Methodological Proposal of Policies and Procedures for Quality Assurance in Information Systems for Software Development Companies Based on CMMI

Doris Cáliz[1*], Gustavo Samaniego[2], Richard Cáliz[2]

[1] Polytechnic University of Madrid, Department of Languages and Systems and Software Engineering (DLSIIS), Campus of Montegancedo 28660 Boadilla del Monte, Madrid, Spain.
[2] FIS Group, Department of Information Systems and Computer Science (DICC), National Polytechnic School, EPN. Ladrón de Guevara E11-25 y Andalucía. Quito, (Ecuador).

* Corresponding author. Email: doris.caliz@alumnos.upm.es
Manuscript submitted July 10, 2015; accepted December 8, 2015.
doi: 10.17706/jsw.11.3.230-241

Abstract: The aim of this work is to make a guide based on Capability Maturity Model Integration (CMMI) that is adapted to the reality of software development companies in Ecuador. The current work initially analyzes a conceptual reference framework with fundamental definitions from CMMI. Then, based on surveys, it presents a study of the current quality situation in software development companies, determining the priority given to the quality of the technological product delivered to the end customer. Subsequently, it proposes a set of policies and procedures based on CMMI for information systems quality control at software development companies. These proposals are present-ed clearly and concisely for each of the processes covered by the Engineering Area of CMMI. Finally, a validation of the applicability of the proposal for a medium-sized, nationally-representative software development company is presented. Additionally, the cost-benefit analysis of the proposal is included to better visualize the investments to be made and their potential benefits.

Key words: Quality software, CMMI software development, software quality policy, quality control.

1. Introduction

Quality assurance should be carried out in each phase of the software development process. For high quality software, records management documents should be quality products. For this reason, it is vital to clearly define the processes and responsibilities for each activity and their respective reviews [1]. However, the concept of quality software products has not been give this importance in Ecuador. This was determined through the responses of surveys given to the most important software companies in Ecuador. The main reason for the lack of quality software products is most likely the ignorance of the damage that poor quality products cause as opposed to the benefits of establishing processes for the development of organized policies for quality.

2. Background

Software quality can be defined as the set of properties that give software the ability to satisfy the explicit and implicit requirements of its user. The quality model ISO/IEC 9126 ISO/IEC 9126 defines the quality of a software product in terms of six main features: functionality, reliability, usability, efficiency, maintainability, and portability ISO IEC 9126-1 (2001). By combining these features, evaluation methods can be grouped as follows: inspection methods, methods of inquiry, and empirical methods [2]. Quality assurance methods based on software process improvement models have always been regarded by the software engineering community as one of the main sources of variability in software productivity [3]. This productivity may be positively influenced by disruptive software development methodologies (e.g., lean methods or automated development tools); however, there may also be impending costs [3]. Capability Maturity Model ® Integrations (CMMI) is a model of a maturity improvement process for product development and services. We can see in Figure1. It consists of best practices that address development activities and maintenance which covers the lifecycle of the product. The CMMI-DEV model provides guidance for applying CMMI best practices in a development organization [4].

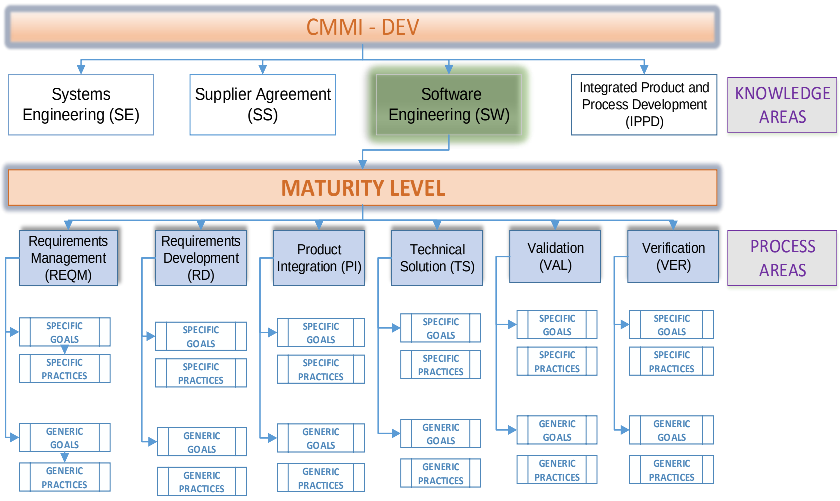

Fig. 1. CMMI scheme.

2.1. Comparison of Quality Management Models

An analysis of the different methodologies and models of quality management is performed: ISO 9000 covers a quality aspect that is applicable to anything, it is not limited just to software [2]. ISO 9001: 2008 is a set of social type and organizational rules to improve capabilities and performance. ISO/IEC 15504 with ISO 12207 applies a standard to the evaluation and improvement of the quality of both the development process and in software maintenance. Six Sigma is a process improvement methodology focused on reducing or eliminating the defects or failures in the delivery of a product or service to the customer. TSP is a set of practices for developing quality software products on time and on budget. PSP is used to improve discipline and competencies of an organization. CMMI is a model of evaluation of the processes of an organization. It is a model for the improvement and evaluation of processes of development and for the maintenance and operation of software systems [4]. After analyzing the different methods, it is clear that the model which best meets our needs is CMMI. The detailed analysis can be found in the full thesis of the authors in [5].

2.2. Analysis of Current Status of Software Development Companies

Two evaluation surveys were administered to determine the current state of software development companies with regards to quality control. The sample, *n*=4, is properly documented in the thesis. **Quality Management System Survey:** To understand the degree of quality control of the various processes in 9001: 2000. It has been noted that no part of the survey received more than 45% of positive answers. While it is done in some companies in a rudimentary way, overall there is basically no system management of representative quality. This can be seen in Table 1 and Fig. 2. Generic Quality Control Area Survey: To determine the size of the organization and the quality area. The analysis of the results found 13% positive answers, which indicates a low percentage of compliance with quality standards.

Table 1. Positives Results of the Quality Management System Survey

Area	Company A	Company B	Company C	Company D
Main channel Documentation Requirements	20	20	40	30
Management Responsibility	25	33	42	25
Resource Management	33	33	67	33
Product Realization	44	36	48	40
Measurement, Analysis And Improvement	12	12	29	29

For the enterprises surveyed, each had less than 1% of their staff dedicated to software quality control. The lack of dedicated personnel in this area highlights the low importance given to software quality assurance.

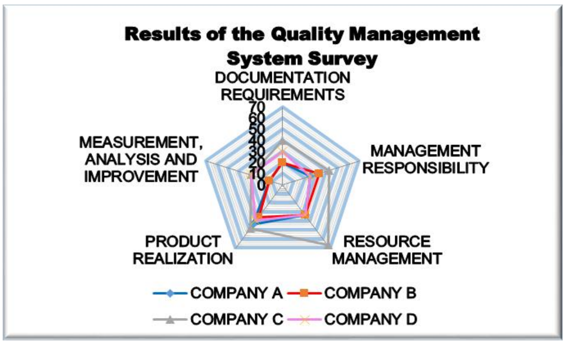

Fig. 2. Results of the quality management system survey.

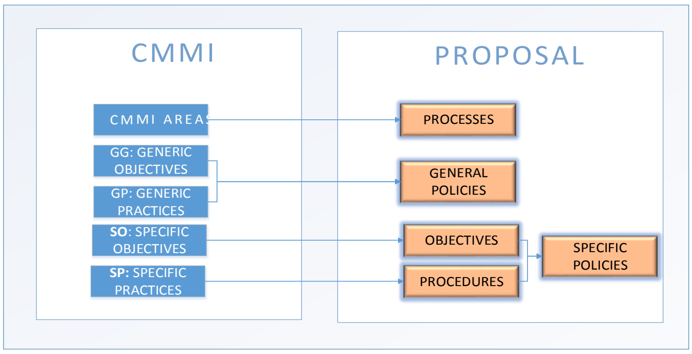

Fig. 3. CMMI vs the proposal.

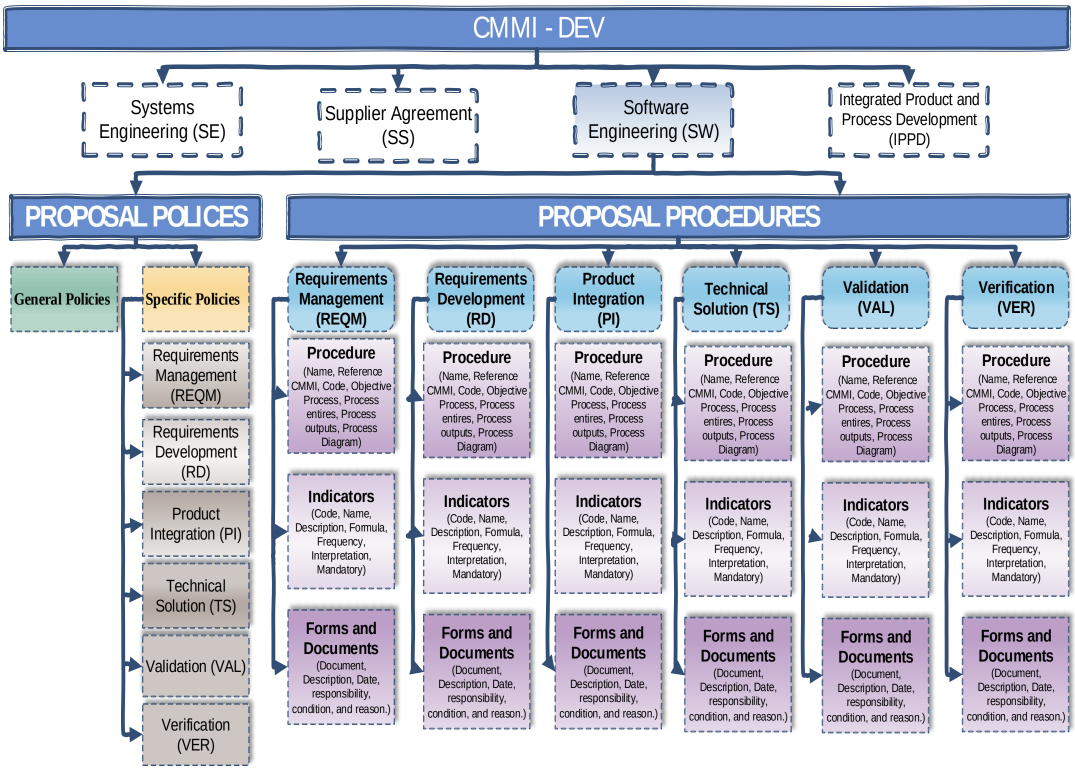

Fig. 4. Proposal.

3. Proposal

After studying the primary methodologies, this research proposes a practical methodology to make a guide based on CMMI that is adapted to the realities of software development companies in Ecuador. The innovative value of this project lies in its methodology, which will allow for quantitative identification of the degree of usability of mobile applications. This includes relevant aspects to be considered when this software is used by the elderly population.

The CMMI-DEV model has four areas of knowledge. Our proposal utilizes the area of software engineering. The proposal is shown in Fig. 4.

This knowledge area has six process areas. Each of these process areas includes: generic goals, specific goals, generic practices, and specific practices. Fig. 5 presents a mode. The proposal is limited to quality control processes in software development, all other process types are outside the scope.

3.1. Policies, There Are two Types of Policies: General and Specific

3.1.1 General policies

Generic practices are components that are common in all process areas. Table 2 describes the policies that will allow for the adoption of the proposed development companies in the country.

Table 2. General Policies

Code	Policy	Description	Responsible	Included
POG-001	**Planning Process Quality Control**	The plan will be documented with a clear description of the process including updating to reflect corrective actions or changes in requirements or objectives	Chief of Systems	– Monitoring activities and process control
POG-002	**Provide resources for the implementation of Quality Management Process (PGC)**	The policy of the institution is to promote the implementation of CMMI processes. The Chief of Systems will manage the activities required to deliver resources to the Financial Manager.	Chief of Systems (manager); Financial Manager (delivery)	– Adequate funding – Appropriate physical facilities – Qualified personnel – Fitting tools
POG-003	**Assign Responsibilities for PGC**	The Quality Control Coordinator will be designated as responsible for carrying out the process.	Quality Control Coordinator	– Detail responsibilities of the role
POG-004	**Train staff to PGC**	Train personnel who will perform or support the process, this process will be led by the Quality Control Coordinator and/or the Chief of Systems.	Quality Control Coordinator and/or Chief of Systems	– Self-study – Tutorial – External courses
POG-006	**Check the status with Management**	The Quality Control Coordinator, Chief of Systems, and Project Leader will review the policy and provide overall guidance of the process, its status, and the results with Management.	Quality Control Coordinator; Chief of Systems; Project Leader	– Activities to make decisions on planning and carrying out the process

POG-007	**Establish a defined process**	The Quality Control Coordinator and Project Leader will establish and maintain a description of the process that will be adapted to all standard processes from business.	Quality Control Coordinator; Project Leader	– Defining standard processes that cover the process area
POG-008	**Collect improvement information**	Quality Control Coordinator will collect the relevant information from work products, measures, measurement results, and information to support and improve future planning and implementation processes.	Quality Control Coordinator	– Store information in the organization's repository – Send the documentation for inclusion in the library
POG-009	**Establish quantitative targets for the process**	The Quality Control Coordinator will be responsible for establishing quantitative objectives for quality and process performance based on customer needs.	Quality Control Coordinator	– Establish quantitative objectives of the process
POG-010	**Ensure continuous improvement of PGC**	The Quality Control Coordinator will select and systematically publish process improvements and on technology.	Quality Control Coordinator	– Establish and maintain quantitative objectives for process improvement.
POG-011	**Correct the root causes of problems**	The Quality Control Coordinator and Chief of Systems will analyze the defects and problems encountered in the process.	Quality Control Coordinator; Chief of Systems	– Prepare a document with the solutions to be applied to errors
POG-012	**Constantly disseminate the results of the process**	The Quality Control Coordinator will provide monthly reports with indicators that show the evolution of the process and the results that are being obtained.	Quality Control Coordinator	– Circulation of indicators per project
POG—013	**Staff specialize in their roles**	The development company ensures the expertise of the people in their respective roles to promote training.	Chief of Systems	– Plan staff training

3.1.2 Specific policies

Table 3 describes the specific policies of the six areas in the engineering process category.

Table 3. Specific Policies

Area	Code	Policy	Responsible	Process
Requirements Management (REQM)	POGRE-001	**Policy to understand requirements.** All analysts should have a complete understanding of the project to be developed and warn or any need.	Analyst	GREPR001

	POGRE-002	**Policy to obtain commitment requirements.** The requirements to make the development must be agreed upon and signed between the parties.	Analyst	GREPR002
	POGRE-004	**Policy to analyze inconsistencies.** Any inconsistency between the requirements and the products will be documented and analyzed.	Project Leader	GREPR005
Requirements Development (RD)	PODRE-001	**Policy needs for identification.** Any request will be formally written with enough detail to continue the development stages.	Analyst	DREPR.002
	PODRE-002	**Formalize requirements.** The architect of the project will support any specific technical need that deserves to be understood.	Architect	DREPR.003
	PODRE-003	**Policy for Requirements Analysis.** Functional prototypes will be developed to validate the capture of requirements made by the equipment system analysis.	Team Project Leader	DREPR.009
Technical Solution (TS)	POSTE-001	**Policy for design test case.** The modules must be properly tested by the Programmer before being sent to the testing group.	Programmer	STEPR.008
	POSTE-002	**Policy for feasibility analysis to make, buy, or reuse.** The decision to buy, reuse, or develop in complex cases must be supported by a document containing the selection criteria and the responsibilities.	Project Leader	STEPR.009
	POSTE-003	**Policy for the selection of solutions.** It is the project architect's responsibility to correct the selection of technological alternatives and solutions.	Architect	STEPR.005
Product Integration (PI)	POIPO-001	**Policy to develop an integration plan.** Integrating products will always be scheduled by the Project Leader with special care of the components.	Project Leader	IPOPR.001
	POIPO-002	**Policy to prepare environments.** All environments required for integration will be managed by the Project Leader and prepared by the group configuration management company.	Project Leader	IPOPR.004
	POIPO-004	**Policy for product delivery.** The product delivery should be a formal process, including a record of delivery and a receipt signed by the relevant parties.	Project Leader	IPOPR.011
Verification (VER)	POVER-001	**Policy to select work products to check.** Work products will be selected based on their contribution to meet the objectives and requirements of the project and determine the risks.	Quality Control Coordinator	VERPR.001
	POVER-002	**Policy to establish a verification environment.** A tool for incident tracking solutions (Mantis Bug Tracker) is established. This tool will collect and process all incidents with metrics.	Quality Control Coordinator	VERPR.002
	POVER-003	**Policy for the corrections plan and settings.** Support and correction cases will not be addressed outside of the incident tracking tool.	Quality Control Coordinator	VERPR.011
Validation (VAL)	POVAL-001	**Policy for validation planning.** The plan will include validation tasks to be performed and should establish those responsible for fulfilling	Project Manager	VALPR.001

	each task.		
POVAL-002	**Policy to select products for validation.** Products and product components will be selected to be validated based on their relationship to the user's needs.	Project Manager; Quality Control Coordinator	VALPR.002
POVAL-003	**Policy to define acceptance criteria.** Performance metrics of products will be defined to determine if they meet or are within the allowed range to be certified.	Quality Control Coordinator	VALPR.003

3.2. Procedure Definitions

The procedures are performed based on each of the following process areas: requirements management (REQM), requirements development (RD), technical solution (TS), product integration (PI), verification (VER), and validation (VAL). This summary will only refer to the first area—the requirements management process. All process areas can be analyzed in greater detail in the thesis by the authors [5].

3.2.1 Requirements management

3.2.1.1 Procedures

Table 4 describes, in detail, the requirements management procedures.

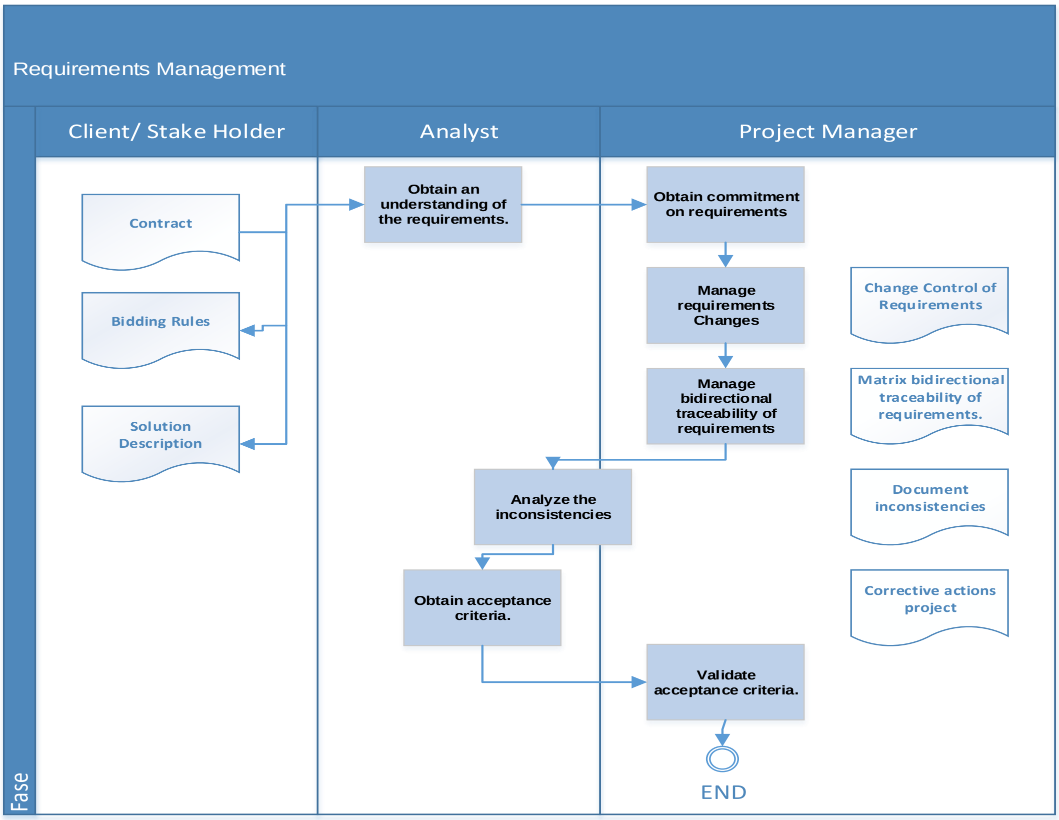

Table 4. Procedure for Requirements Management

Process Definition	
Process Name:	Requirements Management
Reference CMMI:	Requirements Management
Code	GRE

Objective Process: The purpose of Requirements Management (REQM) is to manage the project's product requirements and components, and identify inconsistencies between those requirements, plans, and project work products.

Process Entries	• Customer needs, requirements change needs
Process Outputs	• Document inconsistencies, corrective actions for project, matrix bidirectional traceability of requirements, change control requirements.
Process Diagram:	

Code	Name	Description
GREPR.001	**Obtain an understanding of the requirements.**	Within the development company, the project analyst works to carry out this procedure. The purpose is to develop an understanding of the meaning of the requirements with suppliers. The work products within this procedure are: lists of criteria to distinguish to the requirements providers, criteria for evaluation and acceptance of requirements, analysis of results against criteria, and an agreed upon set of requirements.
GREPR. 002	**Obtain a commitment on the requirements**	When integrated teams are created, the project participants are the integrated teams and their members. Typical work products are: impact evaluations of requirements, and documented commitments of the requirements and their change. Tasks to consider are: assess the impact of requirements on existing commitments, and negotiate and record the commitments.
GREPR. 003	**Manage changes to requirements**	Manage changes to requirements as they evolve during the project. Typical work products are: state of requirements, database requirements, and a database of requirement decisions. Tasks to consider are: document all requirements and changes to the requirements, and evaluate the impact of changes to the requirements.
GREPR. 004	**Manage the bidirectional traceability of requirements**	The intent of this specific practice is to maintain the bidirectional traceability of requirements for each level of product decomposition. Typical work products are: matrix of traceability of requirements, and system of tracking of requirements. Tasks in the procedure are: generate the matrix of traceability of requirements.
GREPR. 005	**Analyze the inconsistencies**	Detail the inconsistencies between the requirements, project plans, and work products and then initiate corrective action to resolve them. Typical work products are: documentation of inconsistencies (including sources, conditions, and reasons), and corrective actions. Tasks in this procedure include: review plans, identify the source of the inconsistency and reason, and start corrective actions.
GREPR. 006	**Obtain acceptance criteria.**	The analyst of the development company should be able to obtain the acceptance criteria. Evaluation criteria and acceptance should be: clearly and properly established, complete, consistent, uniquely identified,

		appropriate to implement, verifiable (can be tested), and traceable.
GREPR. 007	**Validate acceptance criteria.**	The company's development includes an important activity that should be completed by the stakeholder or customer to validate the criteria under which a request will be accepted or denied.

3.2.1.2 Indicators

Table 5 will describe requirements management indicators.

Table 5. Requirements Management Indicators

Code	**Indicator**	**Details**	
DREIN. 001	Percentage of changes to requirements	**Description**	Indicates a percentage of changed requirements on total accepted requirements
		Formula	$\frac{NRC}{NRT}*100\%$ NRC: number of requirements with changes, NRT: number of total requirements
		Frequency	Used as the metric for when the project ends, or monthly, to determine the punctuality of the project, considering all of the projects.
		Interpretation	It is desirable for a development company to have a value of 0%, which indicates that the requirements have been well-understood since the beginning.
		Mandatory	YES
DREIN. 002	Deviation in compliance with project plans	**Description**	Indicates a percentage of the planned value of progress in the construction of the requirements against the actual value.
		Formula	$\frac{TRP}{TPP}*100\%$ TRP = real time for the project, TPP = planned time for the project
		Frequency	Monthly
		Interpretation	It is desirable that this value is a low or negative rate which would indicate that the project is completed correctly and on- or ahead of time.
		Mandatory	YES
DREIN. 003	Corrective actions project	**Description**	Indicates a value to quantitatively know how many corrective actions were given
		Formula	N = number of remedial actions generated by the project.
		Frequency	Monthly
		Interpretation	It is desirable that this value is as low as possible. A larger value will reflect mismanagement at the requirements level. Comparatively, in several measures this value tends to decrease.
		Mandatory	YES
DREIN. 004	Number of inconsistencies found	**Description**	Indicates a quantitative value to quantitatively of how many inconsistencies related to the requirements exist.
		Formula	N = number of inconsistencies generated by the project

in the requirements	**Frequency**	Monthly
	Interpretation	It is desirable that this value is as low as possible. A larger value will reflect mismanagement at the requirements level. Comparatively, in several measures this value tends to decrease.
	Mandatory	YES

3.2.1.3 Forms and documents of management requirements

Table 6 describes the forms and documents of Management Requirements.

Table 6. Forms and Documents of Management Requirements

Document	Description
Document inconsistencies	This document should include a complete record for each inconsistency found that includes: • Date, responsibility, sources, condition, and reason.
Corrective actions project	This document provides a mechanism for monitoring and maintaining an inventory of corrective actions taken by the project. Each project should have a complete record that includes: • Date, project, corrective action, effect desired, effect achieved.
Matrix bidirectional traceability of requirements	Each project should have a matrix which, for each requirement, includes: • Responsibility for the requirement, who did the implementation, function module, testing the requirement, and user acceptance.
Change control of requirements	Change controls must be agreed upon and signed by the relevant parties and should include: • Who requested the change, acceptance, description of the change, affected modules, monitoring, and control.

4. Feasibility Analysis

A case study was performed on a software development company with about 700 employees in Ecuador. The feasibility of applying this proposal at the economic, technical, operational, and organizational levels is analyzed. Economic feasibility: An estimation (NPV, IRR, PRI) was made and the results show that the proposal is feasible from an economic standpoint. They also indicate a strong predominance of the benefits against the costs. Technical Feasibility: The results of the survey appreciate that 72% of the responses obtained were positive and therefore indicate feasibility in this area. Organizational Feasibility: The equivalence between the roles of the Research and Development Company case study and the roles that were raised in the proposal is detailed. Equivalence was positive. Operational Feasibility: A survey was administered to the Area Manager of the technology companies. The results of the survey were 98% positive responses, which indicates feasibility in this area.

5. Conclusions

The objectives of this project were covered in full and created a document that proposes a simplified guide for quality management of software development through a set of processes, policies, and practical procedures companies can implement. Following this guide can improve the quality of software products the companies provide. This approach supports goal-setting and prioritization in process improvements for the development of software products which directly improve the quality of products. The goal is to create a work environment where doing things correctly the first time is the objective and where quality is designed and integrated into each activity rather than inspected after products are made.

6. Recommendations

The realities of each software development company are different. We recommend using the principles of quality control when starting a project to implement a proposed quality management as presented in this thesis. In this way, due to economies of scale, it would generate an increase in economic, technical, and operational feasibility to obtain the results and expected benefits. Constant monitoring for policy compliance and appropriate use should be done and internal staff should be trained.

7. Future Work

The proposal was validated through a case study in one of the most important software development companies in Ecuador. The feasibility assessment was conducted at the economic-, technical-, operational-, and organizational-levels. The results were favorable to our proposal. Based on this research, future work may consider the real implementation in a software development company and analyze the benefits this methodology has provided since the start of use through the completion of the project.

References

[1] Ejaz, R., Nazmeen, M., & Zafar, M. A quality assurance model for analysis phase. *Proceedings of the 2010 Natl. Softw. Eng. Conf. - NSEC '10* (pp. 1–4).

[2] Lavallée, M., & Robillard, P. N. (2011). Do software process improvements lead to ISO 9126 architectural quality factor improvement. *Proceedings of the 8th Int. Work. Softw. Qual. WoSQ 11* (pp. 11–17).

[3] Duarte, C. H. C. (2014). On the relationship between quality assurance and productivity in software companies. *Proceedings of the 2nd Int. Work. Conduct. Empir. Stud. Ind. - CESI 2014* (pp. 31–38).

[4] Desarrollo, C. (2010). CMMI ® para desarrollo, versión 1.3. *C. Para Dsarrolo, Version 1.3*, *23*.

[5] Cñaliz, D., Terán, C., Samaniego, G., Cáliz, R. (2011). Desarrollo de una propuesta de polÍticas y procedimientos para el control de calidad de sistemas de información en empresas de desarrollo de Software basado en cmmi. Tesisf. D. E. I. D. E. Sistemas, Escuela Politécnica Nacional.

Doris Cruz Caliz Ramos received her master in management of information technology and communications from National Polytechnic School Ecuador from 2008 to 2012.

She is in a International Leadership Training, Germany, from 2011 to 2012. She is a PhD Student in Polytechnic School Madrid since 2013.

She is a academic visitor in Middlesex University London since 2015.

She is a professional in National Institute of Public Procurement in 2008-2009. And she is the head of the Quality Control Software in Cobiscorp Company from 2009 to 2011.

Cesar Gustavo Samaniego Burbano is a professor of the National Polytechnic School of Ecuador, a member of the Department of Information and Computer Science "DICC" School of Systems Engineering. Quito Ecuador.

He is a electronics and telecommunications engineer at National Polytechnic School Ecuador.

He received his master of computer science and informatics from National Polytechnic School Ecuador.

He is a learning expert in Processes National Polytechnic School Ecuador. His research interests are in las

emerging technologies, e-learning, TICs, management.

Richarth Harold Caliz Ramos received his master in management of information technology and communications from National Polytechnic School (EPN), Quito, Ecuador from 2008 to 2010.

He has worked in telecommunications and electronics engineering from National Polytechnic School (EPN), Quito, Ecuador from 1995 to 2002.

He worked as a analyst access network management in Andinatel S. A., Quito, Ecuador from 2003 to 2005. He is the head of management network access Dslam in Andinatel S. A., Quito, Ecuador from 2005 to 2008.