Industry Sponsorship of Research in Software Engineering Measurement: A Canadian Initiative

ALAIN ABRAN

Department of Computer Sciences,
Université du Québec à Montréal, (Québec) Canada
Phone: (1-514) 987-3000 (ext. 8900); Fax: (1-514) 987-8477;
E-mail : abran.alain@uqam.ca
WWW http://saturne.info.uqam.ca/Labo_Recherche/lrgl.html

Australian Software Metrics Association Conference

Canberra, Australia, November 13, 1997

Industry Sponsorship of Research in Software Engineering Measurement: A Canadian Initiative

1. SOFTWARE METRICS

1.1 The Quantitative Approach in Management

An approach that employs quantitative measurements makes it easier to understand, evaluate, predict and monitor the development process and the software products developed. This need for measurement was pointed out by Basili (1995), who notes that the measurement process is an excellent conceptual mechanism for learning what works and what doesn't.

The quantitative approach is, in fact, a very important component of the management process in most departments of an organization: market analysis, selection, design and development of new products and services, production, delivery, billing, finance, even human resources. In all organizations, many of the quantitative components needed for decision making are found in the accounting systems, used not only for budgetary controls, but also for performance analyses and the taking of decisions that lead to all manner of changes within the organization. These accounting systems are, indeed, used in the overall process of production cost analysis; the same can be said of management engineering, where improvements to processes and products are made based on quantitative data collection systems.

1.2 Software Metrics?

But is this true of software management? What are the quantitative data collection systems for decision making? What quantitative models and measurements are available to Information Technology (IT) managers for analyzing productivity, assessing development and maintenance projects, and monitoring quality and productivity improvement programs? And how valid and reliable are the measurements and analytical models made available to IT managers? Finally, what approach should be used to analyze this validity and reliability?

In IT, in both North America and Europe, quantitative measurements and models fall under what is traditionally known as "software metrics". Now, for many authors, including Fenton (1991), the English term "software metrics" encompasses a whole series of concepts related to a host of activities having common quantitative elements:
- measurements and models for estimating cost and effort - productivity measurements and models - quality assurance and control - data collection - quality measurements and models - reliability models - performance evaluation - computational complexity - structure and complexity measurement systems.

This term has not, however, been standardized, and the definitions of measurements and metrics vary depending on the author. The standardization work currently being done by ISO sub-committees in the field of software engineering to clarify the concepts and vocabulary and the analytical methods, must therefore be seen in this context. We would cite, for example, the ISO SC7-WG6 sub-committee, whose mandate is to develop quantitative quality measures for software products (ISO- 9126), and the ISO SC7-WG12 sub-committee for the standardization of measures of the functional size of software (ISO 14143-1).

The members of these ISO committees have expertise, within the field of software engineering, in software measurement for the development and validation of the concepts, models, measures and measurement tools of software development and maintenance. The difficulties these sub-committees face are, however, considerable, and it will take years of work by international experts for consensus to be reached.

1.3 Purpose of the Presentation

Software measurement is currently an emerging field fuelled both by the best practices of industry leaders in software engineering and by university research. To combine the strengths of these two approaches, an industry-university alliance has led to the creation, at the Université du Québec à Montréal (UQAM), of a **software engineering management research laboratory.** This alliance is guiding research with a view to providing support to IT managers in their decision-making process.

The purpose of this paper is therefore to introduce the research laboratory, the research issues identified and funded by the industry, the research approach developed in the laboratory of UQAM, and the joint projects now being carried out.

2. THE SOFTWARE ENGINEERING MANAGEMENT RESEARCH LABORATORY

2.1 Mission and Objectives

To facilitate the performance evaluation process for investments in continuous improvement programs, our IT managers need analytical models and instruments of measure. Although the field of software measurement is experiencing growing success, with numerous scientific publications (Côté *et al.*, 1988; Dumke, 1995), several technology transfer projects (AMI 1992; Baumert et al., 1992) and industry successes (Grady, 1987; Grady et al., 1987; Daskalontanakis, 1992), this specialized area of software engineering is still young (SPRC, 1995).

The **mission** of the Software Engineering Management Research Laboratory is therefore to develop, for the IT industry, the analytical models and instruments of measure that will enable it to improve its decision-making processes in order to achieve its business objectives.

The aim of this Software Engineering Management Research Laboratory is to develop an international center of excellence in software measurement research. With a view to fulfilling its mission, the Laboratory has set itself several **objectives:**

1) To collaborate with the industry to develop new expertise in order to manage software more effectively;
2) To collaborate with the industry for the technological transfer of this expertise by adapting it to the various contexts of the industry;
3) To train, for the industry, specialized personnel capable of successfully installing these new management technologies in their IT departments;
4) To develop, in Montréal, an international center with R&D expertise in this specialized area of software management;

5) To contribute actively to the development of international standards for setting up performance measurement data repositories for inter-industry benchmarking, in both software development and maintenance.

2.2 Industry Partners

This laboratory is the outcome of an alliance between Bell Canada (Consult WWW http://www.bell.ca/) and the Université du Québec à Montréal (UQAM -Consult WWW http://www.uqam.ca/). Well aware of the immaturity of measurement processes in software engineering, Bell Canada, a common carrier that serves 60 percent of the Canadian market, granted UQAM significant research funding to create the Software Engineering Management Research Laboratory. This research funding includes, notably, several research fellowships for students enrolled at UQAM (at the master's, doctorate and post-doctorate levels).

Bell Canada is already closely involved in this niche of specialization in software management, and its Trillium model for the evaluation of IT suppliers (Coallier *et al.*, 1994) is on the cutting edge of industry research in this field.

The significant contributions of Bell Canada, in terms of funding, industry data and experienced practitioners, give us the leverage we need to accelerate the research projects undertaken several years ago with collaborators in Québec as well as in Germany and the United States (Abran *et al.*, 1993-94-95). The laboratory has also received substantial funding from the Natural Sciences and Engineering Research Council of Canada.

We would also point out that having access to a test field as large as that offered by a company of Bell Canada's size, is a major factor of the reliability of software engineering research results. The alliance illustrates the new opportunities of corporate-university partnership.

2.3 The Research Team

The computer science department at UQAM has nearly 40 full-time professors and a fair number of part-time lecturers. In addition to its undergraduate enrollment, nearly 250 students are enrolled in the graduate program, most of whom are professionals with an average of over 10 years' work experience.

Full-time staff includes an assistant director, Mr. Pierre Bourque, a software measurement specialist, and two support positions, including a web master. Nine industry researchers participate actively to many of the Laboratory's research projects. An associate professor and ten other professors round out the team.

The Laboratory also works actively with professors at the Université du Québec à Chicoutimi and researchers at the German universities of Magdeburg and Berlin, and is building links with university researchers in Australia.

3. MAJOR ISSUES FOR THE INDUSTRY

The use of systematic measurement in software development and maintenance is, however, far from widespread. This may be shown using the maturity model of the Software Engineering Institute (SEI); this maturity model of IT organizations has five levels, from level 1 (initial chaotic) to level 5 (optimized). We get an idea of the scant use of software development measurements from Baumert and McWhinney (1992), who note that 73 percent of IT groups are still at the chaotic level of the SEI model, and that a group can reach level 2 without having used any measurement. In all, this accounts for over 90 percent of the departments evaluated using this model of organizational maturity in software development!

Why is it difficult to use a quantitative approach in IT decision making?

In cooperation with our industry partners in research, we identified the following three issues in software engineering measurement, ranging from the most general to the most specific: the risks of measurement programs; maintenance measurement; the extension and instrumentation of measurements of functional size. Needless to say, these three issues do not cover the entire field of software measurement research, but they do represent the current concerns and priorities of our industry partners. We will now briefly present each one.

3.1 Risks of Measurement Programs

The first issue is the risks of the measurement programs themselves. The fact that the measurement process, though highly recommended, is not widespread illustrates that introducing measurement programs is not a simple matter. Moreover, Rubin (1991) reports that 80 percent of measurement programs are abandoned within 18 months.

We therefore propose to use the methods for adopting new technologies and success factors in technology transfer to study examples of the success, or failure, of software management measurement programs, and thereby gradually improve the standing of measurement programs in software engineering. In identifying and documenting the reasons for the success or failure of these programs, we hope to facilitate the establishment of measurement programs in various industry contexts.

3.2 Maintenance Measurement

The second issue is maintenance measurement. Many measurement programs have been developed and set up for the development process, while maintenance measurement programs remain the poor cousins (Sharpe *et al.*, 1991). Now, organizations report that software maintenance accounts for 50 to 80 percent of their IT budgets, and there are few productivity analysis measures and models specific to software maintenance.

The industry requires therefore a measurement approach tailored to the maintenance context. This line of research is aimed specifically at developing measures and models for corporate use for the in-house management of maintenance costs, the management of outsourcing contracts, and the development and monitoring of quality and productivity improvement programs in maintenance.

3.3 Extensions and Instrumentation of Functional Size Measurement Techniques

The third issue is the measurement of the functional size of software for use in productivity analysis and estimating. The *Function Points* measurement technique measures the functional size of software, and the results can be used in productivity analyses and the estimating process. The use of this technique is growing (Albrecht and Gaffney, 1983; IFPUG, 1994). Interest in it is evidenced by the fact that several annual conferences on this subject are organized by an international interest group, the International Function Point Users Group, with which a number of national groups in both Europe and Asia are affiliated.

Nevertheless, this measurement technique still far from meets all the needs of the industry, and our industry partners are particularly concerned about the problems related to its structure and use.

Among other things, they are looking for:
• extensions of Functions Points to software outside of the traditional MIS domain;
• improvement to productivity models based on the Functions Point Analysis technique;
• a better understanding of tools using Function Points;
• increase automation of the Function Points technique.

4. THE RESEARCH APPROACH

In order to address these issues identified by the industry, our Software Engineering Management Research Laboratory has explored and defined an approach for validating both measurements and models.

We will use the term "internal validation" to refer to the process of validating the measurements themselves, and "external validation" to refer to the process of validating models which combine several measurements with a view to identifying and analyzing the relations between the measured variables.

This approach has been integrated by the IEEE sub-committee working on the review of their standards on software metrics and by ISO-SC7 work groups on software measurement, and it has been quite favorably received by the international standardization community.

4.1 Internal Validation of Measurements

Internal validation, as defined and proposed to the ISO, is based on the classic concepts of measurement theory, measurement practice and measurement standards. A brief presentation of these three concepts follows:

A) Measurement Theory
- The measurement process must clearly define: a rule for recognizing an example of the object and its characteristic to be measured; a procedure for obtaining the measured value; a scale for interpreting the measured value.

- The measurement result must be plausible, that is, the measurement of an attribute must yield a value which corresponds to what individuals interpret as the measurement of this property.

B) Measurement Practice
The repeatability, independence and accuracy of the measurement process must be verified. In practice, then, it must be possible to ensure that the results of the measurement process do not depend on when or by whom the measurements are taken, and that the levels of accuracy of the measurement process are known.

C) Measurement Standards

A measurement must have a standard, and the measurement process must allow for the measurement results to be verified against a standard (such as the standards for the meter and the kilogram).

4.2 External Validation of Productivity and Estimation Models

By external validation, we mean the process of validating models which combine several quantitative parameters in order to analyze the relations between the measured variables. We are no longer referring to the measurement process, but rather to the use of the measurements. In our approach, we have classified the uses of measurements into three categories: analytical, predictive, and control.

A) Analytical
How does measuring a characteristic of an object add to the description of its relation to another characteristic of that object? Is it possible to construct a model where the measured value of a variable depends on the measured value of another variable? Example: a productivity model.

B) Predictive
Does the measurement make it possible to predict something? Is it possible to build a predictive model where the measured value of an attribute at a precise moment will depend on another attribute measured previously? Example: an estimation model.

C) Control
Are there control variables which enable us to alter the behavior of the production system? With what degree of accuracy, and in what circumstances? Example: a decision aid system.

5. EXAMPLE: THE VALIDATION OF ESTIMATION MODELS

To illustrate our laboratory's approach to tackling the issues identified by our industry partners, we feel it is important to present the key concepts necessary for a critical understanding and analysis of the current knowledge base of a specific component of software management, estimation models. To this end, we will summarize the state of knowledge and of practice in estimating, and go on to propose an analytical model of these models.

5.1 The State of Knowledge and of Practice

Elred (1992) asked the following question about the state of practice and of knowledge in computer projects estimating: Art, science, or voodoo? Let us take a look.

On analyzing the estimation process in IT, we find quite a difference between the present state of practice and of knowledge in software engineering, and what the estimation process should be, in relation to the other knowledge domains (whether administration, conventional engineering, science, or even social sciences). In these fields, the estimation process is based on practices (at times secular), definitions of measurements, data collection, data analysis, model building and, finally, estimating.

The IT industry quite often tends to jump directly to estimating, skipping nearly all the bases of the classic estimation process. In IT, the emphasis is on the end result, the final figure (for example, in monetary costs or human resources), and nearly all the basic principles common to the other administrative and scientific disciplines are ignored. Furthermore, computer analysts are very often optimists when it comes to selling their projects, and advocates or "storytellers" when it comes to explaining delays or cost overruns! This does not help the credibility of the estimation process.

On the academic side, many researchers have done comparative analyses of the results of *a posteriori* models to determine their reliability without, however, studying the predictive estimation process itself.

5.2 An Analytical Model

Let us now borrow analytical models from production management and management engineering that will enable us to position correctly and describe current knowledge in this area of software engineering.

To do so, we will resort to traditional production models and the everyday vocabulary of other knowledge domains, for example, productivity, productivity and estimation models, analytical measurements and models.

5.2.1 The Production Model

The traditional production model can be applied conceptually to software development regardless of the development paradigm (cascade, spiral, prototyping or iterative phase model - Boehms, 1988).

As for productivity, it is traditionally defined as the ratio, in the production process, of output to input. This concept therefore relies on two explicit quantitative dimensions (output and input), as well as on an implicit dimension, namely, the production process. This traditional concept of productivity also relies on quantitative measures available after the fact (once the production process is completed); the explanatory value of a productivity model is analyzed in relation to data available after the production cycles, hence, *a posteriori*.

Classic productivity models, however, rely on very specific output measurement concepts, namely, functional, rather than technical, measurements. For example, the output of an automobile production plant is expressed by the number of cars produced (by category), not by technical measurements of the amount of steel and rubber coming out of the plant! What about our IT models based on lines of code measurements?

5.2.2 The Accuracy of Measurements and the Production Model

Now, what basic conditions must be met in order to have good quality productivity models? We saw earlier that the right kind of measurements are needed to meet the criteria of classic productivity models. The measurement processes of specific aspects must also be of good quality, and the degree of accuracy of the measurements taken known and mastered (accuracy, variability, repeatability, controls). Also, we should not expect a model's level of quality to be higher than the level of quality and reliability of its measured parameters (hence, of the output and input measurements).

We must also know the conditions of stability, or instability-variability, of the implicit dimension, namely, the production process. It is therefore rash to expect a high degree of reliability (explicative) from a productivity model produced from uncontrolled or, should we say, out-of-control development processes!

Should we also expect that a single model will adequately explain all development processes, while other sciences have a model for every type of situation encountered within very well defined and experimentally demarcated limits?

5.2.3 The Classic Characteristics of Estimation Models

The classic notions of estimation models in the sciences of management and engineering are based on productivity models, with the addition of two main factors: the uncertainty surrounding the input variables of the estimation models; and the identification of project risks, not to mention, obviously, the estimation process itself.

In this context, the current vocabulary in IT, among academics and practitioners alike, is most certainly immature compared to the other knowledge domains, since what are commonly called, in IT, estimation models do not take into account the factors of content uncertainty and project risk.

5.2.4 The Quality of Estimation Models

At present, for IT development, there are almost no estimation models which meet the classic criteria of those found in other knowledge domains.

And when such models exist, it would be unrealistic to expect of them a higher degree of accuracy than that of their basic constituents (such as the measurements of the parameters themselves, and of the accuracy and reliability of the productivity models when all parameters are known and there is no longer any uncertainty or risk, the projects having been completed)!

To our knowledge, there are as yet no publications on the reliability of models which would meet the criteria of classic estimation models. You can well imagine why, since so few organizations, on the one hand, measure completed projects, measure them well and have constructed reliable explanatory productivity models, and on the other, are able to identify and quantify, at a project's outset, the uncertainty and risk factors, to say nothing, obviously, of their ability to keep credible, comparative project histories for the construction of classic estimation models!

6. PROJECTS BY RESEARCH THEMES

We currently have over 30 projects underway. Here are a few examples by industry research themes, with references to publications on the initial research results.

6.1 Risks of Measurement Programs

These projects are grouped into three sub-themes:

A) Measurement programs project risks.
This sub-theme includes projects to develop methodologies for identifying and quantifying the quality and the risks of measurement programs.

B) Software measurement programs as new technologies
This sub-theme includes projects to analyze the assimilation of software measurement programs from the viewpoint of the assimilation and deployment of new technologies.

C) Corporate-level Measurement Programs
This sub-theme includes projects to analyze the relevance of bringing various kinds of measurements into balance (satisfaction, productivity, quality, efficiency, capacity, etc.).

6.2 Maintenance Measurement

This second research theme includes projects on:

- Micro-Function Points for maintenance requests of very small functional enhancements (Abran and Maya, 1995);
- Adaptation of management engineering productivity models to software maintenance (Déry and Abran, 1995);
- Evaluation of software maintenance process (Zitouni, Abran and Bourque, 1995)
- Design of an econometric approach for the management of outsourcing contracts.

6.3 Extensions and Instrumentation of Functional Size Measurements

This third research theme of the industry includes projects such as :

- Analysis of the reliability of productivity models,
- Definition of a taxonomy for tools;
- FP tools market survey;
- Functional reuse measurement:
- Reliability of LOC-FP backfiring methods;
- Design of a semi-formal notation for FP;
- Validation protocol for testing automated counters;
- Re-engineering of FP from source code.

Initial research results have already been published , many of them available on our web site (Abran and Robillard, 1994; Abran, 1995; Abran and Paton 1995; Abran and Robillard, 1995 and Abran and Desharnais, 1995).

Also worth mentioning are the recent research results for the application of FP techniques to OO-Usecase Jacobson method as well as the recent design of an extension for real-time software (see our web site for the publications).

Last, but not least, current reuse measurement techniques do not adequately take into account functional size in the building of both productivity analysis models and estimation models. To address this issue, other projects on function points aimed at developing and testing reuse measurement methods not from a technical standpoint, but from the standpoint of software functions.

7. CONCLUSION

Software measurement has made considerable progress since its beginnings with the pioneering work of the seventies. But it still shows some signs of immaturity, notably with respect to practice and the analysis of software measurement results in both industry and academic circles. The Software Engineering Management Research Laboratory, the recent outcome of an alliance between the Université du Québec à Montréal and Bell Canada, is intended specifically to develop an international center of excellence in software engineering measurement. We believe this collaboration between industry and the academic community is one of the keys to success in this field, where testing in the industry environment is more than essential. We also would like to see this alliance lead to the establishment of other collaborative endeavors with new partners from industry and the academic community.

Acknowledgements

We would like to thank the organizers of the Australian Software Metrics Association Conference for their invitation to present this paper at ACSOM 97. We also wish to thank Bell Canada and the Natural Sciences and Engineering Research Council of Canada for research funding.

REFERENCES

- Abran, A. and Nguyen-Kim, H. 1993. "Measurement of the Maintenance Process from a Demand-based Perspective," in *Journal of Software Maintenance: Research and Practice, 5* 63-90.
- Abran, A., Desharnais, J-M., Meyerhoff, D., Mullerburg, M., St-Pierre, D. 1994. "Structured Hypertext for Using and Learning Function Point Analysis," *SEKE94 6th International Conference on Software Engineering and Knowledge Engineering*, Jurmala, Latvia, pp. 164-171.
- Abran, A. and Robillard, P.N. 1994. "Function Points: A Study of Their Measurement Processes and Scale Transformations," in *Journal of Systems and Software,* Vol. 25, May, pp. 171184.
- Abran, A. 1995. "Function Point-Based Production Models," in *Metrics in Software Evolution*, ed. Mulleburg & Abran, Oldenbourg Verlag, GMD Report no. 254, pp. 223-246.
- Abran, A. and Desharnais, J-M. 1995. "Measurement of Functional Reuse in Maintenance," in *Journal of Software Maintenance: Research and Practice*, 7 263-277.
- Abran, A. and Maya, M. 1995. "A Sizing Measure for Adaptive Maintenance Work Products," *Proc. International Conference on Software Maintenance,* Nice, France, pp. 286-294.
- Abran, A. and Paton, K. 1995. *A Formal Notation for the Rules of Function Point Analysis,* Rapport de recherche 247, Département de mathématiques et d'informatique, Université du Québec à Montréal, Montréal.
- Abran, A. and Robillard, P.N. 1995. "Empirical Validation of Function Points Measurement Processes," accepted for publication, in *IEEE Transactions on Software Engineering,* to be published.
- Albrecht, A.J. and Gaffney, J. 1983. "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," in *IEEE Transactions on Software Engineering,* SE-9(6) 639-647.
- AMI 1992. *A Quantitative Approach to Software Management*, Centre for Systems and Software Engineering, South Bank University, London. Consult WWW http:// www.sbu.ac.uk/~csse/ami.html.
- Basili, V. 1995. "Finding an Experimental Basis for Software Engineering," in *IEEE Software,* 12(3), pp. 92-93.
- Baumert, J.H. and McWhinney, M.S. 1992. *Software Measures and the Capability Maturity Model,* Technical Report CMU/SEI-92-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.
- Boehm, B.W. 1988. "Understanding and Controlling Software Costs," in *IEEE Trans. on Software Engineering* 14(10), Oct. pp. 1462-1477.
- Bourque, P. 1995. "Une alliance industrie/université en métrologie des logiciels à Montréal," *Actes des Huitièmes Journées Internationales: Le Génie Logiciel et ses Applications,* Paris, France, pp. 15-25.

- Coallier, F., McKenzie, R., Wilson, J.F., Hatz, J. 1994. *The Trillium Model,* Bell Canada, Longueuil, Canada. This document is also available by consulting WWW http://ricis.cl.uh.edu/trillium/trillium.html
- Côté, V., Bourque, P., Oligny, S. and Rivard, N. 1988. "Software Metrics: An Overview of Recent Results," in De Grace, P., Hulet Stahl, L. *The Olduvai Imperative - CASE and the State of Sofware Engineering Practice,* Yourdon Press Computing Series, Prentice-Hall, 1993. Reproduced in *Journal of Systems and Software,* 8(2) 121-131.
- Daskalontanakis, M.K. 1992. "A Practical View of Software Measurement and Implementation: Experience within Motorola," in *IEEE Transactions on Software Engineering,* SE18(1 1) 998-1010.
- Déry, D. and Abran, A. 1995. "Adapting the SIMAP Productivity Model to Software Maintenance," *Proc. of IEEE Canadian Conference on Electrical and Computer Engineering,* 1038-1041.
- Dumke, R. 1995. *Software Metrics - A subdivided bibliography,* Research Report IRB007/92, July 1995, Otto von Guericke University of Magdeburg, Magdeburg, Allemagne. Available by consulting WWW:http://irb.cs.uni-magdeburg.de/se/bibliography/bib main.html
- Elred, A.E. 1992. "Estimating: Art, Science, or Voodoo?," in *System Development,* June 1992, 1-4.
- Fenton, N. 1991. *Software Metrics - A Rigorous Approach,* Chapman & Hall, London.
- Grady, R.B. 1987. "Measuring and Managing Software Maintenance," in *IEEE Software 4(9)* 3545.
- Grady, R.B. and Caswell, D.L. 1987. *Software Metrics: Establishing a Company-Wide Program,* Prentice-Hall, Englewood Cliffs, NJ.
- IFPUG 1994. *Counting Practices Manual 4.0,* International Function Points Users Group, Westerville, OH, USA.
- Jeffery, R. and Berry, M. 1993. "A Framework for Evaluation and Prediction of Metrics Program Success," *Proc. IEEE First International Software Metrics Symposium,* 28-39.
- Kemerer, C. 1987. "An Empirical Validation of Software Cost Estimation Models," in *Communications of the ACM,* 30(5), May, pp. 416-429.
- Jeffery, R. and Berry, M. 1993. "A Framework for Evaluation and Prediction of Metrics Program Success," *Proc. IEEE First International Software Metrics Symposium,* 2839.
- Laframboise, L. 1995. *Méthode d'évaluation du risque lors de l'implantation d'un programme de mesures,* Centre d'intérêt sur les Métriques.
- Rubin, H.A. 1991. "Measurement in the 1990s: Shifting the Focus to Business Value," *Proc. International Function Point Users Group Spring Conference.*
- SPRC. 1995. *Software Mesurement Practices in Industry,* Software Practices Research Center, Jacksonville,Florida.
- Sharpe, S., Haworth, D.A. and Hale, D. 1991. :Characteristics of Empirical Software Maintenance Studies: 1980-1989," in *Journal of Software Maintenance Research and Practice, 3,* 1-15.
- Zitouni, M., Abran, A., Bourque, P. 1995. "Élaboration d'un outil d'évaluation et d'amélioration du processus de la maintenance des logiciels: une piste de recherche," *Actes des Huitièmes Journées Internationales: Le Génie Logiciel et ses Applications,* Paris, France.