation and in requirements, information related to the management of the requirements, collection of the times of each requirement in each Pretest and Posttest treatment and, finally, the instruments required for a group of experts to evaluate and validate the design of the experiment and the results obtained in the different treatments.

Evaluation of validity. According to the way the sample was selected, using a non-probabilistic sampling by convenience, it is not possible to perform an external validation, in other words, to make a statistical inference; However, the results of the experiments applied to heterogeneously formed control groups allow us to verify the internal validity, since there is statistical evidence that shows a significant difference between the times measured by the research, as shown in Table 3 and Table 4 and Figure 4, which allows an extrapolation to the study population due to the similarity of the characteristics of the projects that were selected for the experiment with those that MIPYMES normally work with.

D. *Operation and data recollection*

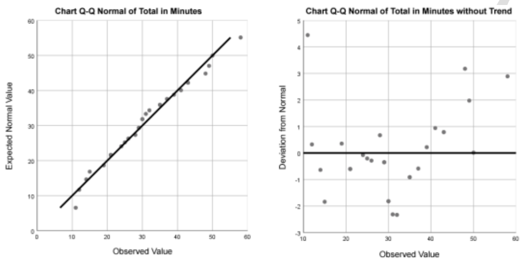
1. **Preparation.** In order to carry out the experiment, SMEs from the coffee axis, independent professionals with experience in software development and undergraduate students, in their last semester from three Higher Education Institutions that have programs in which software engineering is present, were invited. Finally, a group of 30 participants organized into 15 teams was formed.
2. **Execution**. The experimentation process was carried out in two stages, the first one called pretest A, where each team is assigned a software development project, the second stage called Posttest A and B, where 9 groups are assigned the Posttest A, in this case it is the same project that had been initially assigned and the other 6 groups are assigned the Posttest B, which consists of assigning a different project to the one assigned in the pretest A. In total the experiment was implemented with 6 projects, but with the same level of complexity, to be developed, initially without a predetermined model and then with the model proposed by the researchers; all teams performed the two treatments, according to the instructions of the researchers and the respective times were taken.

3. **Results.** A graphical analysis of the collected data is performed indicating that we have a mean μ= 32.9 minutes with a standard deviation σ = 11,2, in Figure 2a we can observe a normal trend in the data, as well as in Figure 2b when applying a smoothed density chart.



Figures 2a - 2b Graphic tests for normality

To supplement the analysis, a QQ theoretical quantile graph of the total minutes of test 3a, and the normality deviation in the QQ 3b graph, which confirms the normality of the data, were performed.



Figures 3a - 3b Graphic test for normality

For the homoscedasticity test or homogeneity of variance test, the Levene test was performed, where the hypothesis that the variance of Total time within each of the 3 levels of TestN is the same is evaluated, the p-value = 0.717319 > 0.10, therefore, it can be concluded that there is no significant statistical difference between the standard deviations of the data at 90% confidence level, as shown in Fig. 4.

	Test	*P-value*
Levene's	0.336357	0.717319

Fig. 4 homoscedasticity test at 90% confidence level

With the test performed we can conclude that the data come from a normal distribution, therefore we can work with parametric data through an analysis of variance (ANOVA), the results can be seen in Table 3 and Table 4. Applying then an analysis of variance, it is concluded that there is a significant difference in the average time, for at least one of the treatments or groups, since the F value is greater than the critical value of F. On the other hand, it can be evident that the lowest mean time to perform the experiment was when Posttest A was applied, followed by Posttest B. The highest average time is given in the groups when applying Pretest A, however it is necessary to verify through another statistical process which treatment or treatments have this significant difference.

Table 3. Summary analysis of variance by type of test

Groups	Count	Sum	Mean	Variance
Pretest A	15	600	40,0	72,6
Posttest A	9	212	23,6	85,3
Posttest B	6	174	29,0	58,0

Table 4. ANOVA for time by type of test 95% confidence level

Origin of variations	Sum of squares	Degrees of freedom	Mean squares	F	Prob.	Critical value for F
Between groups	1633,24	2	816,62	11,09	0,00031	3,35
Within groups	1988,22	27	73,64			
Total	3621,47	29				

Performing the analysis of the box plot, for the times employed in the execution of the software projects without using a requirements quality assurance model and using it, as is the case in Pretest and Posttest, it can be observed that the mean time of posttest A is less than the mean time used for the realization of the projects developed in Pretest A, since it does not overlap. On the other hand, it cannot be concluded in the same way between the mean times of pretest A and Posttest B, nor between Posttest A and Posttest B, since there is an overlap between them; however, a difference can be noted in the mean for each of the treatments, with the highest mean for Pretest A, followed by Posttest B, see Fig. 5.

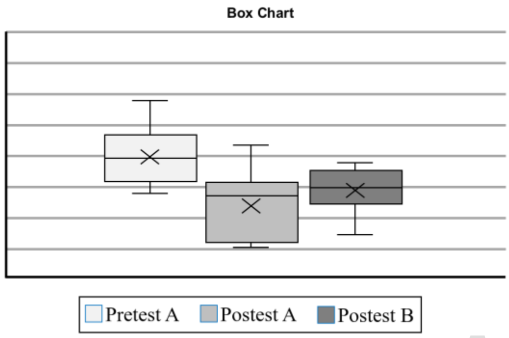


Fig. 5 Box plot of time in minutes, to complete software projects

Consequently, it can be concluded with a confidence level of 95% that there is a significant difference between the mean time for the completion of software projects when using a requirements quality assurance model versus when not using the model, which allows to significantly lower the time for the completion of software projects.

VI. CONCLUSION

If the Process Quality is equal to the product Quality [20]; that is to say, that an improvement in the first impacts positively on the second; then it is valid to conclude that by intervening in the improvement of the software process from the requirements management, and with an experimental engineering process, the quality of the process and product is impacted; in this case, increasing it; this, making use and appropriation of a requirements quality assurance model. Using the experimental process, it was possible to demonstrate that the use of the model through support templates for the tasks of education, elicitation and specification of a requirement was significant and leave a repository of these valid requirements within the context of a problem domain will allow to be reused to improve the good practices of software engineering as a discipline and to reduce costs for reprocessing in the industry [33].

The digital enablers that make I4.0 possible require pillars that ensure their quality and generate trust from users towards this industry [7]. Software Engineering is the discipline that supports the development of these enablers and, as a discipline, it must also transform itself from its foundations. The requirements quality assurance model provided significant knowledge to industry professionals to automate their tasks in requirements management by comparing their techniques used against those offered by the CHAMI model, as well as the need for training of human capital committed to software development. The experiment reflected the commitment of the participants to perform all the tasks they were instructed to do, and it also showed the importance of industry-university collaboration in the process of empirical and practical research when evaluating a new model.

Experimentation processes in software engineering enrich the discipline from applied research and raise the level of quality assurance of software projects for SMEs that, commonly, cannot access large platforms in terms of physical infrastructure and human talent to achieve a product that generates confidence in the end user. The mixed participation of the subjects in the experimental process was enriched by combining experience from the perspective of the lone programmer, or minimum development team made up of at least 2 people or inexperienced students. Consequently, the adaptation process must contemplate a period of getting used to the tool and proficiency to optimize the results, a situation that was observed in the Posttest A tests with better mean times compared to Posttest B due to these factors.

Software development would improve significantly by increasing quality assurance from early stages [1]. The experimental process showed the lack of a key component: training in the mastery of methods, techniques and tools for the stages of education, elicitation of requirements as an initial element to improve software quality and consequently, the digital enablers represented in software for the Industry 4.0 would result in more efficient products in order to make the final user experience more satisfactory and reliable [2]. It is possible that this type of assumptions is reached as conjectures or possible scenarios, but in this case, the research accompanies the conclusions with a statistical analysis resulting from experimentation processes that allow improving the validity of the statements related to the possibility of increasing the quality of the development process.

Finally, if a country like Colombia wants to be more competitive in the international order with an emerging industry such as 4.0 it should bet on improving its software application development processes by promoting specialized training processes to human capital that are part of SMEs and seeking to support, through economic incentives of sponsorship or tax relief, companies that promote improvement and increase their good practices in the production of such products, known as digital enablers

VII. CRediT AUTHORSHIP CONTRIBUTION STATEMENT

L. Peláez-Valencia: Conceptualization, Methodology, Validation, Formal Analysis, Research, Writing - Original Draft, Writing - Review and Editing, Visualization, Supervision, Project Management, Fund Acquisition. **F. Vargas-Agudelo:** Methodology, Writing - Review and editing. **M. Cohuo-Ávila:** Conceptualization, Methodology, Formal Analysis, Supervision, Visualization. **I. Delgado-González:** Writing - Original draft, Research, Visualization. **J. Arias-Vargas:** Methodology, Formal analysis, Visualization, Writing - Original draft, Writing - Revision and editing. **H. Trefftz:** Writing - Proofreading and editing, Formal analysis.

REFERENCES

[1] Dolinina, O. N., Kushnikov, V. A., Pechenkin, V. V., & Rezchikov, A. F. (2018, April). The way of quality management of the decision-making software systems development. In Computer Science On-line Conference, pp. 90-98.https://doi.org/10.1007/978-3-319-91186-1_11
[2] Maxim, B. R., & Kessentini, M. (2016). An introduction to modern software quality assurance. In Software quality assurance, pp. 19-46. https://doi.org/10.1016/B978-0-12-802301-3.00002-8
[3] Peláez Valencia L P, Hurtado Gil S V, Toro Lazo A, and Arias Vargas J L (2018). Aseguramiento de la calidad del Software: acercamiento al marco teórico y al estado de la cuestión desde la disciplina de la Ingeniería de Software. En Investigación e Innovación en Ingeniería de Software, vol. 2, Medellín: Tecnológico de Antioquia – Institución Universitaria, pp. 19–30.
[4] Ardis, M., Budgen, D., Hislop, G. W., Offutt, J., Sebern, M., & Visser, W. (2015). SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering. Computer, 48(11), 106-109. https://doi: 10.1109/MC.2015.345.
[5] Heimann, D. (2014). IEEE Standard 730-2014 Software Quality Assurance Processes. IEEE Computer Society, New York, NY, USA, IEEE Std, 730, 2014.
[6] García Ortega, B. (2021). Industria 4.0. Aspectos clave en su implementación.
[7] Vaidya, S., Ambad, P., & Bhosle, S. (2018). Industry 4.0–a glimpse. Procedia manufacturing, 20, pp. 233-238. https://doi.org/10.1016/j.promfg.2018.02.034
[8] Rubmann M., Lorenz M., Gerbert P., Waldner M., Justus J., Engel P. & Harnish M., (2015). Industry
[9] 4.0, The Future of Productivity and Growth in Manufacturing Industries, BCG-Boston Consulting Group.
[10] Strauß, P., Schmitz, M., Wöstmann, R., & Deuse, J. (2018, December). Enabling of predictive maintenance in the brownfield through low-cost sensors, an IIoT-architecture and machine learning. In 2018 IEEE International conference on big data (big data), pp.1474-1483). https://doi: 10.1109/BigData.2018.8622076.
[11] Deming, W. (1989). Productividad y Competitividad. La salida de la crisis.
[12] Crosby, P. B. (1994). La calidad no cuesta. El arte de cerciorarse de la calidadQuality is free. The art of making quality certain (No. 658.562 C949E). Compañía Editorial Continental.
[13] Juran, J. M., & De Feo, J. A. (2010). Juran's quality handbook: the complete guide to performance excellence. McGraw-Hill Education.
[14] ISO 9000:2015. (2015). Quality management systems — Fundamentals and vocabulary.
[15] ISO/IEC 2382:2015. (2015). Information technology -- Vocabulary, vol. 4.
[16] Fairley, R., & Bourque, P. (2014). SWEBOK v 3.0 Guide to the Software Engineering Body of Knowledge. IEEE Computer Society.
[17] ISO / IEC TS 25011:2017. (2017). Tecnología de la información - Requisitos y evaluación de la calidad de los sistemas y software (SQuaRE) - Modelo de calidad del servicio.
[18] L Valencia, L. E. P., Lazo, A. T., Vargas, J. L. A., & Franco, D. E. R. (2019). Ingeniería de Software: el aseguramiento de la calidad de los requisitos en la industria del software en el eje cafetero colombiano. INGE CUC, 15(2), pp. 110-122. https://doi: 10.17981/ingecuc.15.2.2019.11.
[19] ISO / IEC 25000. (2014). Ingeniería de sistemas y software - Requisitos de calidad y evaluación de sistemas y software (SQuaRE) - Guía de SQuaRE. 2014.
[20] Sommerville, I. (2011). Ingeniería de Requerimientos. Ingeniería de Software, pp. 82-88.

[21] Pressman, R. S. (2011). Ingeniería del software. Un enfoque.
[22] Wiegers, K., & Beatty, J. (2013). Software requirements. Pearson Education.
[23] Lazo, A. T., & Valencia, L. E. P. (2016). Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira. Entre Ciencia e Ingeniería, 10(20), pp. 117-123.
[24] Toro, A., & Peláez, L. (2018). Validación de un modelo para el aseguramiento de la calidad del software en MIPYMES que desarrollan software en el Eje Cafetero. Entre Ciencia e Ingeniería, 12(23), pp. 84-92. https://doi: 10.31908/19098367.3707.
[25] GENERO BOCCO, M. A. R. C. E. L. A., CRUZ LEMUS, J. A., & PIATTINI VELTHUIS, M. A. R. I. O. (2015). Métodos de investigación en ingeniería del software. EDICIONES DE LA U LTDA.
[26] Guéhéneuc, Y. G., & Khomh, F. (2019). Empirical software engineering. In Handbook of Software Engineering, pp. 285-320.
[27] de Oliveira Neto, F. G., Torkar, R., Feldt, R., Gren, L., Furia, C. A., & Huang, Z. (2019). Evolution of statistical analysis in empirical software engineering research: Current state and steps forward. Journal of Systems and Software, 156, pp. 246-267. https://doi: 10.1016/j.jss.2019.07.002.
[28] Peláez Valencia, L, . Cohuo Ávila, M, Toro Lazo, A, Arias Vargas, J, Rodríguez Franco, D, Gómez Osorno, Y y Pérez Correa, D. (2020-12-16.). Diseño de un sistema de información que implementa el modelo automatizado para asegurar la calidad de los requerimientos en proyectos de software. Sello Editorial Tecnológico de Antioquia.
[29] Shaw, M. (2016). Progreso hacia una disciplina de ingeniería de software. In IEEE / ACM 38th International Conference on Software Engineering Companion (ICSE-C), pp. 3–4.
[30] E Teixeira, E., Fonseca, L., & Soares, S. (2018, September). Threats to validity in controlled experiments in software engineering: what the experts say and why this is relevant. In Proceedings of the XXXII Brazilian Symposium on Software Engineering, pp. 52-61. https://doi: 10.1145/3266237.3266264.
[31] Aranda, A. M., Dieste, O., & Juristo, N. (2015). Effect of domain knowledge on elicitation effectiveness: an internally replicated controlled experiment. IEEE Transactions on Software Engineering, 42(5), pp. 427-451. https://doi: 10.1109/TSE.2015.2494588.
[32] Hernández, R., Fernández, C., & Baptista, M. D. P. (2014). Metodología de la investigación (6ta ed.; Mc Graw Hill Educación, Ed.).
[33] Ferrari, R., Miller, J. A., & Madhavji, N. H. (2010). A controlled experiment to assess the impact of system architectures on new system requirements. Requirements Engineering, 15(2), pp. 215-233.
[34] Estándar IEEE para procesos de garantía de calidad de software - Redline", en IEEE Std 730-2014 (Revisión de IEEE Std 730-2002) - Redline. 2014.
[35] Royer, J. C., & Arboleda, H. (2013). Model-driven and software product line engineering. John Wiley & Sons. https://doi.org/10.1007/11557265_22.

Luis Eduardo Peláez Valencia: Undergraduate studies in Technology, Engineering and postgraduate studies in Intellectual Property, Software Engineering and IT projects. He is a Professor and Associate Researcher in the field of Software Engineering, particularly in software quality assurance. https://orcid.org/0000-0002-4836-8336.

Fabio Alberto Vargas Agudelo: Was born in Medellín - Colombia. He received his M.S: Degree in 2010, in systems engineer from the National University of Colombia and His PhD degree in systems engineer in 2016 from the National University of Colombia, he is a full-time researcher at Tecnológico de Antioquia. His principal research interests are software engineering, requirements engineering and agility. https://orcid.org/0000-0002-6921-4918

Miguel Ángel Cohuo Ávila: Was born in Mexico. He received his M.S: Degree in 2008, in Information Technologies from the Latino University and His PhD degree in Computer Systems in 2015 from the Southern university, both studies in Merida Yucatan Mexico, he is a full-time researcher at National Technological Institute of Mexico, Calkini Campus (ITESCAM.). His principal research interests are empirical research in software engineering, requirements engineering and quality models. https://orcid.org/0000-0002-4365-187X

Iván Andrés Delgado González: received his formation as Systems Engineer in 2005, from the Universidad Central, Bogotá, Colombia; from M.Sc. in Computer Systems in 2012, from the Universidad Metropolitana de Educación, Ciencia y Tecnología, Panamá. His research interests include Applications Development, Simulation and Technologies, and Information Systems and Quality Assurance. He is a researcher at Fundación Universitaria Juan de Castellanos, in the Engineering Faculty. https://orcid.org/0000-0002-0417-2649

Juan Luis Arias Vargas: He is an Industrial Engineer and has a master's degree in the teaching of mathematics (Statistics area) from the Universidad Tecnológica de Pereira. He is an associate professor at the Faculty of Basic Sciences and Engineering of the Universidad Católica de Pereira, a member of the research group Between Science and

Engineering classified in A by the ScienTI Colombia System. He is also a Associate researcher recognized by the same System. His areas of interest are industrial engineering and statistics. https://orcid.org/0000-0002-7997-1891

Helmuth Trefftz Trefftz: Professor Trefftz received a B. Sc. In Computer Science from Universidad EAFIT (Colombia), a M. Sc. In Computer Science from Maharishi International University (Iowa, United States) and a Ph.D. in Electrical and Computer Engineering from Rutgers University (New Jersey, United States). His research interests include Virtual and Augmented Reality, Computer Graphics, User Experience Design and Quality Assurance in Higher Education. He is currently a professor at Universidad Eafit, where he leads the Virtual Reality laboratory. https://orcid.org/0000-0003-2441-506X