Process and progress of requirement formalization in Software Engineering

Proceso y progreso de la formalización de requisitos en Ingeniería del Software

Edgar Serna M.^{1*} Alexei Serna A.¹

Recibido 26 de febrero de 2019, aceptado 26 de marzo de 2019

Received: February 26, 2019 Accepted: March 26, 2019

ABSTRACT

Since the middle of the last century was initiated the research in formal methods and was presented proposals and methodologies to apply them in software development. The idea was overcome the diagnosed software crisis through the materialization of the life cycle of this product development. In this article is presented the results of a revision of the literature, about progress and develop the requirement formalization. The conclusion is that both are slow: there is not enough interest in the industry, the academy does not train in formal methods, and there is no sponsorship for this research field, professionals fear mathematics and traditional methodologies of software engineering are still the most used in development teams. Due to the quality efficiency, software security and reliability, it is necessary to reactivate the research and experimentation with all the formal methods, because the hope is that mathematics will be the toll with which the crisis promulgated in the 60s is overcome.

Keywords: Formal methods, requirement engineering, software quality, formal language.

RESUMEN

Desde mediados del siglo pasado se inició la investigación en métodos formales y se presentaron propuestas y metodologías para aplicarlos en el desarrollo de software. La idea era superar la diagnosticada crisis del software mediante la matematización del ciclo de vida del desarrollo de este producto. En este artículo se presentan los resultados de una revisión de la literatura, acerca del progreso y desarrollo de la formalización de requisitos. La conclusión es que ambos aspectos son lentos: no hay interés suficiente en la industria, la academia no capacita en métodos formales, no hay patrocinio para este campo de investigación, los profesionales le temen a las matemáticas y las metodologías tradicionales de la Ingeniería del Software siguen siendo las más utilizadas en los equipos de desarrollo. Debido a la deficiencia en la calidad, la seguridad y la fiabilidad del software, es necesario reactivar la investigación y experimentación con los métodos formales, porque la esperanza es que las matemáticas sean la herramienta con la que se supere la crisis del software promulgada en los años 60.

Palabras clave: Métodos formales, ingeniería de requisitos, calidad del software, lenguaje formal.

INTRODUCTION

Companies that face a changing environment in which demands and solutions for new products have to be carefully analyzed and evaluated. In

order to maintain competitiveness, must choose a proper combination of functions and technologies, and be able to put them on the market in the most economical way possible. With shorter life cycles, they are forced to take the products even faster to

¹ Instituto Antioqueño de Investigación. Medellín, Antioquia, Colombia. E-mail: eserna@eserna.com; alexeiserna@gmail.com

* Corresponding Author: eserna@eserna.com

market maturity [1]. Furthermore, the time or horizons planning for new products must be synchronized carefully. This way, it is possible to develop and line up users' needs solutions.

When a system is designed, is recommend that objectives, functionalities, and restrictions identify as accurately as possible. This what constitutes the requirement document or the specification of the same. Something that is repeated daily in this process in which requirement are usually writing in a natural language, either because engineer does not know the formal languages or because consider that is too soon in the life cycle to use a formal specification (it means, adduce not to have enough data to successfully write a formal requirement). The principal problem with the requirement writing in the natural language is that cannot serve as supplies for verification and validation of automatized techniques. In fact, formal languages are mandatory if the test plan is pretended to use verified automatized techniques because are they are the only one's languages that computers can understand. By the other hand, they are not ambiguous, it means, the sentence cannot be understood in different ways. This implies that humans also can benefit from the use of this languages because will understand completely the requirement without a previous interpretation from the user. The principal problem is that both activities, reading, and writing, are not easy to achieve when it comes to formal languages.

According to traditional industrial practice, the requirements are specified in a natural language and the testing of errors in manually. The deficiencies of this tools chain are well known: ambiguity in the description of the requirements, increase in costs and delivery times, and re-engineering process in the design and architecture so [2]. Besides, the revisions can detect errors, but not guarantee their absence. A way to avoid this inconveniences is through the requirement formalization and the formal automatic analysis. Diverse researchers have been carried out with this aim, especially in support from formal languages and the automatization tools [3-5]. The question is that being able to determine if these results are feasible in practice, cannot achieve without a decisive argument of principles that can be applied uniformly in real life projects.

The process of formalization consists in write requirement using a formal mathematic notation.

Which can be a difficult task it is made directly from an informal requirement and cannot be a real guide to formalization: only knowledge and experience, and maybe some heuristics [6], can help to know how to get a formal writing. The formal approaches can help with a deep analysis trough precis semantic of formal requirement. However, typically the formalization is a difficult task and often is used separated from the validation. The traditional method to resolve this problem has conformed a team with knowledge and experience in formalization and the different components of the planned solution. But, even teams with these characteristics have to recur to training materials, books, system support material, scenarios and even another kind of document that described the process of the business and the configuration approach. The problem is that the analysis of this documentation takes time because contain unstructured and non-formalized information.

The statistics show the importance of select a proper approach to Requirement Engineering to develop the project. Studies performed by Boehm [7-8] and others have proven that the potential impact of a deficient requirement formulation is substantial. Boehm suggests that errors in the specification and analysis of requirement are the more numerous in a system, and the most of them are not found in the developing stage, but in the testing and delivery. The resulting cost to correct these errors has a direct relationship with the time invested in its search. An error found in the requirement stage, cost about a fifth part of what will be if it is in the testing stage, and a fifteenth part of what will cost after the system is in use. Hence the importance of work in the requirement formalization, because this way could delivery non-ambiguous requirement and possible with few errors. In this article is presented an analysis of progress and process of the requirement formalization. The aim is to draw a map of what has been done, what has been achieving and which is the future of the formalization as Requirement Engineering tool.

RELATED WORKS

Lars Mathiassen and Andreas Munk [9], ensure that formalizations are related to both the types of expressions and behavior. In their work argue the limits to apply the formalization in both senses and

illustrated them with practical examples of system development. Besides, establish that formalizations are valuable in some situations, but enough in other. The alternative is to use them in an uncritical way and analyzed each situation in particular, and from there choose one combination between a formal approach and one informal. They do not make an analysis to formalization develop until that moment, neither project its future use.

Fantechi [10] present a prototype assistant to requirement formalization in the design of reactive systems. It is a tool to translate automatically phrases in natural language to logic temporal formulas. According to them, the phrases in natural language is used to express requirement in an informal way, and that temporal logic is adequate to express the proprieties of the systems, specified in terms of process algebra. The question with this work is that if their aim was to bring closer the formalization to a large audience, achieve exactly the contrary due to the high content of the applied logic. It does not present a previous analysis of which been able to leverage their proposal.

To Vilkomir, Bowen, and Ghose [11], the important task in Requirement Engineering is oriented to resolve inconsistencies between regulators and critic informatics systems developers. In their work, propose a new approach to the regulation process, including the use of formal regulatory requirement as the base to develop software evaluation methods. They focus on the differences between the regulations and suggest an intermediate approach. Argue that the normative package must include not a just regulatory requirement, but also the methods of its evaluation. This approach is illustrated with requirement examples of informatics control systems protection through not authorized access and against common software failures, using the notation Z to formalization. In their proposal do not have into account the work that others have advanced in this sense and neither define what to do with the no-critic systems. That is why is difficult to be able to determinate develop of the requirement formalization from their contributions.

Morimoto, Shigematsu, Goto and Cheng [12] proposes a hybrid method to formally verify if requirement specification of a system satisfies the evaluation criteria of security define in the norm

ISO/IEC 15408. Classify the functional requirement in static and dynamic, then formalized the first with the notation Z and the second with temporal logic. As result, developers can use easily the method to verify if the specification of a system satisfies the security functional requirement. According to with the authors, this method is an evolution and improve the previous proposal, where just is apply the notation Z. Although is a valid proposal to the formalized functional requirement, there are not found references to a similar process with the no-functional. Besides, propose neither analyzed the developing state of the requirement formalization, from which could extract and structure their model.

For its part, Ram Chatterjee and Kalpana Johari [13], describe a simplified and corroborative approach to requirement formalization. Their proposal implies cases of use, scenarios and transition diagrams of the state as a base to automatize the formalization process, that achieves through their own development tool, which exemplified the underlying concept and illustrates the facility of automatization. Prior to their proposal, do an analysis of the theories, principles, and contributions around requirement formalization develop. The problem is that their study and proposal is oriented to paradigms of programming oriented by objects, which, although its use is much generalized, still is a static model. This contradicts the necessity of dynamism of the solutions to current complex problems.

To Serna [14], the application of the formal methods in the industry has widely progressed in the last decade and the results are promising. Describe eight experiences of real cases, of which five apply to formal requirement specification. In spite of this achieving and that have been documented in numerous studies, to this author still is common the skepticism about its utility and applicability. With the results obtained in this experiences, he found a profile of the necessities of the industry, and propose that must the community of the formal methods the responsible of given them a solution. Conclude that formal method that is used in the generalized form and surrounds all the Computational Science, but still does not have the necessary reception in the curriculum.

Peres, Yang and Ghazel [15] address an important matter related to Requirement Engineering: how

to guide and help requirement formalization. In order to support the formalization process, propose a methodology based on a formal structure, as the cornerstone of the refinement process. To this authors, system requirement is usually writing in a natural language, due, in general, is the one that has the biggest understanding of the different interest groups. However, the use of this language potentially gives place to interpretation problems, that must be solved before to use automatized verification techniques. Present the revision of a series of works in which propose directly formalized the requirement, but is essentially directed to Software Engineering, but does not describe how to do it in Requirement Engineering.

METHOD

This article is the result of a literature review [16], in which is pretended to find answers to a: 1) what is meant by formal methods? 2) What is published in the literature in relation to the formalization of requirements? And 3) how much progress has been made in formalizing requirements? After applied the methodology described by this authors, a final sample was made of 40 related works, whose results served as the basis to find the answers that are detailed next.

RESULTS

Formal methods

Yan [17] states that formal methods are techniques based in mathematics, often with the support of reasoning tools, which can offer a rigorous and effective way to model, design and analyzed informatics systems. Additionally, Seceleanu [18], Ambrosio and Andrade [19], Kaur, Gulati and Singh [20] and Barbosa and Lumpe [21] agree that formal methods provided a mathematic structure inside which, in a systematic way, is used to specify, develop and verify a system. To Kaur, Gulati, and Singh [20] allow to develop an informatics system that is more complete, coherent and unequivocal, and by the other side, to Gonzalez and Cabot [22] avoid the introduction of inaccuracies or ambiguities in the process.

Must take into account that formal notion I this context must be understood in general terms like logic/mathematics [23]. Additionally, Barlas,

Koletsos and Stefaneas [24] manifest that these mathematics principles do not need language knowledge and that can easily understand in an international context. Exposing some point of views about formal methods, You, Li and Xia [25] reflect that correct and trustable software develop has been a continuous problem that must resolve urgently, and that an effective way to achieve this is a formal theory. Because offer the possibility to create a software with little or any default [26]. This makes a process like this, to be qualitatively different to conventional software engineering because are reliable, sturdy and ideals to use in software design of high integrity systems that involve security issues, where the cost of mistakes can be very high [24].

The use of formal methods allows to clearly state the specification of a system, generating models that define behave in terms of what must do and not how to do it [19]. Thanks the correct specification process, can verify derived properties of each module through techniques of reasoning associated with formal models, such as testers of theorems and models verifiers. When collecting opinions of different authors about formal methods, was found that: they are generally expensive to apply [27], especially to engineer training; are too slow in its application [23]; increase the quality and reliability of a design and show a relation with practical problems and its potential to future, by the other hand, its applicability demands to be easy to use and has efficient verification tools [18]. By the other hand, formal methods can help to resolve some of the problems currently facing the software industry, such as clients' dissatisfaction for incapacity to fulfill with requirements and a high cost of support [27]. The principal aim of Software Engineering is allowed developers to create systems that operate in a reliable form [20], and formal methods are designated to provide the media to achieve a bigger precision in thinking and documentation of this engineering.

The principal advantages of formal methods are: formal specifications are correct and coherent and can be proved and verified; the ambiguity in the specification is avoided automatically [20]; can guarantee the security and reliability of software systems [25]; job error is greatly reduced [25]; and is eliminated the design errors or inconsistencies before its implementation. This is essential to

reliable develop systems, especially in critic software applications [28], which present fewer errors during the verification and often required shorter times of developing and consume less economic resource [29]. These advantages are achieved using mathematics because it is a medium to develop an exact model that can describe the object and the functioning in a brief and precise way. Besides, because it can represent abstract unities and the system specification is coherent with the needs of clients and users. By the other hand, mathematics offers a way of showing the contradiction and the imperfection in the specifications, at the same time that coherence between design and specification [25].

At the other end, also is found limitations to formal methods. According to Kaur, Gulati and Singh [20], are difficult to learn and use and, due to the lack of development models and tools, complex problems are difficult to manage. To Mandrioli and Milano [29] we are in an asocial general attitude against the rigorous reasoning of formal methods. Lockhart, Purdy and Wilsey [28] conclude that, unfortunately, current formal methods are complex and expensive to be used widely in software systems, because the refinement of a formal specification is a complex process, slow and is not well supported by tools. Another important question, but negative in the case of formal methods, is that most of the programmers perceive them as a useless theory that does not have any relation with it does [29]. According to these researchers, there is not a faster way to lose the attention of programmers' audience than show them a mathematic formula. Unfortunately, one of the central problems is that has not been comparable advances in formal methods and, although emerge new languages and new logics, the errors in programming design, that come since 1967, still can find in today software [30]. Serna [31] agree that, despite its significant advantages, formal methods are not widely used in industrial software development, and the academy shows a wide interest to include them in their curriculum.

Requirement formalization

The requirements are considered as a fundamental stone over which can develop software solutions, and currently exist an increasing demand on approaches more rigorous and systematic to formalized them [32]. Serna and Serna [33] argue that formal methods are used today to model complex critic security systems,

but few are worked in requirement formalization from the first stages of requirement engineering.

The presentation more used to the formalized requirement are formal specification languages [32]. With this kind of specification, developers understand better the system and elude ambiguities, the flows, the omissions and inconsistencies of requirement. Furthermore, the specification is an important mechanism of communication between clients and designers, between designers and executors, and between executors and testers [34]. But, formal specifications are not documents that are writing once and generally not achieved at the beginning of the software development process. It needs time to create a first useful version that, after a lot of effort and reviews, allows develop a close specification to needs that clients have in mind [17] and Bollin and Raunner [35] argue that a good level of comprehension is a key matter as a quality attribute. To Wolf [36], using formal specification language, the system can be described with precision as soon as its functionality, concurrence, integrity, accuracy, ... These means system properties can be analyzed without having to really run it.

Using formal specification, system properties are described using a language with a syntax and a semantic defined mathematically [11]. Some formal languages, such as Z, B and VDM, is focus in specified the behaved of sequential systems, where the states are described in mathematical structures, such as sets, relations and functions [25]; while methods like CSP, CCS, Statecharts, Temporal Logic, and Automaton, is focus in the specification of the system behave in terms of sequences, trees or partial orders of events [37]. The use of formal languages like Z and B can improve the trust of the user in the system and its impacts on the use itself [38]. Tamrakar and Sharma [39] affirm that Z, B and VDM are languages of formal specification use to specify the necessities of the users in a mathematical language, which product can be tested and verified automatically.

Z works in high abstraction to system level and provides a solid foundation to design of the same. Through the use of this language, not just is found more errors in the specification but also in the test and maintenance stages. This is convenient because, with traditional engineering, the cost of

correct errors in posterior stages in higher than in the first ones [38]; besides, Z is a way to decompose a specification in small parts call schemes. By its part, B is one formal method more known. It is based on first-order logic, set theory, the arithmetic of integers and generalized substitutions; is used to software design from functional requirement and allows produce test cases that show the exactitude and consistency of the software model applied. The objective of B obtains a proven and reliable product [40]. While VDM is used to probe the equivalence of programming languages concepts [39].

According to Pandey and Batra [32], in the current digital era, companies face to challenge of liberating quality software projects, in time and inside the budget, but the reality is that is delivered with errors, lack of functionality and sometimes with over costs. The extra cost is produced due to errors in the requirement specification, that can cost a lot of time and money correcting them when are detected in late phases of the life cycle. Thus, in requirement specification must select one formal methodology to address reliability during Requirement Engineering and design, because formal methods allow developing errors before product liberation [28].

Specification errors can be reduced drastically through the use of formal methods and, in consequence, the software engineer can create a specification more complete, coherent and unequivocal with conventional methods [20, 38]. For they part, Bollin and Rauner [35] manifest that a good formal specification is syntactically and semantically correct, that allows a mapping without losses among all the specification concepts and the mental model of the specified system; also add that must be complete, coherent and proper and have into account that facility comprehension is an essential requirement to decide about its semantic correction.

Does not exists an informatics tool that can guarantee the complete correction of a computer model [39].

Therefore, although the specification is writing using any of the formal methods, always contains a high error potential. It means that art of writing a formal specification does not ensure that the developed system is coherent, correct and complete. Thus, if the specification is proving and is analyzed with the support of automatized tools, increase the trust about the system with possible potential errors identification, if exist, in the syntax and semantics of the formal description. One of the ideas more disseminated about the use of the formal methods is its almost null application in the software industry. While its adoption has been slow, existing cases of important companies that have had success applying them in real projects [19].

In Abstract, to know what is published in literature related to requirement formalization, in Table 1 is presented the abstract of the research results. The observation line of time is from 2010. Although is known that most parts of the work in this area was presented in the second half of the last century, the aim of this research was verified its current progress. The works of the final sample were classified according to the primary approach presented. Is clarify that many authors present results in more than one typology, it means, their research can be theoretical and at the same time experimental.

- *Theoretical research*: articles that contain definitions or descriptions about the requirement formalization.
- *Experimental research*: show results from laboratory experimentation.
- *Practical application*: describe and apply methods, techniques or formalization procedures for requirement in case studies.
- *Industrial application*: in which requirement formalization is applied in real cases.

Progress in requirement formalization work

In Table 2 is presented the tendency in requirement formalization progress in the last quinquennial, according to this research results.

Table 1. What is publish in the literature about requirement formalization.

Media		Theoretical	Experimental	Practice	Industrial
Journal	66%	63%	34%	20%	23%
Conference	34%				

Table 2. Requirement formalization develop.

Author	Develop
Bhavsar [41]	Presents a use of the formal methods in the requirement formalization focus in the user. According to the author, the results are promising about improvement in the quality.
Kaur, Gulati and Singh [20]	Analyzed the B method to requirement specification and describes the cases of use and the security properties. Although is application is simple, is one contribution toward the massification of the formal methods in the requirement specification.
Honghao [42]	Propose a methodology to formalized Requirement Engineering and Design, oriented to analyzed and manage requirement. This research provides lights to show the utility and the formal methods in Requirement Engineering.
Hassan, Eltoweissy, Bohner and El-Kassas [43]	Propose the analysis and formal design of the security of the requirement using B specifications. The results show effectiveness and capacity in the production safe software.
van der Poll [44]	Describe the formal methods more common and the importance of applying the formal specification in the first stages of the life cycle. Face the positions of the critics and defenders of the formal methods and presents examples of stories of success in the industry.
Singh [45]	Use the notation Z to formalize the requirement, searching protection and security against internal threats, in a successful case of study.
Cimatti, Roveri, Susi and Tonetta [46]	Develops a methodology and techniques to formalized and validate requirement in critic's applications of security. Apply the result in case of study with improve results.
Serna [31]	Describe the application of the formal requirement specification and the improvement that is achieved in the quality of the final product. Propose the incorporation of the formal methods in the curricular contents in the Computational Science program.
Barringer, Groce, Havelund and Smith [47]	Presents a practical method to use formal specifications in register test in Mars Science Laboratory. Shows the advantages of using a formal specification language.
Bishop, Hay, and Nance [48]	Integrates the concepts of formal methods from the beginning of the programming and, through a simple didactic, shows how to make a formal specification and which models can be used.
Fernández, Ambrosio, Andrade, Cruz, José, Ortiz and Sánchez [49]	Is a revision of the application of the formal methods in real projects in the industry, and presents diverse cases of success and describe the current situation in Mexico as for the application of formal methods.
Serna [50]	Makes a historical tour through the formal methods and describe its applications and benefits in the requirement formalization. Propose that research in this area should be oriented to practical applications and search reduce the utilization costs.
Ibrahim, Alagar and Mohammad [51]	Applied a formal base specification and verification of requirement, presents a formal definition and a theory of formal composition to Requirement Engineering. Applied the formal verification and describe the results.
Yan [17]	Describe the application to formal specification in Communication-Based Train Control (CBTC), obtain a deep and exact description of the system, and Z language in developing. Concludes that using formal methods gain confidence to secure the system.
Ammar and Abdallah [52]	Presents a new focus to formal specification and verification base in the logic rewrite. They applied it in case of study with promising results in quality improvement.
Kaur, Gulati and Singh [20]	A comparative study of formal specification methods and description of their reasons of why use them in the industrial process.

Author	Develop
You, Li and Xia [25]	Shows the achievements and formal methods and validates its application to guarantee the security and system stability.
Barlas, Koletsos and Stefaneas [24]	Presents a comparison between the traditional and formal requirement specification. With the second achieves a better comprehension, eliminates ambiguities and force a better precision of the specification. Shows applications and benefits in its use in the industry.
Wolff [36]	Use the experience in the industry to combine the agile develop Scrum with the formal methods, but from a theoretical conceptualization. Presents an evaluation and discussion of the benefits use to formalized requirements.
Bollin [53]	From a theoretical framework concludes that process formalization of the requirement is a problem of transcultural adaptation, analyzes the pros and cons of the same and discloses a refined model for a formal software development process.
Serna and Serna [54]	Theoretical research in which concludes that formal methods have achieved significant advances, which glimpse ambitious applications in the future to support the rigorous and coherent curriculum in computational science.
Pandey and Batra [32]	Argue the diverse languages of formal specification and analyzed its strengths and weaknesses. Perform recommendations to use of formal methods, especially in requirement Engineering.
Atlee, Beidu, Day, Faghih and Shaker [55]	Through a series of recommendations to formalize the requirements, contributes to the practical achievement of the selection of a specific technique. His objective is to improve the acceptance of formal methods in the modeling of requirements.
Serna and Serna [56]	The formal specification still has limited uses, but the community has a different understanding of its usefulness and need. So far, its development is mainly focused on evaluating the related tools.
Chen [57]	Analyze the domain of formal methods to identify defects and reduce failures. Apply the formalization in the specification, modeling, and verification.
Chan, Hexel and Wen [58]	Propose a methodology for modeling requirement to convert informal to formal. With them, seeks to eliminate ambiguities and defects, refine and perfect the system and serve as a means of communication between the interested parties.
Diallo [59]	Makes a study of formal specification languages, describes the models that can be applied and perform a general analysis. Perform an experimentation with debug statistics in a case study.
Noaman, Alsmadi and Jaradat [60]	Applies the formal specification by means of Z to enhance system security and reduce threats. The results are promising for future extensions and elaborations.
Lockhart, Purdy and Wilsey [28]	Shows the utility of formal methods for the requirement specification in critical security system. The formal specification reduces the ambiguity of the design, helps to test consistency, and increases confidence and performance.
Wu, Dong and Hu [61]	Propose transformation rules from model B and describes the requirement with abstracts machines, For the author, the formal specification is an intermediate process between requirement specification and code writing.
Serna and Serna [62]	Perform a revision of the literature, makes a tour of the essence, the function, the use and the inconvenient of the formal specification techniques and analyze the criteria of valuation and evaluation of its weaknesses. Search that formal specification secures as a basic activity of research and academy training.
Bollin and Rauner [35]	Shows a series of recommendations to perform formal specifications, and identify that software reading facility of a specification is a factor to improve the software quality.

Author	Develop
Schraps and Peters [37]	Apply formal grammar with semantic annotations to formalize requirement, so that they can be analyzed and processed by the computer. The formulated requirements can be analyzed and processed in the early stages of development.
Serna and Serna [63]	States that many challenging problems in the system construction required the formal support to its modeling and analysis. In the research and application process of the Computational Science exist a growing number of applications. But still do not are an integral part of the formative process in undergraduate and graduate.
Azeem, Ahsan, Minhas and Noreen [64]	Use formal Z specification to improve the quality and reliability of the system. Shows related work and applies a proposal in a case study in health.
Li, Pan, T. Hu, Sung and Yuan [65]	Applies Object-Z to describe complex systems. Shows its utility in a case study of a gasoline supply system. The results obtained encourage the use of the formal specification.
Serna and Serna [66]	Describes relevant aspects of formal methods and provides a framework for the future of research in the area. The results invite reflection on this component of Computational Science and the need to have a broader and more solid community dedicated to promoting and applying the formalization of the requirements.
Jeffery [26]	Make an empirical study to find the productivity of projects that use formal methods, specifically when formalizing the requirements. Identify a series of questions about the productivity results of these projects.
Tamrakar and Sharma [39]	Compare three methods to formally specify: Z, B, and VDM. Using formal specifications does not ensure a complete system, but increases confidence in it.
Singh and Yadav [67]	Use the Event-B method to write formal specifications, in order to guarantee the compression of the limits in which an algorithm can be used. For the Author, the specification, validation and formal verification with the key to obtaining a better design.
Walter [68]	Make a representation of formalization of a requirement by means of SysML diagrams using RSL semantics. Formalizes only the non-functional requirements and that design applies to a case study in the automotive area.

ANALYSIS AND DISCUSSION

According to the revision of the literature in this research, the work in requirement formalization has not had the boom that was expected in the last five years. Some authors have analyzed this fact and presented their conclusions. Among the causes stand out: lack of academy formation, lack of acceptance in the industry, there is not demonstrations really important of its advantages, the costs, lack of trained personnel, engineers have phobia of mathematics, industry does not want to experiment with principal that has not enough testing and in the last decade formal methods research has lost interest and sponsorship.

The researchers of the middle of the last century were convinced that the only way to improve the quality of

software products was through the mathematization of Software Engineering, but as results begging to be published, the industry starts to find and place barriers to the progress of this work. Just critic software tucked it as solution table to solve its problems of reliability and safety, and are diverse the cases of success in these develops. The problem emerges when development companies of commercial software face situations of a breach, that increase the costs of the process and force them to dedicate less time to the same structure of the activities. The decision is to decrease as much time as possible and, as a traditional form of performing the verification and validation is like a containment barrier to the end of the life cycle, in this phase is the one that is cut to find some time.

Furthermore, requirement formalization progress is slow because the academy does not train on formal

methods. This topic has been relegated to few types of research that convince their postgraduate students to work with them. With the inconvenience analyzed before that professional fear to all that has to be with mathematics. The recommendation is that universities include courses related to undergraduate curriculum in Computational Science, in such a way that students be interested and deepen in its assimilation and comprehension.

According to the authors analyzed in this research, requirement formalization is investigated in a mostly theoretical way, and this way it will not be possible to convince the industry of its application. It may be that the theorization stage has already been overcome in many works that were presented in the last century. The current necessity is massified formal methods and find the way to apply them in the mathematization of Software Engineering, but with a cost that is within reach of the software development companies. This way will be possible rekindle the interest in this area of work and to find necessary sponsorship to execute the necessary research. If the aim is to achieve finally overcome the software crisis, formal methods must be the focus of development. Because its benefits have already been enough proving and because language which works the computer is mathematics.

CONCLUSIONS

In this article is presented the current development and progress of requirement formalization, as result of the research base in a literature revision. The aim was determinate the actual state of research in this area, with the encouragement to present the community a development report reach until the moment.

Results show that its progress and development are slow and that in the last years the interest forgives them an opportunity has diminished. Although is accepted that formal methods are a useful tool and necessary to overcome the crisis of software quality, very few researchers want to increase its level of maturity or acceptance. Academy has part of the guilty in this situation, because still does not include them in the curriculum of undergraduate programs, and relegates them to only some postgraduate.

The industry needs demonstrations of the benefits that formalization represents, but till now does not

have a direct inclination to support initiatives that in this sense propose the community. Only develop of critic systems is appreciate a wide participation of the companies of development and industry, because the aim is that works whiteout jeopardizing human life or economic inversions. It is necessary the biggest work to reduce the current cost of formalization, but, above all, is urgent to count with the persons trained to apply them and give them continuity.

In this society software – dependent, because very few of their activities are outside of the ambit of this technological development. But products that are released or deliver to users still does not satisfy aspects of reliability and security. Mathematics offers the possibility of overcoming this problem and its representation in formal methods is a promising alternative.

REFERENCES

- [1] M. Chalupnik, D. Wynn and J. Clarkson. "Approaches to mitigate the impact of uncertainty in development processes". Proceedings 17th International Conference on Engineering Design. Stanford, USA, pp. 1-2. 2009.
- [2] A. Dahlstedt and A. Persson. "Requirements interdependencies - Molding the state of research into a research agenda". Proceedings Ninth International Workshop on Requirements Engineering. Klagenfurt, Austria, pp. 71-80. 2003.
- [3] M. Heimdahl and N. Leveson. "Completeness and consistency analysis of state-based requirements". IEEE Transactions on Software Engineering. Vol. 22, Issue 6, pp. 3-14. 1995.
- [4] C. Heitmeyer, R. Jeffords and B. Labaw. "Automated consistency checking of requirements specifications". ACM Transactions Software Engineering and Methodology. Vol. 5, Issue 3, pp. 231-261. 1996.
- [5] L. Yu, S. Su, S. Luo and Y. Su. "Completeness and consistency analysis on requirements of distributed event-driven systems". Proceedings 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering. Washington, USA, pp. 241-244. 2008.

- [6] I. Jureta, S. Faulkner and P. Schobbens. "Clear justification of modeling decisions for goal-oriented requirements engineering". *Requirements Engineering*, Issue 13, pp. 87-115. 2008.
- [7] B. Boehm. "Model and metrics for software management and engineering". IEEE. 1984.
- [8] B. Boehm. "Industrial software metrics top 10 list". *IEEE Software*. Vol. 4, Issue 5, pp. 84-85. 1987.
- [9] L. Mathiassen and A. Munk. "Formalization in systems development". *Lecture Notes in Computer Science*, Issue 186, pp. 101-116. 1985.
- [10] A. Fantechi. "Assisting requirement formalization by means of natural language translation". *Formal Methods in System Design*. Vol. 4, Issue 3, pp. 243-263. 1994.
- [11] S. Vilkomir, J. Bowen and A. Ghose. "Formalization and assessment of regulatory requirements for safety-critical software". *Innovations in Systems and Software Engineering*. Vol. 2, Issue 3, pp. 165-178. 2006.
- [12] S. Morimoto, S. Shigematsu, Y. Goto and J. Cheng. "Classification, formalization and verification of security functional requirements". *Lecture Notes in Computer Science*, Issue 4910, pp. 622-633. 2008.
- [13] R. Chatterjee and K. Johari. "A simplified and corroborative approach towards formalization of requirements". *Communications in Computer and Information Science*, Issue 94, pp. 486-496. 2010.
- [14] A. Serna. "Formal Methods in industry. Los Métodos Formales en la Industria, Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS). Vol. 2 N° 2, pp. 44-51. 2012.
- [15] F. Peres, J. Yang and M. Ghazel. "A formal framework for the formalization of informal requirements". *The International Journal of Soft Computing and Software Engineering*. Vol. 2, Issue 8, pp. 14-27. 2012.
- [16] E. Serna and A. Serna. "Methodology for perform reliable literature reviews". *Revista Información, cultura y sociedad*, In press. 2018.
- [17] F. Yan. "Studying formal methods applications in CBTC". *Proceedings International Conference on Management and Service Science*. Wuhan, China, pp. 1-3. 2011.
- [18] C. Seceleanu. "Formal methods applied in industry: Success stories, limitations, perspectives - panel introduction". *Proceedings IEEE 35th Annual Computer Software and Applications Conference*. Munich, Germany, pp. 448-449. 2011.
- [19] M. Ambrosio and G. Andrade. "Métodos formales aplicados en la industria del software". *Temas de Ciencia y Tecnología*. Vol. 15 N° 43, pp. 3-12. 2011.
- [20] A. Kaur, S. Gulati and S. Singh. "A comparative study of two formal specification languages: Z-notation & B-method". *Proceedings Second International Conference on Computational Science, Engineering and Information Technology*. Coimbatore, India, pp. 524-531. 2012.
- [21] L. Barbosa and M. Lumpe. "Formal aspects of component software". *Lecture Notes in Computer Science*, Issue 7684, pp. 253-254. 2014.
- [22] C. González and J. Cabot. "Formal verification of static software models in MDE: A systematic review". *Information and Software Technology*. Vol. 56, Issue 8, pp. 821-838. 2014.
- [23] S. Gruner. "Editorial: Special section on formal plus agile methods". *ACM Software Engineering Notes*. Vol. 36, Issue 4, p. 26. 2011.
- [24] K. Barlas, G. Koletsos and P. Stefaneas. "Extending standards with formal methods: Open document architecture". *Proceedings International Symposium on Innovations in Intelligent Systems and Applications*. Trabzon, Turkish, pp. 1-5. 2012.
- [25] L. You, J. Li and S. Xia. "A survey on formal methods using in software development". *Proceedings International Conference on Information Science and Control Engineering*. Shenzhen, China, pp. 1-4. 2012.
- [26] R. Jeffery, M. Staples, J. Andronick, G. Klein and T. Murray. "An empirical research agenda for understanding formal methods productivity". *Information and Software Technology*, Issue 60, pp. 102-112. 2015.
- [27] S. Shirali and M. Shirali. "Using formal methods in component based software development". In Sobh, T. (Ed.), *Innovations and advances in computer sciences and engineering*, pp. 429-432. Springer, 2010.

- [28] J. Lockhart, C. Purdy and P. Wilsey. "Formal methods for safety critical system specification". Proceedings 57th International Midwest Symposium on Circuits and Systems. College Station, USA, pp. 201-204. 2013.
- [29] D. Mandrioli and P. Milano. "On the heroism of really pursuing formal methods - Title inspired by Dijkstra's 'On the Cruelty of really Teaching Computing Science' ," Proceedings of the Third FME Workshop on Formal Methods in Software Engineering. Austin, USA, pp. 1-5. 2015.
- [30] D. Parnas. "Really rethinking 'formal methods'". Computer. Vol. 43, Issue 1, pp. 28-34. 2010.
- [31] E. Serna. "Formal methods and Software Engineering". Revista Virtual Universidad Católica del Norte, N° 30, pp. 158-184. 2010.
- [32] S. Pandey and M. Batra. "Formal methods in requirements phase of SDLC". International Journal of Computer Applications. Vol. 70, Issue 13, pp. 7-14. 2013.
- [33] E. Serna and A. Serna. "Formal specification in context: Current and future." Ingeniare. Revista Chilena de Ingeniería. Vol. 22 N° 2, pp. 243-256. 2014.
- [34] H. Krad. "Formal methods and automation for system verification". Proceedings 4th International Conference on Modeling, Simulation and Applied Optimization. Kuala Lumpur, Malaysia, pp. 1-12. 2011.
- [35] A. Bollin and D. Rauner. "Formal specification comprehension the art of reading and writing Z". Proceedings 2nd FME Workshop on Formal Methods in Software Engineering. Hyderabad, India, pp. 3-9. 2014.
- [36] S. Wolff. "Scrum goes formal: Agile methods for safety-critical systems". Proceedings First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches. Zurich, Switzerland, pp. 23-29. 2012.
- [37] M. Schrap and M. Peters. "Semantic annotation of a formal grammar by Semantic Patterns". Proceedings 4th International Workshop on Requirements Patterns. Karlskrona, Sweden, pp. 9-16. 2014.
- [38] M. Azeem, M. Ahsan and K. Noreen. "Specification of e-Health System using Z: A motivation to formal methods". Proceedings International Conference for Convergence of Technology. Pune, India, pp. 1-6. 2014.
- [39] S. Tamrakar and A. Sharma. "Comparative study and performance evaluation of formal specification language based on Z, B and VDM tools". International Journal of Scientific & Engineering Research. Vol. 6, Issue 9, pp. 1540-1543. 2015.
- [40] T. de Sousa, J. Almeida, S. Viana and J. Pavón. "Automatic analysis of requirements consistency with the B method". Software Engineering Notes. Vol. 35, Issue 2, pp. 1-4. 2010.
- [41] M. Bhavsar. "Analysis of Multiagent Based Interactive Grid Using Formal Methods - A Reliable Approach". Proceedings 3rd International Conference on Emerging Trends in Engineering and Technology. Goa, India, pp. 462-466. 2010.
- [42] G. Honghao. "Based on formal methods in trustable software requirements engineering". Proceedings International Conference on Internet Technology and Applications. Wuhan, China, pp. 1-4. 2010.
- [43] R. Hassan, M. Eltoweissy, S. Böhner and S. El-Kassas. "Formal analysis and design for engineering security automated derivation of formal software security specifications from goal-oriented security requirements". IET Software. Vol. 4, Issue 2, pp. 149-160. 2010.
- [44] J. van der Poll. "Formal methods in software development: A road less travelled". South African Computer Journal, Issue 45, pp. 40-52. 2010.
- [45] M. Singh. "Formal Specification of Common Criteria Based Access Control Policy Model". International Journal of Network Security. Vol. 11, Issue 3, pp. 139-148. 2010.
- [46] A. Cimatti, M. Roveri, A. Susi and S. Tonetta. "Formalization and Validation of Safety-Critical Requirements". Proceedings Workshop on Formal Methods for Aerospace. Eindhoven, Netherlands, pp. 68-75. 2009.
- [47] H. Barringer, A. Groce, K. Havelund and M. Smith. "An entry point for formal methods: Specification and analysis of event logs". Proceedings Workshop on Formal Methods for Aerospace. New Mexico, USA, pp. 16-21. 2011.
- [48] M. Bishop, B. Hay and K. Nance. "Applying formal methods informally". Proceedings of the Annual Hawaii International Conference on System Sciences. Kauai, USA, pp. 1-8. 2011.

- [49] C. Fernández, M. Ambrosio, G. Andrade, G. Cruz, M. José, R. Ortiz y H. Sánchez. "Métodos formales aplicados en la industria del software". *Temas de Ciencia y Tecnología*. Vol. 15 N° 43, pp. 3-12, 2011.
- [50] E. Serna. "Formal Methods: Perspective and future application". *III Jornadas de Investigación de la Facultad de Ingenierías*. Medellin, Colombia, pp. 64-68. 2011.
- [51] N. Ibrahim, V. Alagar and M. Mohammad. "Specification and Verification of Context-dependent". *Proceedings of the 20th international conference on World Wide Web*. Hyderabad, India, pp. 17-33. 2011.
- [52] B. Ammar and K. Abdallah. "Towards the formal specification and verification of multi-agent based systems". *International Journal of Computer Science Issues*. Vol. 8, Issue 4, pp. 200-210, 2011.
- [53] A. Bollin. "Do you speak Z? Formal methods under the perspective of a cross-cultural adaptation problem". *Proceedings 1st Workshop on Formal Methods in Software Engineering*. San Francisco, USA, pp. 8-14. 2013.
- [54] E. Serna and A. Serna. "Challenges and opportunities of research in formal methods". *XII Conferencia Iberoamericana en Sistemas, Cibernética e Informática*. Orlando, USA, pp. 1-6. 2013.
- [55] J. Atlee, S. Beidu, N. Day, F. Faghih and P. Shaker. "Recommendations for Improving the usability of formal methods for product lines". *Proceedings 1st FME Workshop on Formal Methods in Software Engineering*. San Francisco, USA, pp. 43-49. 2013.
- [56] E. Serna and A. Serna. "Formal Specification - Present and Future". *Proceedings Informática 2013*. La Habana, Cuba, pp. 1-13. 2013.
- [57] X. Chen. "Research on the implementation of internal control in enterprise information system by domain analysis and formal methods - A case study of sales activities internal control under Chinese enterprise environment". *Proceedings Fourth World Congress on Software Engineering*. Hong Kong, China, pp. 171-175. 2013.
- [58] L. Chan, R. Hexel and L. & Wen. "Rule-based behavior engineering: Integrated, intuitive formal rule modelling". *Proceedings 22nd Australian Software Engineering Conference*. Melbourne, Australia, pp. 20-29. 2013.
- [59] R. Diallo. "The need for usable formal methods in verification and validation". *Proceedings Winter Simulation Conference*. Washington, USA, pp. 1257-1268. 2013.
- [60] M. Noaman, I. Alsmadi and A. Jaradat. "The specifications of E-Commerce secure system using Z language". *The Research Bulletin of Jordan ACM*. Vol. 2, Issue 3, pp. 127-131. 2013.
- [61] T. Wu, Y. Dong and N. Hu. "Formal specification and transformation method of system requirements from B Method to AADL Model". *Proceedings 17th International Conference on Computational Science and Engineering*. Chengdu, China, pp. 1621-1628. 2013.
- [62] E. Serna and A. Serna. "Formal specification in context: Current and future". *Ingeniare. Revista Chilena de Ingeniería*. Vol. 22 N° 2, pp. 243-256. 2014.
- [63] E. Serna and A. Serna. "Formal methods in context". *XIII Conferencia Iberoamericana en Sistemas, Cibernética e Informática*. Orlando, USA, pp. 61-66. 2014.
- [64] M. Azeem, M. Ahsan, N. Minhas and K. Noreen. "Specification of e-Health system using Z: A motivation to formal methods". *Proceedings International Conference for Convergence of Technology*. Pune, India, pp. 1-6. 2014.
- [65] Y. Li, X. Pan, T. Hu, S. Sung and H. Yuan. "Specifying complex systems in Object-Z: A case study of petrol supply systems". *Journal of Software*. Vol. 9, Issue 7, pp. 1707-1717. 2014.
- [66] E. Serna and A. Serna. "Perspective and application of the formal methods". *XIII Conferencia Iberoamericana en Sistemas, Cibernética e Informática*. Orlando, USA, pp. 67-72. 2014.
- [67] A. Singh and D. Yadav. "Formal specification and verification of total order broadcast through destination agreement using e Vent-B". *International Journal of Computer Science & Information Technology*. Vol. 7, Issue 5, pp. 85-95. 2015.
- [68] S. Walter. "Towards formalized model-based requirements for a seamless design approach in safety-critical systems development". *Proceedings 18th International Symposium on Real-Time Distributed Computing Workshops*. Auckland, USA, pp. 111-115. 2015.