# A Software Project That Partially Failed: A Small Organization That Ignored the Management and Technical Practices of Software Standards

**Claude Y. Laporte and Guillaume Verret,** École de technologie supérieure

**Mirna Muñoz,** Centro de Investigación en Matemáticas

*We explore what went wrong when a small organization ignored the proven practices of a software engineering standard and what should have been done to meet the needs of this organization if the practices of software engineering standards have been used.*

Software engineering standards are sources of codified knowledge. Studies have demonstrated the benefits of them, such as product interoperability, increased productivity, market share gains, and improved interaction with stakeholders such as enterprises, government organizations, and the public. Standards and associated technical documents could be considered a form of technology transfer, and, if the right standards are selected and used correctly, they should have economic impacts in an organization. Unfortunately,

*Digital Object Identifier 10.1109/MC.2023.3253979*
*Date of current version: 3 May 2023*

EDITOR **RICCARDO MARIANI**
NVIDIA; rmariani@nvidia.com

process standards, unlike other engineering disciplines based on the laws of nature, do not guarantee a successful project (for example, all functionalities, all quality characteristics, within budget and schedule), but this is not a good reason to ignore them.

## INTRODUCTION

Besides, implementing international standards in very small entities (VSEs), that is, private or public organizations having up to 25 people, can be a path with many obstacles due to the effort required to achieve a correct implementation. The typical characteristics of VSEs are as follows: a) they lack previous experience in the use of documented development processes and the implementation of software engineering standards, b) the pressure they have to face to work harder to survive in the software market, c) they have few employees with little or no experience in the use of international standards, and d) they do not have the financial resources to improve their development process.[1]

The implementation of software engineering standards in very small organizations is critical because they represent a significant percentage of software organizations worldwide. They are often suppliers to small and medium enterprises and larger organizations. Therefore, the development of high-quality products or services is fundamental to their survival and growth.[2]

## BACKGROUND

A nonprofit organization mandated a supplier to develop a new transactional website to provide paid services to more than 400 members and partners. To protect the confidentiality of the nonprofit organization, the name *Acme* is used. This small organization was a "naïve" software customer, that is, Acme had a lack of experience, was overly trusting its supplier and was also lacking experience in managing a software supplier. This is not a condescending remark. Acme, like thousands of public or private organizations, was a user of software, it had no experience in documenting detailed software functionalities and software quality characteristics (for example, performance, usability, security). Acme was a typical customer that will "know what it wants when it will see it."

The implementation of software engineering standards in very small organizations is critical because they represent a significant percentage of software organizations worldwide.

A budget of US$90,000 was approved by the managers of the nonprofit organization. Unfortunately, the newly hired project manager (PJM) did not find any document (for example, e-mail, minutes of meeting) that justified that budget. The marketing director (MD) of Acme played a major role in the project. The MD wrote the request for proposals (RFPs) for the new website, selected five companies that were invited to submit a proposal, and he selected three companies for a presentation of their proposal to the selection committee of Acme.

During the selection process of the supplier, a member of the supplier selection team, very familiar with the business domain of Acme, noted two weaknesses, in a similar project, of the supplier that was later selected: web design and user experience design and user interface design. That supplier had previously developed a content management system (CMS) for organizations like Acme. That was one determining factor for the selection of that supplier.

The new PJM tried to use an ISO/IEC 29110 engineering and management guide,[3] described next, within Acme and with the supplier of the new website. Since the ISO/IEC 29110 was not cited in the RFP, members of Acme were reluctant to use that standard. In addition, the MD of Acme did not agree to use the standard because he did not know it. The new PJM decided to use the ISO/IEC 29110 informally as a reference during the project, for example, to compare the actual execution of the project with the project management (PM) and the software implementation (SI) processes of the ISO/IEC 29110.

## SOFTWARE ENGINEERING STANDARDS

Software engineering, like other engineering disciplines, is based on the use of well-defined practices for ensuring the quality of the products or services offered. There is a wide portfolio of IEEE and ISO standards that covers all aspects of software life cycle development, maintenance, and management. There are more than 200 published systems and software engineering ISO standards developed by experts of more than 60 countries and professional organizations such as the IEEE. As an example, the ISO/IEC/IEEE 12207[4] provides, for an organization or a project, processes that can be employed for defining, controlling, and improving software life cycle processes and, the ISO/IEC/IEEE 29148[5] provides the processes that result in the requirements throughout the life cycle.

To help meet the needs of VSEs, the International Organization for

Standardization and the International Electrotechnical Commission jointly published the four-stage road map ISO/IEC 29110 series of standards and guides. These publications target VSEs, ranging from startups to grownups, with little or no experience or expertise in selecting the appropriate processes from systems or software engineering lifecycle standards, such as ISO/IEC/IEEE 12207, and tailoring them to a project's needs.[6]

## A LIGHT SOFTWARE ENGINEERING STANDARD

The ISO/IEC 29110 series targets small private or public entities with little or no experience or expertise in selecting the appropriate processes from lifecycle standards and tailoring them to a project's needs. The ISO/IEC 29110 Basic guide targets VSEs developing a single product with a single team; it defines software implementation (SI) and project management (PM) processes.

As illustrated in Figure 1, a customer provides a statement of work, or a description of the functionalities and quality characteristics (for example, usability, security) required within a specified time frame and budget, as an input to trigger the PM process. Then, a project plan developed by a supplier guides the execution of software requirements analysis, architecture and detailed design, construction, integration and test, and the product delivery activities. Finally, the PM process delivers the software configuration—that is, the complete set of software artifacts that comprise the product, including user documentation, code, and so on—to the customer and obtains the customer's acceptance to formalize the end of the project. Although the Basic guide might give the impression of a waterfall development cycle, the ISO/IEC 29110 series isn't intended to dictate the use of any particular life cycle, whether waterfall, iterative, incremental, evolutionary, or agile.

## WHAT HAS BEEN DONE AND WHAT SHOULD HAVE BEEN DONE

Many risks and problems faced by Acme could have been either avoided or greatly attenuated if a minimal number of management and technical practices had been used. In the following paragraphs, we briefly describe what Acme should have done as well as the impacts of management and technical decisions taken.

### RFP of Acme and the proposal of the selected supplier

A minimal RFP provides a description of work to be done related to software development. A more detailed RFP may include: a product description, the purpose of the product, the general customer requirements, a scope description of what is included and what is not and, the list of products (for example, code, documentation such as a user guide) to be delivered to the customer.

As mentioned previously, Acme, being a naïve customer, did not know how to write functional and nonfunctional requirements. Acme is the typical customer "that will know what it needs when it will see it." The RFP of Acme listed needs typical of a user that knows almost nothing about the functionalities of a transactional website and the importance of software quality characteristics. In the RFP, Acme listed the following needs:

- overall characteristics of the site (a site that is simple to navigate, efficient in the organization of subjects and themes, intuitive when performing searches, visually elegant, and autonomous in its management)
- description of the cost of each phase and modules of the website
- schedule of activities
- list of top-level functionalities (for example, description of services, interactive map, events, newsletters, publicity, blog).

Acme also requested that the proposals include a presentation of the preselected suppliers, identified the members of the team assigned to the project as well as its expertise with the business domain of Acme, a detailed pricing by module and implementation phases, the payment terms and a certification that once completed, the new website belongs to Acme.

The selected supplier provided a two-page proposal listing the main tasks and associated costs (for example, analysis–US$9,000, programming–US$50,000, newsletter–US$4,000,

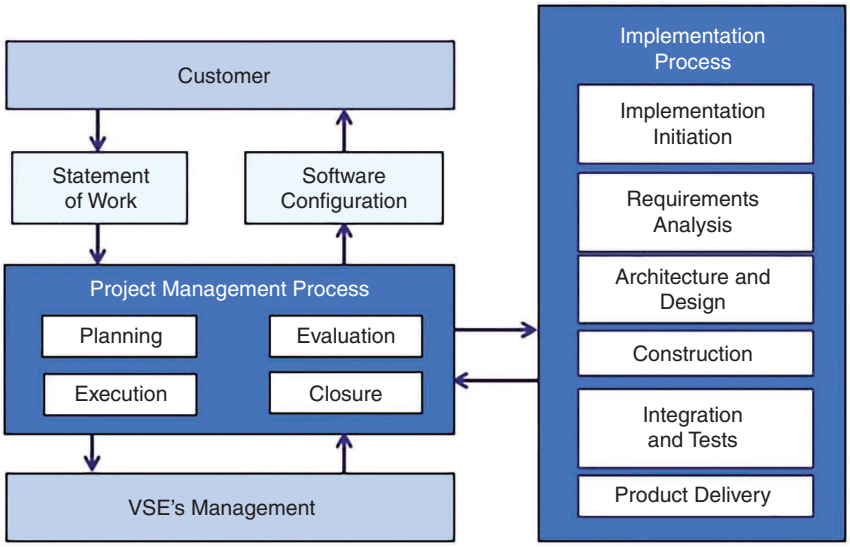


FIGURE 1. Overview of processes and activities of an ISO/IEC 29110 Guide.[3]

hosting–US$4,000), and a 24-week schedule.

### Project plan

A minimal project plan should present how the management and technical processes and activities will be executed to provide all of the functionalities and quality characteristics within budget and schedule, and a set of deliverables (for example, website, documentation such as user guide).

A typical project plan includes the following elements: a list of work products to be delivered to the customer, a list of tasks such as reviews (for example, verification, validation) with the customer and the development team, an estimated duration of tasks, resources (for example, humans, materials, standards, equipment and tools), a schedule of the project tasks (expected start and completion date for each task, and the relationship and dependencies of the tasks, an estimation of effort and cost, an identification of project risks, a version control strategy and the delivery instructions (for example, elements required for product release identified (that is, hardware, software, documentation), the delivery requirements, a sequential ordering of tasks to be performed, and an identification of all delivered software components with version information).

To minimize bad surprises for a customer (for example, deliverables, schedule, and cost), the ISO/IEC 29110 Guide specify that the project plan must be reviewed and approved by the supplier and the customer. Unfortunately, the MD decided that its RFP and the two-page proposal of the selected supplier were acceptable as a project plan!

It is well known that all software development projects have management and technical risks. Therefore, one element of the PM process is the identification and monitoring of project risks. For this project, risks have been ignored, even the two weaknesses of the selected supplier expressed by one experienced member of the selection team (that is, web design and user experience design and user interface design). Since one objective of Acme was to provide services to hundreds of customers, these two risks could greatly impact the quality of the site.

### Needs of the customer

The RFP written by the MD illustrates the naivety of the customer. The RFP states that the site had to be simple to navigate, intuitive when performing searches, visually elegant, and autonomous in its management. The RFP also requested that the cost of each phase and modules of the website and a schedule of the activities be provided. Finally, the RFP listed the desired functionalities (for example, description of services, interactive map, events, newsletters, publicity, blog).

Acme did not provide, as described in the ISO/IEC 29110 Guide, any nonfunctional requirements (for example, response time, throughput, execution time, storage capacity, number of simultaneous users) and the specifications of quality characteristics (for example, security, portability) as defined in the ISO/IEC 25010 standard.[8] Naive customers are those that are users of a software, most customers have no experience in the development of a software. They are not aware of the important of the quality characteristics that are at least as important as the functionalities needed. A supplier could benefit this lack of experience of customers to its favor. As an example, Acme listed quality characteristics, such as visually elegant, that would be, from both sides, difficult to objectively validate or challenge.

To minimize bad surprises to naive customers, the Basic Guide provides a template of the requirements specification that a supplier must complete. The Basic Guide also provides a task, see Table 1, requiring the supplier to review and validate them with the customer.

## WHAT SHOULD HAVE BEEN DONE AND WHAT WAS DONE

Unfortunately, there were large gaps between what should have been done and what was done by Acme and its supplier. A few issues, such as progress review and the management of changes, are presented to illustrate that the knowledge documented in software standards has been ignored in many areas.

### Progress review meetings

Progress review meetings are conducted periodically, between the customer and its supplier, to evaluate the progress of the project against commitments documented by the supplier (for example, in the project plan, requirements document) about the following issues:

**TABLE 1.** Validation of requirements by the customer.[7]

| Roles | Task | Input Work Product | Output Work Products |
|---|---|---|---|
| Customer | SI.2.4 Validate and obtain approval of the requirements specification. | • Requirements specification [*verified*] | • Requirements specification [*validated*] |
| Analyst | Note 1 : Validate that requirements specification satisfies needs and agreed upon expectations, including the user interface usability.<br>Note 2 : The results found are documented in a validation record and corrections are made until the document is approved by the customer. | | • Validation record |

- status of actual tasks against planned tasks
- status of actual results against established objectives/goals
- status of actual resource allocation against planned resources
- status of actual cost against budget estimates
- status of actual time against planned schedule
- status of actual risk and mitigation against previously identified
- record of any deviations from planned tasks and reason why

During a progress review meeting, the issues discussed are documented, and the PJM of the customer and the supplier sign the report of the issues discussed during the meeting. Besides, during this meeting decisions are taken to address any problems identified. The supplier must determine and document the tasks needed to correct a deviation or a risk concerning the accomplishment of the project plan as follows:

- identifies the initial problem
- defines a solution
- identifies corrective actions taken
- identifies the person responsible for completion of defined actions.

### Changes and change requests

All software development projects have changes during development, either changes requested by the customer that wants to add, delete, or modify functionalities or quality characteristics or changes requested by the supplier. For this project, communications about changes were sometimes done by telephone, sometimes by e-mail, or on the project management platform. A customer, such as Acme, that has no experience in software development could easily agree over a short phone conversation about a change that could look minor or with no impact on quality characteristics, functionalities, budget, or schedule.

At the beginning of a new project, the customer–supplier relationships are usually cordial. But, when functional or nonfunctional problems are reported, or schedule and budget get challenged, phone conversations are usually not an appropriate mechanism for a naïve customer to make sound decisions. A minimal procedure about the handling of a change, requested by a supplier, allows the customer to take time to analyze its impacts (for example, functionalities, qualities, schedule, budget) before deciding (for example, accept as proposed, accept with modifications, postpone, reject).

The following elements are the minimum information of a change request:

- the purpose of the change
- the requester contacts information (for example, customer, project manager), the impacted software components
- the impact to operations of existing software, the impact to associated documentation
- the request state (for example, initiated, evaluated, accepted, and rejected).

In case of a conflict that leads to litigation between the customer and the supplier, a telephone conversation may not be accepted, as evidence, by a judge, unless at least another person confirms the conversation. Documents are a stronger proof in front of a judge. To protect itself, Acme should have put in writing issues discussed with the supplier. A short e-mail between a customer and a supplier to confirm a conversation is an acceptable proof.

### Development of the software

As described in the RFP of Acme or the two-page proposal, the supplier was under no obligation to develop or deliver software documentation (for example, architecture, tests cases, user guide). The only obligations of the supplier were to develop and host the website.

The documentation, that should have been produced by the supplier, would have been used for software acceptance by Acme and for the maintenance activities (for example, correction of defects, deletion, addition and modification of functionalities and the quality characteristics (for example, security) over the life of the website.

### Traceability of needs, requirements, and tests

Traceability is the ability of the customer or its supplier to trace work products (for example, list of needs, requirements, architecture, code, tests) across the development and maintenance and operation activities. As an example, for a small project, a spreadsheet could record the relationship between the artifacts developed. During the maintenance and operation activities of the website, a change request submitted by Acme could be linked to a need or a requirement of the traceability spreadsheet.

It is very likely that during the many years of operation of the website, there will be rotations among the staff of the supplier (for example, arrival, departure, promotion). Most probably, some knowledge about the software will be lost. A traceability spreadsheet would facilitate the analysis of the ramifications of a change request from the customer or the correction of a defect. The spreadsheet could also have been used by the PJM to track the progress of the project (for example, a need has been coded, tested), during progress review meetings, and to better understand the impact of a change on cost, schedule, functionalities, and quality characteristics.

### Acceptance of the website by Acme

Since the quality characteristics documented by Acme (for example, simple to navigate, visually elegant) were difficult to objectively validate

or challenge, therefore Acme did not have a strong leverage to challenge the supplier. Acme could not rely on a requirements specification document to evaluate if its needs had been objectively met, since the supplier did not provide it. A traceability table or spreadsheet could have been used by the new PJM to verify that all needs and requirements have been fulfilled and successfully tested and that the latest version of all requested work products ware delivered before accepting them and authorizing the final payment to the supplier.

## UNINTENDED SHORT-TERM AND LONG-TERM IMPACTS

In the event of the bankruptcy of the supplier or a takeover by a larger organization, Acme could have had all of the software work products, listed in the ISO/IEC 29110 guide, "safeguarded" externally to the supplier site with an escrow with a notary or a lawyer. An escrow is a mechanism that keeps in the custody of a mutually agreed third party (for example, attorney, notary) the source code as well as all requested documentation until specified conditions (for example, bankruptcy) have been fulfilled (definition adapted from ISO/IEC/IEEE 24765:2017[9]).

A major reason why Acme selected its supplier was the availability of a proprietary CMS software. If Acme ever decides to break the business relationships with its supplier, Acme may have to spend a large sum of money and wait for many weeks to get another supplier ready to support and host its transactional website.

About maintainability, since no documentation had been demanded by Acme, a new supplier will have to do some reverse engineering activities to document the architecture and the requirements of the website. It may even be more productive, for a new supplier, to redevelop, almost from scratch, a new website.

## DISCUSSION

Acme is one of the thousands of private and public organizations that are naïve customers. Such customers are often not equipped to manage a development contract for a software needed for their day-to-day operation. Unfortunately, software engineering standards documenting codified knowledge and publicly available, for many decades, are not used, or are ignored, by many private and public organizations. As an example, two cases are listed:

- Software inspections, initially developed by Fagan at IBM in the early 1970s, documented in the IEEE-1028 standard[10] are still not used to their full potential as reported by Fagan in 2002: "Even 30 years after its creation, it is often not well understood and more often, poorly executed—yielding results that are positive, but well below their potential."[11]
- A recent survey of 90 requirements engineering practitioners about the ISO/IEC/IEEE 29148 requirements engineering standard,[5] reported that about 47% of the respondents, working as requirements engineers or business analysts, did not know the ISO/IEC/IEEE 29148 and about 24% of the respondents never used the standard. Even if most respondents had university degrees, unfortunately universities take only the fifth place when it comes to where the respondents learned about the standard. Only 22% of respondents cited university studies as a source for the knowledge of requirements engineering-related standards.[12]

Customers, like Acme, rely on the expertise of a supplier to develop a software product that will be used daily to provide paid services and information to its numerous members. Malpractice could be defined as any inappropriate, wrong, illegal, or careless actions that a professional does while working. If a professor teaching future software engineers could be accused of malpractice for not teaching standards to future software engineers, could a supplier or a developer, that ignore or do not use the practices published in standards, be accused of malpractice by its customer[13]?

Many customers and technical people underestimate the importance of a minimal project management process. Even if a supplier has competent developers, without a minimum number of project management tasks, a project may fail to meet all of the objectives of a customer (that is, functionalities, qualities, budget, and schedule). Unfortunately, when faced with problems, delays, and additional costs similar to Acme, unhappy customers may have to resort to litigations, that is, the process of taking a lawsuit against an organization to court, to recover some of the impacts to their operations.

Over the last decade, the use of software standards has been increasing by private and public organizations in the development of quality products within approved budget and schedule. As an example, about 700 VSEs in Thailand have obtained a certification to the ISO/IEC 29110. Many Thai VSEs are important as suppliers for many medium and large private and public organizations. In addition, since hundreds of Thai VSEs are using the same framework, they can easily team up and bid on large software development projects.

Unfortunately, implementing and using software engineering standards is not free and is not an easy task since resistance to change of managers and developers could either slowing down or even preventing the use of standards. The Acme case highlights the bad consequences and impact that a VSE can have if it avoids the use of software engineering standards.

Acme lacked the skills to properly identify its needs and manage a software project developed externally by a supplier. Acme has paid the supplier for

the development of the technical documents, for example, requirements, architecture, code, test cases. Acme could have "protected" its investment by demanding the delivery of documents defined in an ISO/IEC 29110 Guide.

Acme's decision not to use a standard such as the ISO/IEC 29110 internally and not to impose basic management and technical practices to the supplier led to several negative consequences that could have been avoided or reduced during development and maintenance over the many years of operation. Since the management and technical documents have not been demanded nor delivered and the website is using a proprietary CMS software, Acme is almost forced to depend on its supplier for many years.

If Acme and its supplier had used the management and engineering guide, such as the ISO/IEC 29110, the website project could have been a "win–win" during the development and the maintenance and operation over the life of the website. If Acme and the supplier had used the management and engineering guide, once it would be time to develop a second generation of the site, many documents produced, for the first generation, could have been reused for the development.

Software engineering standards are sources of codified knowledge extracted from thousands of successful and failed projects. Ignoring the lessons learned captured in standards, customers like Acme are almost doomed to repeat the same mistakes again. In an "Impact" column in IEEE Software,[14] the authors wrote, "We had been hoping that would follow the same trajectory as its older established cousins, such as civil engineering, but we have seen no real evidence of this." C

## REFERENCES


1. M. Muñoz, "Can undergraduates get the experience required by the software industry during their University?" in *New Perspectives in Software Engineering (CIMPS 2022)*, vol. 576, J. Mejia, M. Muñoz, Á. Rocha, and V. Hernández-Nava, Eds. Cham, Switzerland: Springer, 2023, pp. 152–161.
2. G. Ibarra, S. Vullinghs, and F. J. Burgos, *Panorama Digital de Las Micro, Pequeñas y Medianas Empresas (MiPymes) de América Latina 2021*. Santiago, Chile: GIA Consultores, 2021.
3. *Software Engineering-Lifecycle Profiles for Very Small Entities (VSEs)—Part 5-1-2: Management and Engineering Guide Generic Profile Group: Basic Profile*, ISO/IEC, ISO/IEC TR 29110-5-1-2:2011, 2011. [Online]. Available: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html
4. *Systems and Software Engineering—Software Life Cycle Processes*, ISO/IEC/IEEE 12207, International Organization for Standardization, Geneva, Switzerland, 2017.
5. *Systems and Software Engineering—Life Cycle Processes—Requirements Engineering*, ISO/IEC/IEEE 29148, International Organization for Standardization, Geneva, Switzerland, 2018.
6. C. Y. Laporte, M. Muñoz, J. M. Miranda, and R. V. O'Connor, "Applying software engineering standards in very small entities: From startups to grownups," *IEEE Softw.*, vol. 35, no. 1, pp. 99–103, Jan./Feb. 2018, doi: 10.1109/MS.2017.4541041.
7. C. Y. Laporte and R. V. O'Connor, "Systems and software engineering standards for very small entities: Accomplishments and overview," *Computer*, vol. 49, no. 8, pp. 84–87, Aug. 2016, doi: 10.1109/MC.2016.242.
8. *Systems and Software Engineering–Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models*, ISO/IEC 25010, International Organization for Standardization, Geneva, Switzerland, 2011.
9. *Systems and Software Engineering - Vocabulary*, ISO/IEC/IEEE 24765, International Organization for Standardization, Geneva, Switzerland, 2017.
10. *IEEE Standard for Software Reviews and Audits*, IEEE Standard 1028-2008, Aug. 2008.
11. M. Fagan, "A history of software inspections," in *Software Pioneers*, M. Broy and E. Denert, Eds. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 562–573, doi: 10.1007/978-3-642-59412-0_34.
12. X. Franch, M. Glinz, D. Mendez, and N. Seyff, "A study about the knowledge and use of requirements engineering standards in industry," *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3310–3325, Sep. 2022, doi: 10.1109/TSE.2021.3087792.
13. C. Y. Laporte and M. Munoz, "Not teaching software engineering standards to future software engineers-malpractice?" *Computer*, vol. 54, no. 5, pp. 81–88, May 2021, doi: 10.1109/MC.2021.3064438.
14. M. van Genuchten and L. Hatton, "Ten years of 'impact' columns—The good, the bad, and the ugly," *IEEE Softw.*, vol. 36, no. 6, pp. 57–60, Nov./Dec. 2019, doi: 10.1109/MS.2019.2932495.



**CLAUDE Y. LAPORTE** is an adjunct professor of software engineering at École de technologie supérieure, Montréal, QC H3C 1K3, Canada, and the lead editor of the ISO/IEC 29110 series of standards and guides. Contact him at claude.laporte@etsmtl.ca.

**GUILLAUME VERRET** received his master's degree in engineering from École de technologie supérieure, Montréal, QC H3C 1K3, Canada. Contact him at gllme.verret@gmail.com.

**MIRNA MUÑOZ** is a professor of software engineering at Centro de Investigación en Matemáticas, Zacatecas, 98160, Mexico, and coeditor of the ISO/IEC 29110 Agile Software Development Guide. Contact her at mirna.munoz@cimat.mx.