Putting Best Practices to Work

www.qualityprogress.com | September 2015

QP

QUALITY

The **Ishikawa Family Legacy**
p. 46

KEEP CALM AND PREPARE FOR ISO 9001:2015

Let our experts help p. 18

Plus:

How to select an ISO 9001 consultant p. 8

Retool audits using the Kano model p. 38

ASQ
The Global Voice of Quality™

QUALITY PROGRESS | SEPTEMBER 2015

ISO 9001:2015

VOLUME 48/NUMBER 9

Contents

FEATURES

18 STANDARDS
Keep Calm and Prepare for ISO 9001:2015
Standards experts provide basic analyses and map out five key changes included in ISO 9001:2015 to help make sense of what they mean, what your organization must do to meet the new requirements, and resources available to help you navigate the transition.

by Bill Aston, Susan L.K. Briggs, Charles A. Cianfrani, Deann Desai, Allen Gluck, Paul C. Palmes, Denise Robitaille and John E. "Jack" West

30 SOFTWARE QUALITY
A Hard Look at Software Quality
A pilot project proved how the ISO/IEC 25000 family of standards can be used to evaluate software quality and let developers better address maintenance issues.

by Moisés Rodríguez, Mario Piattini and Carlos Manuel Fernández

38 AUDITING
Amazing Audits
A new approach that incorporates the Kano model can help you evaluate the performance of your quality audit programs.

by Lance B. Coleman Sr.

46 HISTORY
Like Father, Like Son
A look back at the lasting contributions Ichiro Ishikawa and Kaoru Ishikawa made to total quality management and the world of quality.

by Gregory H. Watson

ONLY @
www.qualityprogress.com

- **Small Business Challenge**
  A sidebar for this month's cover story ("Keep Calm and Prepare for ISO 9001:2015," pp. 18-28) examines the resources that small organizations can use to help with the transition to ISO 9001:2015.
- **Audit Add-Ons**
  More figures to illustrate how to evaluate your audit programs, the subject of "Amazing Audits," pp. 38-45.
- **Stay in the Loop**
  Subscribe to the new *Standards Connection* enewsletter and stay on top of the latest ISO 9001:2015 developments.
- **Back to Basics**
  Translated into Spanish.
- **Talk Back**
  Rate and comment on this month's four feature articles.

DEPARTMENTS

6 **LogOn**
- Advice on developing high-quality suppliers.

8 **Expert Answers**
- Tips on initial ISO 9001 implementation.
- Choosing the right ISO 9001 consultant or registrar.

12 **Keeping Current**
- Dealing with drones.

14 **Mr. Pareto Head**

66 **QP Toolbox**

68 **QP Reviews**

COLUMNS

5 **Up Front**
Stress relief.

53 **Measure for Measure**
Updates needed to the occupation classification list.

56 **Quality in the First Person**
Learning early on to always do things right—not just when others are watching.

58 **Career Corner**
In job searches, do your homework on prospective employers.

60 **Statistics Roundtable**
Including nuisance factors during experimentation.

71 **One Good Idea**
Using fault data spectrums to check corrective actions.

BONUS
72 **Back to Basics**
Are standard certifications always necessary?

SPECIAL SECTION
ASQ'S STANDARDS AND AUDITING GUIDE p. 62

NEXT MONTH

- STRUCTURED DECISION MAKING
A new five-step process to evaluate and balance competing objectives.

- SUPPLIER DEVELOPMENT
How one energy company uses an automotive standard to assess and manage its suppliers.

ASQ's Vision: By making quality a global priority, an organizational imperative and a personal ethic, the American Society for Quality becomes the community for everyone who seeks quality technology, concepts or tools to improve themselves and their world.

Quality Progress (ISSN 0033-524X) is published monthly by the American Society for Quality, 600 N. Plankinton Ave., Milwaukee, WI 53203. Editorial and advertising offices: 414-272-8575. Periodicals postage paid at Milwaukee, WI, and at additional mailing offices. Institutional subscriptions are held in the name of a company, corporation, government agency or library. Requests for back issues must be prepaid and are based on availability: ASQ members $17 per copy; nonmembers $25 per copy. Canadian GST #128717618, Canadian Publications Mail Agreement #40030175. Canada Post: Return undeliverables to 2835 Kew Drive, Windsor, ON N8T 3B7. Prices are subject to change without prior notification.

Postmaster: Please send address changes to the American Society for Quality, PO Box 3005, Milwaukee, WI 53201-3005. Printed in USA.

QP
QUALITY PROGRESS

Mail
Quality Progress/ASQ
600 N. Plankinton Ave.
Milwaukee, WI 53203

Telephone	Fax
800-248-1946	414-272-1734
414-272-8575	

Email
Follow protocol of first initial and full last name followed by @asq.org (for example, vellifson@asq.org).

Article Submissions
Quality Progress is a peer-reviewed publication with 85% of its feature articles written by quality professionals. For information about submitting an article, call Valerie Ellifson at 800-248-1946 x7373, or email manuscripts@asq.org.

Author Guidelines
To learn more about the manuscript review process, helpful hints before submitting a manuscript and QP's 2016 editorial planner, click on "Author Guidelines" at www.qualityprogress.com under "Tools and Resources."

Photocopies, Reprints And Microform
Article photocopies are available from ASQ at 800-248-1946. To purchase bulk reprints (more than 100), contact Barbara Mitrovic at ASQ, 800-248-1946. For microform, contact ProQuest Information and Learning, 300 N. Zeeb Road, Ann Arbor, MI 48106, 800-521-0600 x2888, international 734-761-4700, www.il.proquest.com.

Membership and Subscriptions
For nearly 70 years, ASQ has been the worldwide provider of information and learning opportunities related to quality. In addition, ASQ membership offers information, networking, certification and educational opportunities to help quality professionals obtain practical solutions to the many problems they face each day. Subscriptions to *Quality Progress* are one of the many benefits of ASQ membership. To join, call 800-248-1946 or see information and an application on p. 65.

List Rentals
Orders for ASQ's member and nonmember buyer lists can be purchased by contacting Michael Costantino at the Infogroup/Edith Roman List Management Co., 402-836-6626 or fax 845-620-1885.

A HARD LOOK *at Software Quality*

In 50 Words Or Less

- Software products are more scrutinized today than ever because problems can be costly.
- One pilot program used the ISO/IEC 25000 family of standards as a framework to evaluate software product quality.
- The structure let developers address maintenance issues, as well as product quality and issues related to development processes.

Pilot program uses ISO/IEC 25000 family to **evaluate, improve and certify** software products

by Moisés Rodríguez, Mario Piattini and Carlos Manuel Fernández

SOFTWARE QUALITY IS a key challenge for organizations because of its impact on final costs. At the same time, software quality can be a competitive differentiator an organization can offer to its customers.

The importance of software quality is accentuated when you consider the losses organizations can incur when quality problems arise in software development projects. In fact, only 39% of software projects finish on time with the planned resources and with acceptable quality, according to the Standish Group's latest report "The CHAOS Manifesto."[1] See Figure 1 (p. 32).

To highlight the importance of software quality, a laboratory in Spain performed a pilot program to evaluate, improve and certify the quality of software products.

Software evaluation

Activities related to software quality have become increasingly important, especially as more organizations outsource their software functions. This means client organizations must assess and monitor the quality of the products they receive from software development organizations. In turn, these software development organizations must have the necessary resources to ensure the products developed will meet client expectations.

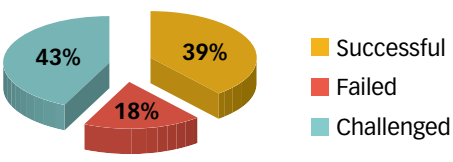
Despite the proliferation of software quality process certifications (for example, *ISO/IEC 15504 Information technology—Process assessment and certifications* from the Capability Maturity Model Integration Institute), there is little evidence conformance to process standards actually guarantees good products. In fact, critics suggest process standards guarantee only the uniformity of output and can possibly institutionalize the production of mediocre or bad products.[2]

The idea that software evaluations should be based on direct evidence about a product's attributes rather than circumstantial evidence about processes used to build a product[3] is becoming more widespread. An ever-increasing number of organizations therefore are concerned not only about the quality of the processes, but also the quality of the products they develop and acquire. This is because they may have serious quality problems after the product has been implanted into their servers.

The new ISO/IEC 25000 family of standards, also known as SQuaRE (Software Product Quality Requirements and Evaluation), appears to meet these needs. ISO/IEC 25000 aims to create a common framework to evaluate software product quality, replacing ISO/IEC 9126 and ISO/IEC 14598 and becoming the cornerstone of this area of software engineering.

ISO/IEC 25000 is composed of several parts: ISO/IEC 25040[4] defines the process of evaluating software product quality, and ISO/IEC 25010[5] determines the software product characteristics and subcharacteristics that can be evaluated (see Figure 2).

Project resolution results / FIGURE 1



Results from 2012 CHAOS research

The Spanish Association for Standardization and Certification (Asociación Española de Normalización y Certificación, or AENOR), which is a member of the International Certification Network (IQNet), in collaboration with Alarcos Quality Center (AQC), a spin-off of the University of Castilla-La Mancha, conducted the evaluation and certification pilot project using the new ISO/IEC 25000 family of standards.

AQC and AENOR are independent organizations. AQC is responsible for evaluating software product quality, thus permitting organizations to meet a quality level and improve their software products. AENOR, as an auditing entity, is responsible for certifying the software product in cases in which organizations, in addition to improving their products, wish to become certified.

Initially, this pilot project mostly focused on the quality characteristic of maintainability because:

- Maintenance is one of the most expensive phases in the software life cycle and, in some cases, can reach up to 60% of costs.
- Maintainability is one of the features most frequently requested by software customers. It must be possible for customers to evolve software products they acquire—either by themselves or via a third party.
- Maintenance work on products with little maintainability is more likely to introduce new bugs into software products.

The evaluation and certification based on this family of standards provides software development organizations and organizations that acquire such software a set of benefits, which are shown in Figure 3 (p. 34).

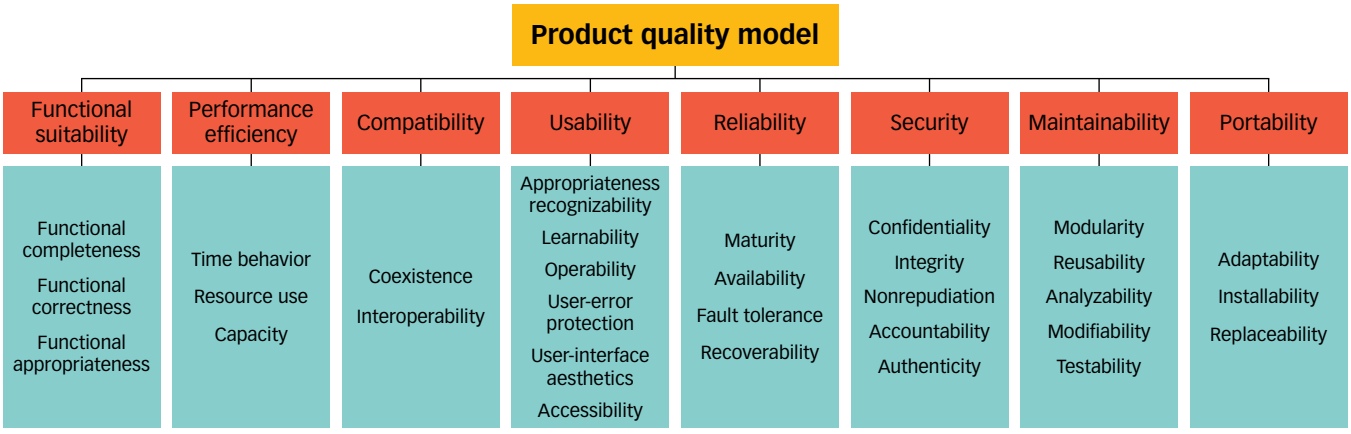
Five activities

For this project, it was first necessary to create the AQC Lab using an agile approach,[6] responsible for carrying out the software product quality evaluation process. This laboratory performs evaluations by conducting the five activities proposed in ISO/IEC 25040:

1. **Establish evaluation requirements:** This activity's aim is to establish an agreement between the laboratory and client about which purpose and quality requirements will be considered for the product evaluated. To this end, parts of the software product that will be evaluated are identified, and the rigor of the evaluation that will be performed is clearly defined.

   The results are an initial version of the evaluation plan along with a list of requirements. The pilot project's scope was the complete software products. Although some organizations had defined quality requirements for their software products (such as low complexity or good documentation), a decision was made to expand them to the requirements associated

Software product quality model from ISO/IEC 25010 / FIGURE 2



with the laboratory model.

2. **Specify the evaluation:** This activity's objective is to select the quality metrics that will be measured in the software product (that is, to create the evaluation module). The decision criteria that establish thresholds for metrics and indicators used to assign values to the subcharacteristics and quality characteristics of the product also are defined.

   The result of this activity is a refined version of the evaluation plan with metrics and defined criteria. In the pilot project, all the properties of the quality laboratory model (covered later) were selected as the thresholds established by the laboratory to certify the product.

3. **Design the evaluation:** This activity's purpose is to plan the measurement and evaluation activities that must be conducted, taking into account the prespecified scope and availability of laboratory resources. The result of this activity is the final version of the evaluation plan, with details of the dates of evaluation activities and the specific resources involved.

   In the pilot project, an initial assessment was planned for each product. This assessment lasted two weeks. Each organization requested a month to correct its software products. Finally, the laboratory planned a second evaluation of each product, which took one week.

4. **Execute the evaluation:** This activity's purpose is to perform measurements on the software product and apply the decision criteria specified in the second activity. Thanks to the automated laboratory environment, this activity can be performed quickly and reliably, and the evaluation results are stored in a secure repository, which can be accessed via a web application that allows visualization and analysis.

   The result of this activity is a set of values obtained for all metrics, indicators, subcharacteristics and quality characteristics.

5. **Conclude the evaluation:** This activity's purpose is to analyze the results obtained in the previous activity to ensure no errors occurred during the evaluation and to generate the final evaluation report. This report is the main output of the evaluation process and will be delivered to the customers to allow them to review it and, if necessary, express disagreement.

Figure 4 (p. 34) shows sections of this evaluation report for one of the pilot project products. During this activity, the laboratory also performs the disposition of the evaluation data—that is, all records that no longer need to be maintained (for example, product source codes) are removed. This also occurs for confidentiality reasons because any data that the client does not wish to remain in the laboratory after the completed evaluation are returned or eliminated.

It is also necessary to consider that the family of ISO/IEC 25000 standards defines models and processes to assess the quality of software products, but it does not establish a correlation between the metrics and thresholds needed to identify the specific level of quality a software product has.

The laboratory, therefore, has defined not only a model and a quality process that are aligned with ISO/IEC 25000, but it also has identified a set of measurable quality properties from the software product's source code.

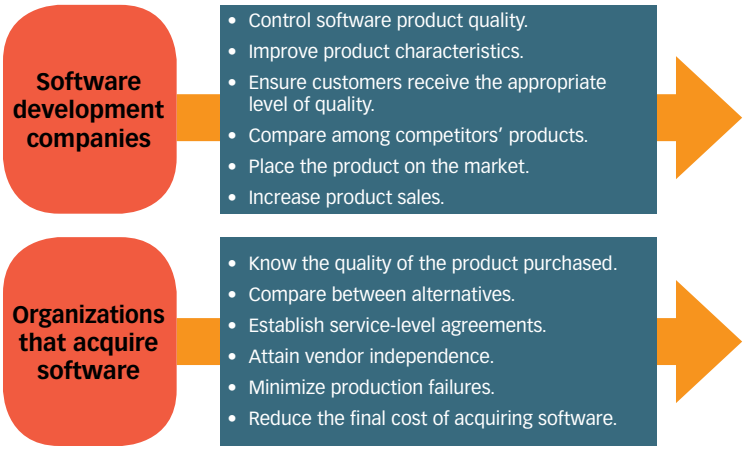
Quality properties

The objective of identifying these quality properties was not to form the largest set of these quality properties possible, but rather to assemble a whole group that was not controversial, based on previous studies and research, and accepted by the scientific community. The main properties are:

- **Violation of encoding rules**—This property is based on encoding rules for each language (such as Java code standard) and the percentage of violations. A set of related rules are checked for each quality subcharacteristic.
- **Code documentation**—This property refers to comments in the code that are used to explain the code's functionality. Well-documented code helps developers understand what it does. Comments also may influence the ability to reuse software because if you know the functionality of a module, you can reuse that module in another system.
- **Cyclomatic complexity**—This property is related to the difficulty involved in implementing, testing, understanding, maintaining or modifying a program. As this definition shows, the complexity is related to the following subcharacteristics: analyzability, modifiability and testability. In applications with high complexity, maintenance tasks require more effort and, therefore, are more costly.
- **Structuring**—This property refers to dividing the system into smaller parts. The quality of the system design is closely related to this property: The correct structure of a system at any level (for example, subsystems, packages and classes) facilitates development and maintenance.
- **Method size**—The size of a system directly affects its maintainability because a larger system obviously requires more maintenance. It is considered more appropriate to evaluate the size of the elements at a lower level by using, for example, methods that directly evaluate the overall size of the system. This is because a change does not usually affect the entire system, but rather a set of elements.
- **Duplicate code**—This property refers to code fragments repeated in different parts of the system. Duplicate code makes it difficult to modify a software application because solving a bug or making an improvement to the code requires changes to all parts of the system where the duplicate code appears.
- **Coupling**—This property indicates the degree of interdependence among software units (modules, functions, classes and libraries, for example). Generally, the lower the coupling in a software application, the better its design is considered because low coupling improves maintainability (if there is no coupling, changes to one unit do not affect other units) and increases the reusability of software units.
- **Cycles**—This property refers to the existence of cycles of dependence among the system packages. The acyclic dependencies principle states there must be no cycles in the dependency structure of a system. The existence of cycles in a system has a negative effect on its maintainability because a change that is made to one packet affects all the packets in the cycle.[7]
- **Cohesion**—This property indicates the degree of relationship between the elements in a module. A class has low cohesion when it performs several unrelated functions. The functionality provided by its methods thus has little in common with them. The analyzability, modularity and, in general, maintainability are negatively affected because these systems are more difficult to understand. They provide features that are unnecessary and changes in requirements that affect several modules.

The evaluation process, the quality model and the properties, along with a measurement environment, allow a laboratory to evaluate a software product and calculate its quality values. The measurement environment automates the evaluations by up to 90% and has three different levels:

Benefits of evaluating software product quality / FIGURE 3



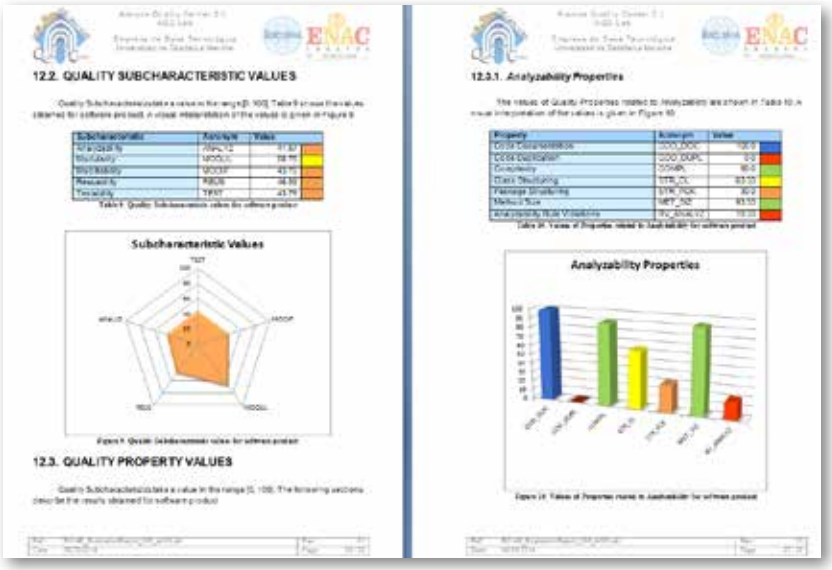
1. **Measuring tools**—These constitute the first level. Their mission is to analyze the source code and generate files (usually XML) containing information about metrics and violations of programming rules.

   The advantage of this level is that it can easily be expanded by adding new tools (an organization's own or via a third party) that analyze new programming languages or compute new metrics for other quality characteristics.
2. **Evaluation system**—This involves the intermediate level of the environment. Its objective is to analyze all the files generated by the lower level and apply criteria with which to evaluate the quality model. The result is the values for quality properties, subcharacteristics and characteristics.
3. **Visualization environment**—This represents the top level of the environment. Its objective is to present the information obtained after the software product evaluation in an understandable manner.

   Besides showing the values of quality properties, subcharacteristics and characteristics, this environment also allows you to obtain a historical report for multiple versions of a product, compare different products and even generate predefined reports.

Thanks to this framework, the laboratory has achieved accreditation as a laboratory for quality software product evaluation as it relates to the ISO/IEC 25000 family of standards from Entidad Nacional de Acreditación, a member of the International Laboratory Accreditation Cooperation under ISO/IEC 17025.

Example of evaluation report results / FIGURE 4



Six steps to certify products

After organizations have evaluated and improved their software products, they can choose to certify these products. To do this, the certification entity (in this case, AENOR) has developed a six-step process for product certification and has defined the communication flow, shown in Figure 5 (p. 36).

- **Step one**—If interested organizations wish to certify the quality of a software product, the first step is to contact an accredited laboratory and request an evaluation report. At this point, the five evaluation activities will begin.
- **Step two**—As a result of product quality evaluation, the organization will obtain a quality evaluation report (shown in Figure 4). Based on the report's results, the organization may choose to become certified (if the level is good) or to refactor the product to meet established quality thresholds.
- **Step three**—When the organization has an evaluated product with an adequate level of quality, it can request certification from AENOR by providing its data and the product to be certified.
- **Steps four and five**—AENOR will contact the accredited laboratory (step four) to check whether the organization really has evaluated its software product and the level obtained is adequate for the certification. In this case, the laboratory will provide AENOR with the evaluation report (step five) for it to review and continue the certification process.
- **Step six**—Finally, AENOR will carry out the certification audit using information from the evaluation laboratory and information obtained after a visit to the developer. As a result, AENOR creates an audit report with the results and presents the certificate of product quality when applicable (see Figure 6, p. 36).

Several assessments and refactoring cycles were necessary during the pilot project. Three organizations eventually attained quality certifications for their software products: a document management system, a digital library for mobile platforms and supplier application

software for healthcare centers.

The chief information officers of the organizations that achieved certification noted the main benefits for their certified software:

- The amount of code developed was reduced 40%.
- The execution and load speed of applications increased.
- The corrective maintenance effort was reduced 90%.
- Product quality evaluations were incorporated into the software development life cycle.

This pilot project has gained widespread acceptance by the participating organizations—those that succeeded in certifying their products and those that did not—because they were able to detect the principal maintenance problems so they wouldn't repeat them in the future.

This pilot project also has shown that the software development sector can address certification as it relates not only to the quality of the processes, but also to the quality of the software product by using a scheme of international standards such as the ISO/IEC 25000 family.

Finally, after this first pilot project, the scope of laboratory evaluations is now being broadened to include new quality characteristics of ISO/IEC 25000, such as functionality, usability and performance. Furthermore, AENOR also will support the certification of the new quality characteristics, and it will work to align the software product certification with the software process certification. QP

REFERENCES

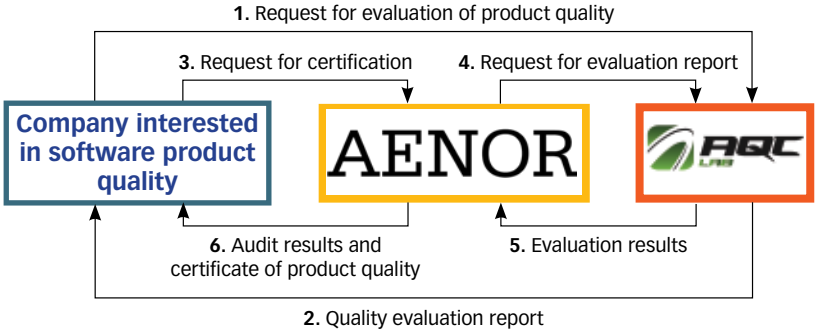
1. *The CHAOS Manifesto: Think Big, Act Small*, Standish Group, 2013.
2. Barbara Kitchenham and Shari Lawrence Pfleeger, "Software Quality: The Elusive Target," *IEEE Software*, Vol. 20, No. 1, 1996, pp. 12-21.
3. Tom Maibaum and Alan Wassyng, "A Product-Focused Approach to Software Certification," *Computer*, Vol. 41, No. 2, 2008, pp. 91-93.
4. International Organization of Standardization (ISO), *ISO/IEC 25040 Systems and software engineering—Systems and software quality requirements and evaluation (SQuaRE)—Evaluation process*, 2011.
5. ISO, ISO/IEC 25010, *SQuaRE—System and software quality models*, 2011.
6. Javier Verdugo, Moisés Rodríguez and Mario Piattini, "Using Agile Methods to Implement a Laboratory for Software Product Quality Evaluation," presentation at the 15th International Conference on Agile Software Development, Rome, May 26-30, 2014.
7. Diomidis Spinellis, *Code Quality: The Open Source Perspective,* first edition, Addison-Wesley Professional, 2006.

MOISÉS RODRÍGUEZ is CEO of the Alarcos Quality Center (AQC) and director of AQC Lab in Ciudad Real, Spain. He has a master's degree in computer science from the University of Castilla-La Mancha in Ciudad Real. Rodríguez is an Information Systems Audit and Control Association (ISACA)-certified information system auditor, Asociación Española de Normalización y Certificación (AENOR)-certified ISO/IEC 15504 chief auditor and Test management approach (Tmap) Next Certified.

MARIO PIATTINI is a full professor at the University of Castilla-La Mancha, where he leads the Alarcos Research Group. He is also founder and scientific director of ACQ in Ciudad Real. He has a doctorate in computer science from the Technical University of Madrid in Spain. Piattini is a certified information systems auditor (CISA), certified information security manager (CISM), certified in the governance of enterprise IT (CGEIT) and certified in risk and information systems control (CRISC)—all from ISACA. He is a senior ASQ member.

CARLOS MANUEL FERNÁNDEZ is manager of information and communications technology (ITC) and ITC auditor leader at AENOR. He has been an associate professor at Pontific University of Salamanca in Madrid since 1987. Fernández has a master's degree in computer science from the Technical University of Madrid. He is a CISA and CISM from ISACA and has been certified by the Information Technology Infrastructure Library Foundations.

Flow followed during the pilot software product certification / FIGURE 5



AENOR = Asociación Española de Normalización y Certificación, or Spanish Association for Standardization and Certification
AQC = Alarcos Quality Center

Example of software product certificate / FIGURE 6