December 2003

A Methodology for Self-Diagnosis for Software Quality Assurance in Small and Medium-Sized Industries in Latin America

Elizabeth Herrera
Universidad Autónoma de Bucaramanga

Raul Ramírez
Instituto Tecnológico y de Estudios Superiores de Monterrey

A Methodology for Self-Diagnosis for Software Quality Assurance in Small and Medium-Sized Industries in Latin America

Elizabeth Munive Herrera
Universidad Autónoma de Bucaramanga
lyz_unab@hotmail.com

Raúl A. Trejo Ramírez
Instituto Technológico y deEstudios
Superiores de Monterrey
raul.trejo@itesm.mx

Abstract

Currently, only a small fraction of software companies around the world achieve a high quality level for their development process. A vast majority of software producers, which have not yet implemented a methodology for software quality assurance, are paying high costs of production and systems maintenance, and are therefore being displaced from the global market, not being on the same competitiveness level than companies that posses a quality assurance method.

There are several models for software quality assurance, such as CMM, SPICE, MOSCA and the ISO 9000 norm. These models provide quality patterns that the company should adhere to, therefore improving its software production process.

Unfortunately, the successful implementation of such models is not generally possible within the context of small and medium-sized software organizations in Latin America, given deterring factors such as cultural impact and lack of resources, among others.

Our proposal for implementing and improving software quality assurance methods on such companies is a new methodology for self-diagnosis, based on concepts, goals and activities defined by CMM. Companies with minimal experience can use this methodology with quality standard methods.

We also define rules that allow the automation of the methodology through the use of an expert system, which would play the role of a "virtual quality auditor", supporting the definition of an action plan and thus reducing the investment required for processes improvement.

Keywords: Quality models, software quality assurance, expert systems, CMM, systems design and implementation in Latin America

Introduction

In a world of rapid changes and global competition, software development companies are pressed to achieve higher efficiency with lower costs. Medium and small software companies in Latin America (often referred to with the Spanish acronym "Pymes") are aware of their need for efficiency and quality if they desire to compete in the global market. Still, for many years, software production in Latin America was done more with an "artisan" approach fashion, as opposed to an "industrial" or engineering approach to developing. It is just in recent years that software engineering methods were considered and introduced by companies, the key goals to achieve being software quality and software quality assurance.

However, the proper implementation of software engineering techniques is often a difficult task for these small software organizations: in fact, it is even a difficult and complex task for big companies. For small organizations it is even more difficult, given that they often operate on limited resources and with strict time constraints, not to mention cultural issues like resistance to change from the employees or the management areas, who regard the extra work required for quality assurance as a useless and complicated burden put on the developing team. Hiring a consulting firm to start or improve a software quality assurance program is typically out of the question, due to budget constrains.

Still, the need for a good quality assurance program is becoming more evident, and managers are striving to achieve international quality standards that, in the long run, result in lower production costs.

In view of these particular needs for Latin American small businesses, in this paper we present a methodology for the auto diagnosis of the quality standards of a software company, which doesn't impose a great amount of resources, and which can help in creating and maintaining quality policies for the company. This methodology takes into consideration the special characteristics of Latin American organizations. The proposed methodology, based on CMM standards, provides a guide and evaluates the software development processes, providing a continuing improvement plan. In this paper we also show how this process can be automated with an expert system, thus creating a "virtual quality auditor" that would help in defining an improvement quality plan.

Is There a Software Crisis in Latin America?

The amount of resources invested in quality assurance for software systems is growing every day, since the investment in information systems is itself growing at a rapid rate (commonly, the software cost represents more than 50% of the total cost of the system). Software evaluation is based on quality assessments of productivity levels as much as in characteristics of the development method and the product itself.

Still, it is important to note that quality control represents a considerable amount of time and resources. It is widely recognized, however, that such investment is necessary and actually reduces costs in the long run, so organizations have created an open-minded culture towards software engineering techniques.

For Latin American companies, the growing expectations are the increased satisfaction of the client, a noticeable decrease in product defects, a reduction in maintenance costs due to a lack of good design, and a simplified project management. Still, budget constrained companies usually find the initial cost of implementing a good quality program challenging. Very often, at the end of software development projects within these small organizations, budgets are underestimated, planning is off by months, there is no control over the productivity of programmers and the product is hard to maintain.

As an example, the majority of software producing companies in Mexico are owned by small and micro entrepreneurs who are not interested in adopting formal methods for controlling their projects (more than 80% of Mexican companies fall into the small and micro enterprises category, according to Ania, 2003): their perspective is focused in meeting the requirements and appointed deadlines, without offering any added value. So, in general, these companies develop software products on a "case by case" basis, or based only on the experience of the current project developers. The authors have had the opportunity of observing this pattern of lack of stable quality programs in several companies in Mexico. The small fraction of Latin American companies that possess an official recognition from a quality institution of some kind (like CMM, ISO 9000) is also an indicator of this lack of quality standards. For example, only 10 Latin American organizations have been appraised by the SEI, three of them are major companies, like EDS, and only one is appraised as being Level 5, according to SEI's page (see SEI 2003).

Towards a Methodology for Quality Assurance: The Capability Maturity Model

In view of the observations presented in the previous section, we consider that a good starting point for a small business is to diagnose (in a way that is not considered expensive by the management) their current quality standards, obtaining guidelines for improvement as a result of such diagnosis. This diagnosis step is present in several well-known standards. The Capability Maturity Model by the Software Engineering Institute (SEI) is one of such standards, which we have chosen as the basis for our methodology, for reasons explained later in this work. Let us present a brief reminder of CMM.

CMM: A Brief Review

The Capability Maturity Model is a suit of practices or procedures that must be implemented by any business interested in developing software with quality and in improving the maturity of their processes (Bush, 1995). This model is applied to the software developing process and describes *five levels of maturity* that a business can achieve, in relation to its development processes, its process capability and its performance.

These five levels are:

- Initial
- Repeatable
- Defined
- Managed
- Optimized

A company that wishes to improve its processes through CMM must achieve the mentioned levels in a progressive, continuous and ascending manner, this is, the lower levels are the basis for the upper levels. One of the advantages of this model is that the official questionnaires are designed as a list of non-hierarchical, binary questions, so that figuring out the maturity level of the company is straightforward.

Each of the maturity levels contain a number of *key process areas* (KPAs), each of which include certain *goals*, which in turn encompass *commitments, activities and practices*, with the corresponding *abilities* to achieve them. We focus on the KPAs, given that they indicate the areas where software processes can be improved. Every KPA will have its associated set of activities and practices required for improving process capability. The five levels of CMM and its corresponding KPAs are summarized in Table 1.

In order to achieve a given maturity level, all key process areas within that level must be achieved, as well as those of lower levels. For appraisal purposes, failure to fully achieve a KPA means failure to complete the corresponding level.

Table 1. CMM Levels and Their KPAs

CMM Level	KPAs
1. Initial	• No KPAs
2. Repeatable	• Requirement Management (RM). • Software Projects Planning (SPP). • Tracking and Control of the Software Project. • Management of Software Subcontracts. • Software Quality Assurance. • Management of Software Configuration.
3. Defined	• Focus on the organization's process • Definition of the organization's process. • Training Program. • Integrated Management of Software. • Software Products Engineering. • Coordination among groups. • Peer Revisions.
4. Managed	• Quantitative Management of processes. • Software Quality Management.
5. Optimized	• Error prevention. • Management of Technological change. • Management of process change.

The Methodology for Self Diagnosis

The CMM model describes activities and practices required to achieve a maturity level within the software developing processes. Is precisely the use of these activities and practices what we consider valuable and feasible for small or micro software producing companies, so we start constructing our methodology for self-diagnosis on top of this model. Besides, the use of an internationally accepted standard serves as an extra incentive for adopting the methodology.

The main goal of our self diagnosis methodology is to gather facts that allow us to take a "snapshot" of the current processes implemented by the organization, so it is possible to understand them and then, to identify strengths and areas of improvement, thus determining the degree of completeness of every one of the KPAs of the CMM model. In order to gather this information related to the current processes of the organization, we have created three questionnaires, which we describe in the following subsections.

The Extended Maturity Questionnaire (EMQ)

Our first questionnaire is based on the *Maturity Questionnaire* developed by the SEI. This questionnaire poses questions about specific KPAs, allowing for four possible answers: *YES*, *NO*, *I DON'T KNOW*, AND *NOT APPLIED*. In the Maturity Questionnaire, all questions are related directly to goals and commitments for each KPA, but not to specific activities that should be performed to achieve such goals. For an example of the Maturity Questionnaire, see Bush, 1995.

CMM Level 2 Key Area: Requirement Management			
QUESTION	**YES**	**PARTIALLY**	**NO**
1. Do planning, final product and activities change in order to achieve consistency with requirement changes?	Planning, activities and final product are changed when requirements change.	Only the final product is changed when requirements change.	Neither planning, activities nor final product are changed when requirements change.
2. Do projects follow a written company policy in order to manage software system requirements?	There exists a written company policy for RM and is applied to all software projects.	There exists a written company policy for RM but is seldom or never applied to software projects.	There is no written company policy for RM.
3. Are all individuals in charge from the software engineering group trained for management and assignment of software requirements?	All individuals in charge are trained for management and assignment of software requirements	Only some individuals in charge are trained for management and assignment of software requirements	No individuals in charge are trained for management and assignment of software requirements
4. Are metrics defined and used in order to determine the status of the activities related to requirements management?	Metrics are defined and used in order to determine the status of the activities related to RM	Metrics are defined but not used	There are no metrics for RM
5. Do all activities related to management and requirements assignment are subject to revision by the SQA group?	All activities related to management and assignment of requirements are subject to revision by the SQA group	Only some activities related to management and assignment of requirements are subject to revision by the SQA group	No activities related to management and assignment of requirements are subject to revision by the SQA group

Figure 1. Extended Maturity Questionnaire

It has been our experience that the direct application of the Maturity Questionnaire in small and micro organizations results in a great majority of questions answered with a "*NO*", the main reason for this tendency being that many of the goals proposed in the questionnaire are only *partially* achieved. Therefore, we have modified the Maturity Questionnaire with the addition of an answer option, which allows for describing incomplete goals. Now we can clearly separate absent goals from incomplete ones. An example of this Extended Maturity Questionnaire is shown in Figure 1: the figure shows questions related to the first KPA of CMM's level 2. Questions related to CMM levels greater than 2 don't apply, and have been removed beforehand, thus the DOESN'T APPLY option has been eliminated.

The purpose of this extended questionnaire is to guide the system's administrator of a small organization so that he or she can answer the questions in a way that reflects more accurately the status of the organization. With this questionnaire the administrator is capable of identifying the goals and commitments that are being only partially fulfilled, so that they can be included in the improvement plan by the audit leader in charge of the self-diagnostic process.

The Goals, Activities and Responsibilities Matrix (GAR)

As it was mentioned before, the success of a model based on CMM depends on the complete achievement of certain goals and commitments for every KPA. These goals include measuring, documenting, reviewing and validation activities for software quality assurance. There is a close relationship among goals, activities and abilities, which is not that immediately apparent from the 344 pages description of the CMM standard (See Bush, 1995). It is obvious, then, that assimilating and interpreting this amount of information is a long and difficult process. In order to facilitate the task of the software administrators, we propose a matrix that makes explicit the relationship among abilities, activities, goals and commitments, as well as the responsible individuals for each activity. This matrix is designed under an OVAR pattern (objectives, variables, activities and responsible), where the objectives are the goals and commitments to achieve in each KPA, the variables are the abilities, the activities are the practices, sub-practices and activities associated to each KPA and, finally, the responsible person assigned to an activity is the person who plays the role proposed by CMM for each KPA.

As an example, consider the matrix for the Management of Requirements KPA. For clarity purposes, we divide the corresponding GAR matrix in two parts, shown in Figures 2 and 3. On Figure 2 we can see the first column containing questions regarding activities, commitments, abilities, measurement and verification, each identified by the codes shown on Table 2. The next column relates the question with the codes used in CMM. The next two columns relate each activity to the particular goals corresponding to this KPA, thus making easier to track which goals are being fulfilled. Goals related to questions answered with a "NO" will be considered as incomplete for audit purposes. The goals related to a particular KPA could be determined from the CMM documentation. For this example, Goal 1 is "To control software related system requirements in order to establish a baseline for software engineering and administrative uses", and Goal 2 is "To maintain consistency between plans, products and software activities and software system requirements".

Figure 3 shows the second part of the matrix, where we can see the relationship between the activities and the persons responsible of such activities, according to the roles specified by CMM. It is important to note that, for the majority of small organizations, it does not exist a specialist for each activity, so that it may be possible for a single person to be responsible of playing several roles. In this example, we identify the following roles involved with the fulfillment of the different activities: the client, the requirement analyst, the software engineering group, the manager, and the quality assurance group. The mark on the role column means this role is responsible of performing the activities described on the corresponding row.

Additionally, we included a column to identify the documentation required to perform the activity, or the documentation generated as evidence that the activity was performed.

When this matrix is used as a control within the life cycle of the project, it works not only as a diagnosis on the quality of the project, but as a "step-by-step" guide of action that helps the project manager to correct deficiencies and achieve the goals required by the KPA, thus providing an improvement in the overall process.

Goals, Activities and Responsibilities Matrix					
		GOALS		Answers	
QUESTIONS	**TYPE**	**Goal**	**Goal**	**YES**	**NO**
Level 2 Key Area: Requirement Management		1	2		
1.A.C.1 Does the project follows a written management policy to manage software system requirements?	C1	X	X		
1.H.1 For each project, are responsible individuals assigned for analyzing system requirements corresponding to hardware, software or any other system components?	H1	X			
1.H.1.1 ¿Are system requirements and their assignations managed and documented through the whole life cycle of the project?	H1	X			
1.H.1.2 Are there changes made to system requirements and their assignation?	H1	X			
1.H.2. Do related requirements include non-technical requirements (like agreements, conditions and/or contract agreements) that may affect or determine the activities related to the project?	H1	X			
1.H.2.2 The related requirements include technical requirements (like end-user, operator, support or integration functionality; performance requirements, design restrictions, programming language) that may affect or determine the activities related to the project?	H2	X			
1.A.1 Does the Software Engineering group revise the related requirements before their incorporation to the software project?	A1	X			
1.A.1.1 Are incomplete or omitted related requirements properly identified?	A1	X			
1.A.M.1 Are measurements defined to determine the state of the activities related to management of related requirements?	M1	X	X		
1.A.V.1 does the chief manager revise management activities for related requirements in a periodic basis?	V1	X	X		

Figure 2. The Goals, Activities, and Responsibilities Matrix, Part I

Table 2. Code Names for GAR Matrix

Description	CMM	GAR
Commitment Activity	C1	A.C.1
Measurement Activity	M1	A.M.1
Activity	A1	A.1
Verification Activity	V1	A.V.1
Abilities	H1	A.H.1

Comment: This relationship between activities and goals can be automated by means of an *expert system*, by transforming the relationships between rules, responsible and activities into inference rules that can be triggered by a knowledge base consisting on the answers given to the questions inside the matrix. An expert system simulates the behavior of a human expert; in this case we would have a "virtual quality auditor" which would help in defining an improvement plan based upon the areas of opportunity discovered by the GAR matrix, in a step-by-step fashion. The expert system could help a internal (or external) auditor in evaluating the maturity of the development processes, by determining the degree of satisfaction of each KPA according to CMM. In this way, by having the knowledge base and by filling the questionnaire, we could get a rating of the state of the organization, with no extra work, except for the conscious documentation of activities. Naturally, the capabilities of an expert system go beyond the implementation of the GAR matrix. The next step is to add decision-making abilities based on the knowledge of a human auditor. For techniques on how to extract knowledge from experts see, e.g., Giarratano, 2001.

The original CMM model is presented as a structured text. The GAR matrix provides a visual representation, which simplifies comprehension of the vast amount of information required for the evaluation of goals.

Goals, Activities and Responsibilities Matrix		**RESPONSIBLE**					
QUESTIONS	**TYPE**	**Client**	**Req. Analyst**	**Mgmt**	**Sw Eng. Group**	**SQA Group**	**Documents and Comments**
Level 2 Key Area: Requirement Management							
1.A.C.1 Does the project follows a written management policy to manage software system requirements?	**C1**	**X**	**X**	**X**			Doc: MR policy
1.H.1 For each project, are responsible persons assigned for analyzing the system requirements corresponding to hardware, software or any other system components?	**H1**		**X**	**X**			Doc: hardware software requirements analysis
1.H.1.1 ¿Are system requirements and their assignations managed and documented through the whole life cycle of the project?	**H1**	**X**	**X**		**X**		Doc: System requirements thru project life cycle
1.H.1.2 Are there changes made to system requirements and their assignation?	**H1**		**X**		**X**		
1.H.2. Do related requirements include non-technical requirements (like agreements, conditions and/or contract agreements) that may affect or determine the activities related to the project?	**H1**	**X**	**X**				Doc: Non technical requirements
1.H.2.2 The related requirements include technical requirements (like end-user, operator, support or integration functionality; performance requirements, design restrictions, programming language) that may affect or determine the activities related to the project?	**H2**	**X**					Doc: Technical requirements
1.A.1 Does the Software Engineering group revise the related requirements before their incorporation to the software project?	**A1**		**X**		**X**		Doc: Related requirements revision
1.A.1.1 Are incomplete or omitted related requirements properly identified?	**A1**		**X**		**X**		
1.A.M.1 Are measurements defined to determine the state of the activities related to management of related requirements?	**M1**			**X**	**X**	**X**	Doc: Defined measurements
1.A.V.1 does the chief manager revise management activities for related requirements in a periodic basis?	**V1**			**X**	**X**		Doc: revised measurements definition

Figure 3. The Goals, Activites, and Responsibilities Matrix, Part II

The Directed Questionnaire

The last format of our self-diagnosis methodology is a directed questionnaire with which a leader auditor can construct a knowledge base. This questionnaire has the essence of the original maturity questionnaire from CMM, but in this case each new question is generated based on the answer to previous questions. So a new question may be directed to complement information obtained earlier, or to confirm such information. In any case, useless questions are discarded.

The auditor leading the diagnostic uses his expertise in order to determine whether a given fact is achieved or not. For example, when the answer to the question "*Are the associated requisites managed and controlled?*" is affirmative, then the auditor continues with "*Could you describe how you receive, control and update requirements?*" In this way, the auditor can verify the facts presented. The manual realization of this task is a lengthy and slow process: it typically requires tow of the five days programmed by a certified auditor on CMM. Thus, this questionnaire should be included in the knowledge base of the virtual auditor described previously.

Evaluating the Results of the Self Diagnosis

The results obtained from the proposed questionnaires answer the basic question: Are the Key Process Areas required by CMM for a certain level achieved? For each KPA, there are four possible answers: The KPA is either fully achieved, partially achieved, not achieved, or it doesn't apply. A self diagnosis system could present these results in a visually oriented manner, by adding a color code to each answer: for example, a green color code would mean the KPA was achieved, a yellow code would mean the KPA was partially achieved, and a red code would mean a failed KPA. This coloring scheme is summarized in Table 3.

Table 3. Color Scheme for Diagnosis Results

Color	Result
Green	KPA Achieved
Yellow	KPA partially achieved
Red	KPA not achieved
No color	KPA doesn't apply

In this manner, the yellow and red colored KPAs are the areas of opportunity for improvement, and that should be part of an action plan. It is suggested that an external or internal consultant guide these activities.

The Methodology at a Glance

When the systems auditor applies the Extended Maturity Questionnaire, he or she obtains a first diagnosis of the quality processes of the organization, and generates a list of actions to perform, corresponding to the activities that where partially achieved or not achieved at all. The Goals, Activities and Responsibilities Matrix makes explicit the relationship between the activities and the goals that would be fulfilled by properly performing such activities, assigning a person responsible for such activities. The Directed Questionnaire allows the auditor to verify the affirmations given by the systems personnel, and to obtain additional information about the processes. The final result of applying these three questionnaires along the life cycle of the project provides a quantitative analysis, produces analytic views, shows strengths and weakness, identifies areas of improvement, provides recommendations and suggest an improvement plan.

Depending on the experience of the auditor and the overall state of the organization, not all questionnaires need to be used.

Suggestions for Using the Self-Diagnosis Questionnaires

- If it is desired to obtain a quick, global diagnosis of the current maturity of the development processes, and the auditor is knowledgeable in the concepts of CMM, we suggest to apply the Extended Maturity Questionnaire, which allows for a quick snapshot and provides an action plan, even if it is a general one. In this manner, the goals to achieve are identified, while the actual actions to be performed are left to the system managers.

- If it is desired to implement a quality assurance plan, we suggest to apply the Goals, Activities and Responsibilities matrix, which will support the quality assurance plan with a step by step guide about the strengths and weakness of the organization, clearly identifying the responsible individual for each activity.

- If the organization is considering obtaining an appraisal of its CMM level by the SEI, we suggest the use of the Directed Questionnaire, which will allow the auditor to obtain extra evidence about the activities currently performed by the organization, therefore evaluating if the organization is ready for the appraisal.

- The questionnaires can be used on a periodic basis to assess the improvement of the organization processes in a continuing improvement plan.

- It should be noted that the use of the three questionnaires could be used as a "pilot" appraisal before investing resources in obtaining the actual appraisal by the SEI. A small or micro organization can evaluate its processes in an inexpensive way before submitting itself to the actual certification.

Conclusions and Work in Progress

We have presented a methodology for self diagnosis based on CMM, which can be used by a small or micro organization as a part of the internal audit process. The methodology can generate a simple diagnosis, or help generate an action plan. The methodology can be used as a pre-appraisal towards an official appraisal by the SEI.

We are currently working out a collaboration agreement with a Mexican software company to test the questionnaires on several development teams, and perform an analysis of their effectiveness. A systems auditor will monitor the process and contrast results against those obtained by development teams using standard SEI tools.

The questionnaires presented here can be automated with an expert system, which would act as a "virtual auditor" guiding the organization in the improvement of their software development processes.

We are currently working on the creation of the knowledge base and the inference engine for an expert system that would guide the organization through CMM's levels 2 and 3.

Acknowledgments

This work is supported by the Department of Information Systems at Tecnológico de Monterrey. The authors are grateful to Hanna Octava and Agustín de la Maza, for their valuable comments and insight during the seminal stages of this work. We are also grateful to the anonymous referees for their comments and suggestions.

References

Ania, Ignacio, *The Software services Industry in Mexico*, AIS Report, March 2003.

Bush, Marilyn, *CMM, The Capability Maturity Model. Guidelines for Improving the Software Process,* Carnegie Mellon University, Software Engineering Institute, SEI Series in Software Engineering, Addison-Wesley, 1995.

Crosby, Phillip B., *La Calidad No Cuesta, El arte de cerciorarse de la Calidad*, Editorial Continental S.A. de C.V., España, 1994

Dunaway Donna K. and Masters, Steve, *CMM®-Based Appraisal for Internal Process Improvement* (CBA IPI), Version 1.2 Method Description, Technical report CMU/SEI-2001-TR-033 ESC-TR-2001-033, November 2001

Folgar, Oscar Francisco, *ISO9000 Aseguramiento de Calidad*, Ediciones Macchi, México, 1996

Giarratano, Joseph, *Sistemas Expertos: Principios y Programación*, International Thompson, Tercera Edición, 2001.

Munive, Elizabeth, *Una Metodología para el Autodiagnóstico del Aseguramiento de la Calidad del Software en las PYMES*, B.S. Thesis, Universidad Autónoma de Bucaramanga, Colombia, 2002.

Pressman, Roger, Ingeniería de software: un enfoque práctico, McGraw Hill, USA 1998

Software Engineering Institute, *Compiled list of organizations who have publicly announced their maturity levels after having an appraisal performed*, SEI website at **http://seir.sei.cmu.edu/pml/**, Carnegie Mellon University, 2002.

Trejo, Raúl and Munive, Elizabeth, *Una Metodología para el Autodiagnóstico del Aseguramiento de la Calidad del Software en las Pequeñas y Medianas Empresas*, Information Weekly México, to appear, 2003

Watts, S. Humphrey, *A Discipline for Software Engineering*, SEI Series in Software Engineering, Addison Wesley, 1995.

Watts, S. Humphrey, *Managing the Software Process*, Addison-Wesley Pub. Co., London, UK, 1998

Watts S. Humphrey, *Introduction to Team Software Process*, SEI Series in Software Engineering, Addison Wesley, 2000.