Selecting Tools for Software Quality Management

LUIS E. MENDOZA, MARÍA A. PÉREZ, TERESITA ROJAS, ANNA GRIMÁN
LISI, Dpto. de Procesos y Sistemas,
Universidad Simón Bolívar

LUISA A. DE LUCA
Dpto. de Gerencia de Sistemas de Información,
Banco Central de Venezuela

The objective of this article is to propose a set of metrics to support the selection of tools for software quality management. The feature analysis case study evaluation method was used as a framework, selected by applying the DESMET method, specially developed to evaluate software engineering methods and tools. As a result of this research, a set of 16 features with 59 metrics has been formulated to help in the selection of tools that support the software quality management process.

The features proposed were applied to nine software tools selected from those available in the market. The result was a well-founded decision for selecting a tool that was best suited for the specific needs of the organization.

Key words: quality features, quality management, software engineering tools, software process quality, software product quality, strategic planning

INTRODUCTION

For software products, quality must be built in from the beginning; it is not something that can be added later. To obtain a quality software product, the software development process must also reach some quality level.

Some international evaluation norms and models for software quality are centered in product quality, while others are centered in process quality. In the first group, ISO/IEC 9126 (JTC 1/SC 7 1991) and the Dromey (1995) model can be included. In the second group, ISO 9000 (Vidal, Wan, and Han 1998), the Capability Maturity Model for Software (CMM) (Paulk et al. 1993), ISO/IEC 15540 (JTC 1/ SC 7 1997), and the IDEAL model (Gembra and Myers 1997) can be considered. There are tools to allow software quality management from different points of view, and they can help in some of the tasks and activities of the software development process. Some of these tools are based on international norms and models of the software quality evaluation.

Therefore, the objective of this article is to propose a set of features that support the selection of software quality management tools. The final result is a quality assurance plan that supports the selection process of one of these tools.

By using the proposed features, Venezuelan organizations now have an objective guideline to select a tool for supporting software quality management. In this way, they will be able to map out a quality assurance plan and make the necessary tasks tool-aided. Therefore, high-quality software could be developed more effectively in order to deliver competitive products to the market.

A subset of these features evaluates technical issues of the tools, while others are related to organization. The weight assigned to each feature will depend on its importance to the organization.

The application of these features does not require previous experience, but it does require a well-defined quality management process. The time required to apply these features will depend on knowledge related to the tool directly. It does not, however, imply the necessity of acquiring it.

This article provides a description of quality management and software quality tools. It then explains the method used in this research, followed by a description of evaluated tools, an explanation of features proposal and scoring, and, finally, the analysis of results, conclusions, and recommendations are discussed.

QUALITY MANAGEMENT AND SOFTWARE QUALITY MANAGEMENT TOOLS

Achieving a high level of product or service quality is the objective of most organizations. In this respect, software is the same as any manufactured product. The definition of software quality, however, includes several aspects that are unique to software. The most relevant is that quality must be built in; it is not something that can be added later (Humphrey 1997). To obtain a quality software product, the software development process must also be of quality (JTC 1/SC 7 1991).

Quality management is not just concerned with ensuring that software is developed without faults and conforms to its specifications (Sommerville 1996). A critical part of quality planning is selecting critical attributes and planning how these can be achieved. Software quality managers are responsible for three kinds of activities (Sommerville 1996):

1. **Quality assurance:** They must establish organizational procedures and standards that lead to high-quality software.
2. **Quality planning:** They must select appropriate procedures and standards and tailor them for a specific software project.
3. **Quality control:** They must ensure that procedures and standards are followed by the software development team.

There are tools to support software quality management from different points of view (planning and estimate, processes, documentation, and so on), and these tools can help in some of the tasks and activities of the software development process. Currently, few software development organizations have tools to support quality management, mainly due to lack of information about their availability. There are no guidelines to support software development organizations in their selection. Therefore, the objective of this research is to propose a set of features that support the selection of software quality management tools.

EVALUATION METHOD

DESMET is used to select methods for evaluating software engineering methods and tools (Kitchenham, Linkman, and Law 1996). DESMET is based on technical (evaluation context, nature of the expected impact of using the method or tool, nature of the object to be evaluated, scope of impact of the method or tool, maturity of the method or tool, learning curve associated with the method or tool, and measurement capability of the organization undertaking the evaluation) and practical (elapsed time that is needed for the different evaluation options, confidence that a user can have in the results of an evaluation, and cost of an evaluation) criteria in order to determine the most appropriate evaluation method in specific circumstances (Kitchenham, Linkman, and Law 1996).

The DESMET evaluation method separates evaluation exercises into two main types: *quantitative* evaluations aimed at establishing measurable effects of using a method or tool; and *qualitative* evaluations aimed at establishing method or tool appropriateness, that is, how well a method or tool fits the needs and culture of an organization. Some methods involve both a subjective and an objective element. DESMET calls these *hybrid* methods (Kitchenham, Linkman, and Law 1996).

In addition to the separation between quantitative, qualitative, and hybrid evaluations, there is another dimension to an evaluation: the way in which the evaluation is organized. DESMET has identified three ways to organize an evaluation exercise: (Kitchenham, Linkman, and Law 1996)

1. As a formal experiment where many subjects (that is, software engineers) are asked to perform a task (or variety of tasks) using the

methods or tools under investigation. Subjects are assigned to each method or tool such that results are unbiased and can be analyzed using standard statistical techniques.

2. As a case study where each method or tool under investigation is tried out on a real project using the standard project development procedures of the evaluating organization.
3. As a survey where staff or organizations that have used specific methods or tools on past projects are asked to provide information about the method or tool. Information from the method or tool users can be analyzed using standard statistical techniques.

In all, DESMET identified nine distinct evaluation methods, including three quantitative evaluation methods, four qualitative evaluation methods, and two hybrid methods:

- Quantitative experiment
- Quantitative case study
- Quantitative survey
- Quantitative screening
- Qualitative experiment
- Qualitative case study
- Qualitative survey
- Hybrid method 1: qualitative effects analysis
- Hybrid method 2: benchmarking

Technical Criteria Analysis

The evaluation context in this case is a set of tools involving the initial screening of many alternatives, with a detailed discussion of the tools. The initial exploring can then be based on feature analysis. Since the information systems management of Banco Central de Venezuela (BCV) expects to improve on the quality of the development process, the nature of the impact in this case is qualitative. The nature of the evaluation object is clearly a tool, meaning a specific approach within a generic paradigm.

BCV would like to select a software quality management tool. The scope dimension of the impact is the extent of it. It identifies how the effect of the tool is likely to be felt by the product life cycle. In this case, it is limited to all stages of software development, aiming to improve the quality software process. According to the maturity of the tools, they become very relevant to organizations that aim to improve their software products or process. The learning time aspect involves two issues: the time required knowing the tool, and the time required to become skilled in its use. The tools studied here are available in the market. Finally, the authors assume that the evaluation maturity of the organization is a qualitative evaluation capability, because they have well-defined standards for software development and the adherence to these standards can be monitored.

Practical Criteria Analysis

With respect to the timescale criterion, it has been ranked as "long" or "medium" in the authors' case, three to five months for an academic research. DESMET recommends case study (quantitative and feature analysis) and feature analysis survey for a long or medium timescale. According to the comments given for each method, the authors observe that feature analysis case study (FACS) is the favored method. For the second practical criterion, the authors have ranked the risk as "high," because an incorrect decision would be regarded as serious if large investment decisions were affected. When the risk is high, DESMET recommends the quantitative case study and FACS or screening model.

Both methods could be applied with a high-risk ranking. However, FACS seems to be better, since quantitative case study requires more than one project. Finally, the cost criterion will be considered. It has been ranked as a "medium" cost for this study, since a student makes the evaluation. DESMET recommends the following methods: quantitative case study, FACS, feature analysis survey, feature analysis screening mode, and benchmarking.

Based on the technical and practical criteria analysis, the evaluation method FACS was selected and applied into the Information Systems Management of BCV context. According to Kitchenham and Jones (1997), there are certain considerations when presenting and analyzing the results of an evaluation based on the FACS method. The analysis should be based on the differences among the values obtained for each evaluated tool when there is an explicit level of acceptance.

To use the FACS method, DESMET proposes several criteria: benefits difficult to quantify, benefits observable on a single project, stable development procedures, tool or method user population limited, and timescales for evaluation commensurate with the elapsed time of one's normal size projects.

There is a group of activities, specific for the evaluation, that should be performed. These are:

- To identify the tools to evaluate
- To identify a group of characteristic to evaluate
- To evaluate the tools against the identified characteristics
- To select a project pilot
- To test each tool in the project pilot
- To assign value to each characteristic for each tool
- To analyze the resulting values and carry out an evaluation report

The deliveries of these activities are described in the next sections.

EVALUATED TOOLS

Nine tools were selected from those available in the market (a description of these tools can be found in the appendix, which appears in the online version of this article). Each tool supports certain quality management tasks (estimation, project management, so on). This will enable one to determine the meeting percentage of functionality presented by each one vs. minimal requirements related to quality management. The features proposed, which are broad and generic for any tool that tends to support this discipline, will represent this. The comparison between the tools makes it possible to determine the scope of each tool with respect to minimal requirements established (that is, the weight assigned to every feature). In this way, the more complete tool will be the one that has more and better features.

The characteristics used to evaluate each tool are reflected in the set of features proposed to select tools that support software quality management, which constitutes the objective of this research.

FEATURES PROPOSAL

A set of 59 features was used to support the process of selecting a software quality management tool. These features are inspired by an extensive review of innovation and diffusion literature on software quality, quality management, quality assurance, quality planning, quality control, and technology management.

The set of proposed features has been classified in technological and organizational types. These features support the selection process of a quality management tool. The technological features refer to the tool directly, such as design, use, and its atmosphere. The organizational features are related to the use of this type of tool in organizations.

Based on Rojas et al. (2000) and De Luca et al. (2001), the features for selecting tools that support software quality management have been classified (as well as the technological and organizational ones) as internal or external features. Internal features are related to the evaluated item (tool or organization) issue. External features are related to the context issue.

Since features and metrics proposed are generic and represent the requirements for a quality management tool, some will not be applied depending on the particular scope of every tool. However, a whole application enables one to evaluate the meeting level of organizational requirements (that is, the weight assigned to every feature). It means that the best tool will be the one that shows the highest values for the features evaluated.

Metrics for Technological Features

The metrics for technological features, either internal or external, are classified based on 12 approaches: methodology, phases, functionality, reliability, maintainability, evaluation and certification models, structural forms, online help, platform, licenses, costs, and support. Figure 1 shows internal and external technological features.

Metrics for Organizational Features

The metrics for organizational features, either internal or external, are classified based on four approaches: project management, development of personal, institutional image, and interinstitutional relationship. Figure 2 shows the internal and external organizational features.

FEATURE SCORING

Each of these metrics, either for technological or organizational features, has a series of questions associated

FIGURE 1 Internal and external metrics for technological features

Category	Approach	Feature
Internal	Methodology	Supports to quality methodologies Supports to necessary methodologies that the organization requires Satisfaction rate
	Life-cycle phases	Number of life-cycle phases that supports Number of life-cycle phases that the organization requires Satisfaction rate
	Functionality	Adaptability Interoperability Security
	Reliability	Maturity Fault tolerance Recoverability
	Maintainability	Stability
	Evaluation and certification models	Promotes the use of evaluation and certification models of software quality Promotes the use of evaluation and certification models of software quality that the organization uses Satisfaction rate
	Structural forms	Quality planning Quality control Software quality evaluation Processes documentation Development processes quality analysis Software product quality analysis Measuring the product and the software development process quality Costs estimate Resources estimate Defect estimate Data analysis Data import/export Reports and graphics
	Online help	Presence of Help facility Facility of use Satisfaction rate
External	Platform	Hardware readiness to operate the tool Readiness of additional software to the tool Satisfaction rate
	Licenses	Server/user licenses Commercialization system
	Costs	Tool and transfer costs Training costs Maintenance costs Technical support costs Additional software costs Additional hardware costs
	Support	Technical support in the country Training in the country Training type Bring updated versions Manuals Base installed Satisfaction rate

FIGURE 2 Internal and external metrics for organizational features

Category	Approach	Feature
Internal	Project management	Acceptance Maintenance Standardization Quality plan
	Personal development	Training Learning Ability
External	Institutional image Interinstitutional relationship	Vision Impact

FIGURE 3 Classification of the answers according to the domain type

Domain	Value
Y/N	Represents presence (Y) or not (N) of the characteristic that is evaluated.
1 – 5	Integer between 1 and 5, corresponding to the life-cycle phases of the development of systems: planning, analysis, design, construction, and tests.
1 – n	n is a positive real number and it represents costs of the tools.
0 – 1	Possible values: 0, 0.5, and 1 that mean: 0: more than two versions in two years 0.5: one version in two years 1: two versions in two years
%	A positive integer that represents a percentage rate.
0 – 2	Possible values: 0, 1, and 2 that mean: 0: negative experiences 1: without experiences 2: positive experiences
1 – 2	Possible values: 1 y 2 that mean: 1: only licenses for server or only for user 2: licenses for the server and for the user

with it that help determine if the feature is present in the tool being evaluated. There are seven types of answers that can be obtained (see Figure 3):

To carry out any mathematical and logic operation on the answers to the questions associated with each feature, the values of the answers should be standardized using the domain types. In this sense, the answers were given on a scale from 1 to 5 (where 1 is the minimum and 5 is the maximum value).

The facilitator assigns a weight to each question. The weight corresponds to a real value between 1 and 5, where 1 represents less importance and 5 represents more importance to the evaluator organization. Once all answers are standardized in the 1 to 5 scale, each must be multiplied by its associated weight. In this way, the final value of the answer is obtained.

Now, each feature has an associated series of questions, and their answers have values in one domain. To assign a value to the metric, an algorithm should be applied to take into account the final value of the answers. Therefore:

- If at least half of the answers have values greater than or equal to 3, then the metric value is the average of the answers;
- Otherwise, the metric value is 1

Applying the same algorithm, the value for each approach can be calculated: 1) based on the obtained values of the metrics for each category (internal or external); 2) based on the obtained values of the approaches and even for each feature type (technological or organizational); and 3) based on the obtained values of the category.

The value of each approach is:

- If at least a half of the metrics have a value greater than or equal to 3 points, then the approach value is the metrics average;
- Otherwise, the approach value is 1

The value of each category is:

- If at least a half of the approaches have a value greater than or equal to 3 points, then the category value is the approaches average;
- Otherwise, the category value is 1

The value of each feature type is:

- If at least a half of the categories have a value greater than or equal to 3 points, then the metric type value is the categories' average;
- Otherwise, the metric type value is 1

Special Considerations for Tools Test

To carry out step 4 of the evaluation method, a case study of BCV is considered. Based on the concepts presented related to FACS evaluation method, it should be noted that, for evaluating the selected tools, the manager of information systems of BCV carried out the role of sponsor, and the authors of this article carried out the evaluator and advisor roles. The technological user was the head of one department within the Information Systems Management of BCV.

A questionnaire was developed containing the questions associated with each feature. Then, a repository was created for each tool to collect the questions, standardized answers associated to each feature, and the organization weight assigned to each question. This repository also contained the calculations according to the algorithm presented previously to obtain the value of the metrics for approach, category, and type. A second repository was used to collect the values of the metrics and their precedent hierarchies (approach, category, and type).

Construct Validation of Questionnaire

Content validity, which assesses the completeness and soundness of the measurement, was established through the careful selection of items that had been validated in prior studies. To further reduce the possibility of any nonrandom error, five academic experts from different universities and senior information systems executives were asked to review the questionnaire with respect to its validity, completeness, and readability.

Cronbach's alpha coefficient was calculated to assess measurement reliability. The analysis showed that the reliability of variables was significantly higher than the value of 0.86 suggested for the early stages of basic research (Hernández, Fernandez, and Baptista 1998). Principal component factor analysis was used to test this validity property. The result of Cronbach's alpha to the questionnaire was 0.95.

Data Collection

To obtain the answer to each question, an evaluation of each tool was made by using the product and analyzing its whole documentation, considering the characteristics and constraints of the case study as the pilot project. For the questions whose answers were not observable directly, a questionnaire was sent to both suppliers or dealers and clients or users to obtain more information.

ANALYSIS OF RESULTS

There are certain important considerations when presenting and analyzing the results of an evaluation method based on FACS (Kitchenham and Jones 1997). When there is an explicit level of acceptance, the analysis should be based on the differences among the values obtained for each tool evaluated.

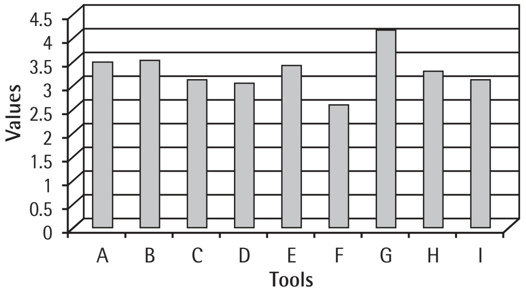
The final value of the evaluation was considered in order to make a decision regarding the tool to be acquired. A tool with a value smaller than 3 (explicit level of acceptance, according to the FACS evaluation method) is not advisable and needs a bigger analysis on the part of the evaluator organization. All the studied tools had values greater than 3 (see "total" row in Figure 4).

G was the one that obtained an evaluation with more value (4.4068), followed by B (4.0373), A (4.0088), and H (3.9795). The difference between the

FIGURE 4 Application results of the features, contained by type and category, for each one of the tools

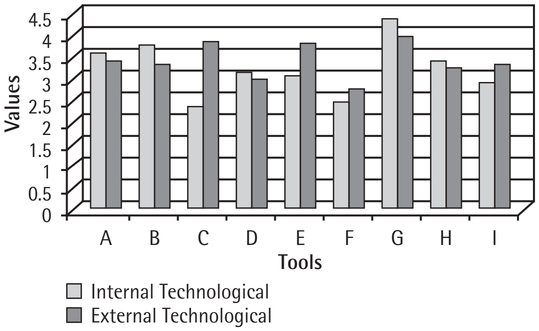
Type	Category	A	B	C	D	E	F	G	H	I
Total		4.0088	4.0373	3.8780	3.5610	3.8733	3.4395	4.4068	3.9795	3.8783
Technological		3.3926	3.4495	3.0060	2.9554	3.3299	2.5040	4.0635	3.2089	3.0066
	Internal	3.4936	3.6699	2.2500	3.0417	2.9455	2.3413	4.2580	3.2869	2.8109
	External	3.2917	3.2292	3.7619	2.8690	3.7143	2.6667	3.8690	3.1310	3.2024
Organizational		4.6250	4.6250	4.7500	4.1667	4.4167	4.3750	4.7500	4.7500	4.7500
	Internal	4.2500	4.2500	4.5000	3.3333	3.8333	3.7500	4.5000	4.5000	4.5000
	External	5.0000	5.0000	5.0000	5.0000	5.0000	5.0000	5.0000	5.0000	5.0000

FIGURE 5 Evaluation results of the tools according to the technological features



© 2002, ASQ

FIGURE 6 Comparison of internal and external technological features



© 2002, ASQ

first and the second is quite big (0.3695) with regard to the difference between the second and the third (0.0285), and the difference between the third and the fourth (0.0293). This suggests that for evaluator organizations, it would not be very complicated to select G as the best tool. Figure 4 shows the obtained results.

Since the value obtained in the organizational features oscillates between 4.166 and 4.750 points, an analysis of the results can be made based on the technological features. If some doubt is presented, the final decision can be based on the organizational metrics. The results of the technological features are shown in Figure 5.

It is important to note that only seven of the nine tools have obtained a value greater than 3: G (4.0635), B (3.4495), A (3.3926), E (3.3299), H (3.2089), C (3.006), and I (3.0066). The other two tools not mentioned in this group have their strength in aspects that the evaluator organization did not consider as high of a priority.

The higher tools still are the same, but the fourth became E. The difference between the first and the second is even bigger (0.614) with regard to the difference between the second and the third (0.0569) and the difference between the third and the fourth (0.0627). Again, it suggests that it would not be complicated to select G as the most appropriate tool.

On the other hand, the values can also be analyzed starting from the internal and external technological features. Only four tools obtained both values higher than 3 points: G (4.258 and 3.869, respectively), B (3.6699 and 3.2292, respectively), A (3.4936 and 3.2917, respectively), and H (3.2869 and 3.131, respectively). The differences among the values of the first two tools are again big (0.5881 and 0.6398, respectively), when the differences between the second and the third (0.1763 and 0.0625, respectively) and the third and the fourth (0.2067 and 0.1607, respectively) were compared. These results are shown in Figure 6.

The D, F, I, C, and E tools do not figure in the first places because of reasons similar to those outlined when the analysis of the tools evaluation results, according to the proposed technological and organizational features, was carried out.

Figure 6 shows the five tools whose values of internal and external technological features are lower than 3. Four (C, E, F, and I) show external feature values bigger than internal ones. This indicates that the final value of the technological features for these tools is being driven by the external technological metrics. This aspect has been considered lower than the external ones for the decision-making process.

The evaluation of these tools was carried out according to BCV needs. Because of this, the results can vary from one organization to another. The remaining five tools obtained good results in aspects that the evaluator company did not consider important according to their current needs. Although the tools D, F, I, C, and E were not inside the first four positions mentioned previously, it is important to highlight the following:

- The *processes documentation* feature shows that D is an excellent tool to manage the documentation activities.
- The *quality planning* and *costs estimate* features place F as a good tool to carry out the inherent activities to the quality planning process and to reduce the defects correction costs.

- The *development processes quality* analysis, *processes documentation*, *quality planning*, and *software quality evaluation* features demonstrate that I is a good tool to introduce quality to the software processes of conformity to ISO 9001 and 15504.
- The *quality control* and s*oftware quality evaluation* features place C as an excellent tool that serves from support to the planning phase of the measuring of programs based on the goal question metrics (GQM) paradigm.
- The *software product quality analysis, software quality evaluation,* and *measuring the product and the software development process quality* metrics show that E is a good tool to implement the metric derived from the GQM paradigm.

CONCLUSIONS AND RECOMMENDATIONS

To produce quality software products it is necessary to build quality in from the beginning. The tools that support software quality management have a great utility, since they make it possible to plan and track down the quality activities characteristic for each of the software development process phases.

To select a software quality management tool, it is necessary to keep in mind the technical aspects of the tool, as well as the organizational aspects of the company that is to adopt it, inasmuch as the selection is not only a product of the properties inherent to the tool itself, but also of the characteristics of the software development project and the characteristics of the organization that is going to adopt it. Therefore, the influence of the organizational environment in the selection of software quality management tools in supporting the software developing process must not be set aside.

The results of the case study led to the conclusion that the proposed features reflect the realities faced by software quality management tools. The features point to the strengths and weaknesses presented by software quality management tools according to organizational needs and the software development process.

As a result of this research, a set of 16 features with 59 metrics has been formulated and applied to nine commercial software products to guide the selection of tools that support the software quality management process. By using the proposed features, Venezuelan organizations will have an objective guideline to help them select a tool to support software quality management. They will be able to map out a quality assurance plan and make the necessary tasks tool-aided. Therefore, high-quality software could be developed more effectively in order to deliver world-competitive products to the market.

Some of these features evaluate technical issues of the tool, while others are related to the organization. Since particular requirements could be presented, the weight assigned to each feature will depend on its importance to the organization.

The application of these features does not require previous experience of the organization but a well-defined quality management process. The time required to apply these features will depend on knowledge related to the tool directly; however, it does not imply the necessity of acquiring it. It makes evaluation more cost effective.

The set of proposed features in this article guarantees an objective, repeatable, analyzable way to support the selection process of a software quality management tool and allows standardizing the requirement expected from these kinds of tools.

The case study outlined in this article allowed validating the proposed features. However, the authors suggest applying the evaluation to a set of tools in other organizations with external clients. This will provide a deep validation of the features using a different perspective.

ACKNOWLEDGMENTS

This work was financed by CONICIT S1-2000000437 and DID-CAI-USB S1-00094 projects.

REFERENCES

De Luca, L., L. Mendoza, M. Perez, and T. Rojas. 2001. Indicators for the selection of software quality management tools. In *Proceedings of XXVII Conferencia Latinoamericana de Informática-CLEI 2001, Mérida, Venezuela.*

Dromey, G. 1995. A model for software product quality. *IEEE Transactions on Software Engineering* 2: 146-162.

Gremba, J., and C. Myers. 1997. *The IDEAL^{SM} model: A practical guide for improvement.* Pittsburgh: Software Engineering Institute, Carnegie Mellon University. See URL http://www.sei.cmu.edu/ideal/ideal.bridge.html .

Hernández, R., C. Fernández, and P. Baptista. 1998. *Metodología de la Investigación.* New York: McGraw-Hill.

Humphrey, W. 1997. *Managing technical people: Innovation teamwork and the software process.* Reading, Mass.: Addison-Wesley.

JTC 1/SC 7. 1997. *SPICE. Software Process Assessment—Part 1: Concepts and Introductory Guide.* See URL http://www-sqi.cit.gu.edu.au/spice .

JTC 1/SC 7. 1991. ISO 9126- 1.2 *Information Technology—Software Product Quality.* See URL http://www-sqi.cit.gu.edu.au/iso9126 .

Kitchenham, B., S. Linkman, and D. Law. 1996. DESMET: A method for evaluating software engineering methods and tools. (Report ISSN 13 53-7776), Department of Computer Science, Keele University.

Kitchenham, B., and L. Jones. 1997. Evaluating software engineering methods and tools. Part 8: Analyzing a feature analysis evaluation. *Software Engineering* Notes 22: 10-12.

Paulk, M., C. Weber, S. Garcia, M. Chrissis, and M. Bush. 1993. *Key Practices of the Capability Maturity Model SM version. 1.1.* See URL http://www.ibp.com/pit/ispi/cmm.html .

Rojas, T., M. Pérez, A. Grimán, M. Ortega, and A. Díaz. 2000. Modelo de Desición para Soportar la Selección de Herramientas CASE. *Revista de la Facultad de Ingeniería*, UCV 15: 117-144.

Sommerville, I. 1996. *Software Engineering.* Reading, Mass.: Addison-Wesley.

Vidal, H., J. Wan, and X. Han. 1998. *Capability Models: ISO and CMM.* Kansas: Department of Computing and Information Sciences, Kansas State University. See URL http://www.cis.ksu.edu/~hankley/d841/old/cis841/Capability/

BIOGRAPHIES

Luis Eduardo Mendoza Morales is a professor at Simón Bolívar University. He has a master's degree in systems engineering from Simón Bolívar University. His research interests include the impact of information systems into organizations, systems integration, and information technology. His areas of expertise include information systems, software engineering, and methodologies. He can be reached at lmendoza@usb.ve .

María Angélica Pérez de Ovalles is titular professor at Simón Bolívar University. She has a doctorate in computer science from Central de Venezuela University; a master's degree in Information Systems from Católica Andrés Bello University, and a bachelor's degree in electrical engineering from Metropolitana University. Her research interests include process improvement, software engineering, methodologies, CASE tools, and information technology. Her areas of expertise are information systems, methodologies, and software engineering. She can be reached at movalles@usb.ve .

Teresita Rojas is associate professor at Simón Bolívar University. She has a master's degree in managerial engineering and a bachelor's degree in computer engineering from Simón Bolívar University. Her research interests include improvement of the development process of information systems. Her areas of expertise are information systems development and organizational aspects, case studies in Venezuela. Rojas can be reached at trojas@usb.ve .

Anna Cecilia Grimán Padua is a professor at Simón Bolívar University. She has a master's degree in systems engineering from Simón Bolívar University and a bachelor's degree in informatic engineering from Lisandro Alvarado University. Her research interests include knowledge management systems and software architecture quality evaluation. Her areas of expertise are information systems, methodologies, knowledge management, and software engineering. She can be reached at agriman@usb.ve .

Luisa Amelia De Luca Arocha is a systems analyst at Information Systems Management of the Banco Central de Venezuela (BCV). She has a master's degree in information systems and a bachelor's degree in computer engineering, both from Simón Bolívar University. Her research interests include software engineering and information technology. Her areas of expertise are information systems, relational database, and software engineering. She can be reached at ldeluca@cantv.net .

How helpful did you find this article? Please provide feedback at our online reader survey, which will be active approximately ten weeks after this issue's publication: www.asq.org/mr/sqp4_issue4.html .