QUALITY MODEL FOR SOFTWARE DEVELOPMENT PYMES LOCATED IN THE VALLE DE ABURRÁ METROPOLITAN AREA

María Nelcy González Ramírez[1], Adriana Xiomara Reyes Gamboa[2], Julián Andrés Henao M.[3]

[1]Informatics Engineer, Professor in Tecnológico de Antioquia, Cll 78B No 72A-220, Medellín, Colombia. Email: nelcyga@hotmail.com
[2]M.Sc, Ph.D(c), Professor in Politécnico Colombiano Jaime Isaza Cadavid, carrera 48 No 7-153, Medellín, Colombia. Email: axreyes@elpoli.edu.co
[3]Informatics Engineer, Tata consultancy services, Carrera 52 # 14-30 Of. 331. Medellin- Colombia, Email: jhenao86@gmail.com

ABSTRACT

The purpose of this article is to showcase a research project (Quality Model for Software Development Pymes Located in the Aburrá Valley Metropolitan Area), and to present the resulting Quality Model, intended to support PYMES that develop software and are located in the Aburrá Valley Metropolitan Area. Many important models were analyzed for its development, such as MoProSoft, TST/PSP, Mosca, CMI, CMMI, from which the best practices were extracted. Additionally, important concepts were taken into account, such as Quality Models, Process Management, Development Processes, and Project Management. This quality model aims serve as a basis for small companies by supporting their software development teams, improving their performance and effectiveness in project elaboration.

Key Words: Software, Software Development Process, Quality Models, Software Quality

Received: April 15th, 2014. *Accepted: November 13th, 2014.*
Recibido: 15 de abril de 2014. Aceptado: noviembre 13 de 2014.

ABSTRACT

MODELO DE CALIDAD PARA PYMES DESARROLLADORAS DE SOFTWARE UBICADAS EN EL ÁREA METROPOLITANA DEL VALLE DE ABURRÁ

Mediante este artículo se desea mostrar el proyecto de investigación (Modelo de Calidad para PYMES Desarrolladoras de Software ubicadas en el Área Metropolitana del Valle de Aburra) se presenta un Modelo de Calidad que busca servir de apoyo a las Pymes desarrolladoras de software del Área Metropolitana del Valle de Aburra. Para su creación se realizó un análisis de modelos importantes como: MoProSoft, TST/PSP, Mosca, CMI, CMMI, de los cuales se toman como base las mejores prácticas. De igual manera se analiza conceptos importantes como Modelos de Calidad, Gestión de Procesos, Metodologías de Desarrollo, Gestión de proyectos. Este modelo de calidad busca servir de base a las pequeñas empresas, creando apoyo a los equipos desarrolladores de software mejorando su rendimiento y efectividad en la elaboración de los proyectos.

Palabras clave: *Software, Metodologías de desarrollo, modelos de calidad, calidad de software.*

1. INTRODUCTION

Software Engineering, alongside many different communication technologies, has become a very influential factor, not just for business, but for communication between people and companies.
Currently, software development PYMES (Small and Medium Companies), have become an important economic factor for the state of Antioquia. The software development market has quickly positioned itself as a part of our cultural development. The majority of the small companies have been based on development processes, but they have not adopted quality models because of the cost that this represents, thus creating the need to define a quality model for small companies based on the best practices of successful models for large companies, but with fewer costs and a due contextualization for our region.

The objective is to implement the right quality model, which can be adapted to the company's processes and needs, allowing them to deliver high quality software products to their current customers and fan out towards more potential customers.

It is necessary to structure all the elements that comprise the Quality Framework in a way that can be controlled, and that insures all the processes related to quality, by validating a prototype in an existing Pyme and taking samples of the results.

2. MATERIALS AND METHODS

Process Management in Software Development Companies

As for process management [1], every company that develops software has to remember that are in the field of software, and that the model that they choose to apply for development and maintenance processes has relevant implications in the business' general efficiency.

The people who are in charge of implementing more efficient methods may encounter issues if they become disoriented with the excessive supply of current quality models, process models and working techniques, or if they implement the first model they come across without an adequate preliminary investigation of the company's needs and requirements.

Project management

Project management [2] is in charge of all Integrated Project Direction techniques, of the need and importance of applying these techniques both for priority programs and projects, and for business development. A good Project management requires an initial investment of time and effort, the company's disposition to give them support in a disciplined manner, and the determination to break any disorganized working pattern to adhere to a new project methodology responsibly.

When you solve problems with a project-centered approach, a new work culture arises that differs from the traditional culture the company is used to, and a new figure emerges to lead in the solution of the denoted problem: the project manager. In other words, the project becomes an autonomous entity, governed by its own design and dependent on its own budget.

Project management guarantees that problems will be solved quickly, that resources will not be wasted and that they will avoid the chaos that results from working in areas outside the project's reach.
It also focuses on escaping future risks before problems arise. It strives to handle the expectations and communications between customers, collaborator, and interest groups in a more effective way. Finally, it seeks to create products of the highest quality from the start and to finish projects in time.

Software Development Processes

Development Processes [3] teach us that software development is not an easy feat. Many methodologies that influence different dimensions of the development process can be found nowadays. We have more traditional options that center on process control by rigorously establishing activities, artifacts, tools and notations that will be used. These options have proven to be effective and necessary for many projects, but they have also had problems in other projects.

One possible improvement would be to include more activities, artifacts and restrictions in the processes, based on the weak points they have detected. However, the final result would be an even more complex development process that could limit the team's ability to carry out the project. Another approximation is to focus on other elements, such as the human factor or the software product. This is the core of agile development processes, which give more importance to the people involved, to the collaboration with the customer and to the incrementing development of the software through short iterations. This approach has had been very effective in projects with changing requirements and when you need to cut back development time drastically while maintaining a high quality perspective.

Agile Development Processes have revolutionized the way that software is created, and have launched a wide debate between its adepts and its skeptics, because many don't consider them a real alternative to traditional Development Processes.

Software Quality

Software Quality [4] states that the quality of the software results from planned and systematic activities that make the (software) product reliable, and that it satisfies the quality requisites observed in Figure 1.

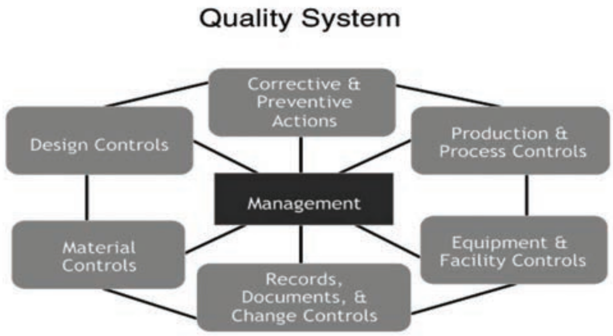

Quality System
Corrective & Preventive Actions
Design Controls
Production & Process Controls
Management
Material Controls
Equipment & Facility Controls
Records, Documents, & Change Controls

Fig.1. Software Quality System

The assurance of software quality lies in:

- Tools and methods for analysis, design, programming and testing
- Formal technical inspections through every step of the software development process
- Multiple-scale strategies
- Controlling software documentation and any changes that have been made
- Procedures that focus on adjusting to standards (and determining when the product is not adjusted to them)
- Mechanisms for metrics
- Registering audits and writing reports
- Activities that insure the software's quality
- Software metrics for project control
- Software verification and validation throughout its lifecycle
- Inclusion of tests and process revisions and inspections
- Software configuration management

Software Product Quality

Producing high quality software products [5] should be the main concern of the software development team, as you can observe in Figure 2.
However, a few points should be taken into account:

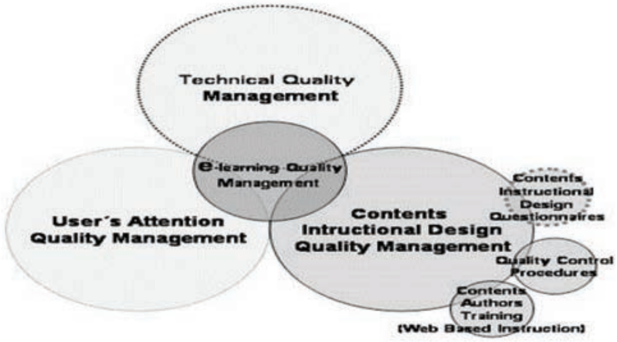

Technical Quality Management
e-learning Quality Management
User´s Attention Quality Management
Contents Intructional Design Quality Management
Contents Instructional Design Questionnaires
Quality Control Procedures
Contents Authors Training
(Web Based Instruction)

Fig. 2. Quality Regulations in the TIC (Information and Communication Technologies) Industry

- Software products are created by people, for people. As such, developers must take into account that people who make mistakes constantly are the ones who are going to be using their product. This is why software development assistants aren't reliable enough to replace a human hand in the development process. Software development is subject to many factors that can make it unreliable.
- Many people consider that quality is exclusive to the product, and that it can only begin to be considered when you begin to program. However, the concept of "Quality" includes many factors that come before this stage, and attention must be paid to them as well.

Accordingly, the quality that a software product can have is directly related to how each of the stages in the product lifecycle is carried out, starting from the definition of the idea all the way to the delivery and maintenance. Therefore, delivering a high quality product requires activities such as the following:

- Quality Management, making sure the differences between the designated budget and the budget that is actually used in the different stages.

These resources cover staffing, equipment and development times.

- The efficient use of Software Engineering, and development tools and processes.
- The use of formal techniques throughout the entire process.
- The reduction of variations between products by diminishing the differences and defects between versions.
- Rigorous testing in the different stages of the development process.
- Controlling both the support documentation and the documentation presented to the customer during every stage and verifying the possible changes.
- Providing the proper maintenance and post-sale services.

Development Processes

Different development processes [6] have many things in common. These elements have a common basis, but they are presented in different ways and under different names, which is why they seem to be referring to different things when, in reality, they only change the way they represent information and have different emphases on different elements, as can be observed in Figure 3.

However, we do encounter a change in paradigms, such as when agile development processes started to take off in the early 90's, as opposed to its sisters, the traditional "heavier" processes that prevailed.

This radical change in approaches was brought on by the endemic problems (recurrent problems) in software development:

- The requirements are never clearly defined before the project begins.
- Users only understand what they want after seeing an initial version of the product.
- Requirements change frequently during the project's development phase.
- The use of new tools and technologies (that have not been used before) make an a priori definition of the best work strategies very difficult.

This is the main difference between both development processes:

- Traditional processes highly emphasize the project planning phase, in having everything well specified before the start, following the plan precisely and documenting everything rigorously.
- Agile processes place their emphasis on delivering good coding to the customer, and on delivering results that satisfy them, by constantly adapting to their changing needs.

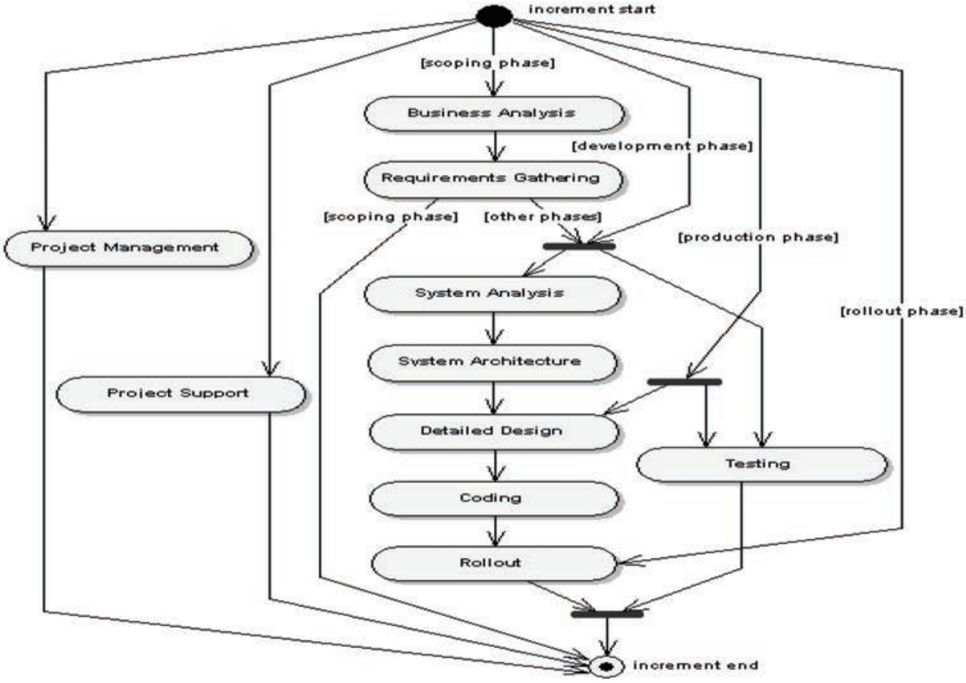


Fig. 3. Software development processes

In the words of the creator of agile methodologies: (extracted from their manifesto "Agile Development"):

- People and interactions vs. processes and tools.
- Software that works vs. exhaustive documentation.
- Customer help vs. contract negotiation.
- Response to changes vs. following the plan.

However, this doesn't mean that agile methodologies completely abandon established tools and procedures, or that they abandon discipline. It only means that they use them in a more flexible way, as tools that bring "a little order to the chaos they face". The main point is to obtain good results, not in doing things perfectly.

Agile development processes:

Agile development processes [7] are adaptive, not predictable. On the other hand, traditional development processes enable the planning of the entire long-term development process in a detailed manner. However, when something changes, everything the team planned can crumble; meanwhile, agile development create processes that adapt to circumstances and evolve with changes, even changing the team because agile development is more human-centered than process-centered. They work with the nature of the people assigned to the project instead of against it, allowing software development tasks to become fulfilling and interesting. Some of these agile development processes have been successfully used, but they lacked diffusion and recognition.

Process Revision

Every process has [8] special characteristics that highlight certain specific elements:

Scrum: A project management framework that has been used successfully for the past 10 years. It is especially appropriate for projects with fast-changing requirements. Its main features are: 30-day sprints of iterative software development; Meetings throughout the entire project, including a daily 15 minute meeting with the development team for coordination and integration.

Crystal Methodologies: A set of development processes characterized by being based on the people who make up the team (the success of the project depends on them) and on the reduction of produced artifacts.

Dynamic Systems Development Method (DSDM): Defines the framework for the development of a software production cycle.

Adaptive Software Development (ASD): Its main characteristics are: iterative, software-component-oriented (not task-oriented), and tolerant to changes. The lifecycle has 3 main phases: speculation, collaboration, and learning. During the first phase, the project begins and the software features are planned; during the second phase, the features are developed, and during the third phase, the quality is revised and it is delivered to the customer.

Feature-Driven Development (FDD): An iterative, 5-step process. The iterations are short (up to 2 weeks long). Its focus is on the design and implementation of the system based on a list of features that the software must have.

Lean Development (LD): Defined by Bob Charette's experience with the Japanese automotive industry during the 80's and used by many telecommunication projects in Europe.

Comparison between Agile and Traditional Processes

In Chart 1, the main differences between agile and traditional processes (regrettably called not-agile or heavy) can be observed [9].

The Software Industry in Colombia

The Colombian software industry [10] has acknowledged that, although high-quality software was needed, they never made an effort to insure that our industry generates products that allow us to compete in international markets. In the past decade, the Colombian Software Industry has improved its position internationally. "Despite these efforts, this area barely contributes between 1.5% and 2% of the Gross domestic Product. In 2009, the industry grew 7.7% in comparison to Latin America's growth of 8.9%, and though these figures can be improved, it is necessary for companies to invest in the TIC (Information and Communication Technologies) area. Using the software industry diagnosis carried out by Vive Digital, Colombia could develop its IT and BPO&O (Business Process Outsourcing & Offshoring) sectors by committing to a long-term program that allows significant barriers to be removed in different sectors. On a global scale, the TI Industry is about US$900,000, and grows at an approximate yearly rate of 7%.

Chart 1, Differences between agile and traditional Development Processes.

Agile Processes	Traditional Processes
Few artifacts. Modeling is expendable, models are disposable.	More artifacts. Modeling is essential, models are preserved.
Fewer, more generic, flexible roles.	More, highly specific roles.
There isn't a traditional contract, it's very flexible.	There is a fixed contract.
The customer is part of the development team, *in situ*.	The customer interacts with the development team during meetings.
Mostly used for smaller projects. Short (or with frequent deliveries), small teams (< 10 members) working in the same place.	Used in projects of any size, but most effective for bigger projects with a more disperse team.
Architecture is defined and improved throughout the entire project.	It suggests that the architecture be defined early in the project.
Emphasizes human aspects: the individuals and teamwork.	Emphasizes process definition: roles, activities, artifacts.
Based on code-production heuristics.	Based on development environment standards.
Changes are expected during the process.	No high-impact changes are expected during the project.
Agile development process	Traditional development process
Few artifacts. Modeling is expendable, models are disposable.	More artifacts. Modeling is essential, models are preserved.

In order of importance, these are the sectors where the Colombian market has found niches: telecommunications, financial areas, government institutions, manufacturing industry and finally, PYMEs, which have shown great dynamism in the increase of software-oriented expenses.

Correspondingly, you can observe an increase in companies that are certified in the CMMI International Standard, supported in this process by SENA, COLCIENCIAS and Proexport, among others.

Likewise, reports that in April 2011, there were a total of 37 companies certified in the IT Mark Model. To strengthen the CMMI Certification Program, on April 18th 2011, FEDESOFT and SENA had an open call for their TSP/PSP Certification Program for development companies that were part of the FEDESOFT Federation. Actions like these contribute to promote the use of high-quality Software development processes.

The Colombian case

In Colombia [11], ParqueSoft is one of the main Latin American information technology service providers. It has worked with around 300 companies, which means more than 1000 specialized professionals in the information industry. Its greatest potential lies in research, and software development and business innovation.

Another successful Colombian case in the software sector is PSL Software Producer, which received the IEEE/SEI Software Procesos Achievement Award in May 2006. As such, there are other companies that stand out because of the quality of their services, such as Ofimática and Compuredes, both located in Medellín.

However, things could improve significantly for the Colombian Software Industry, which remains imperceptible in National and International Contexts.

- While Oil has a participation 26% greater than Software, and carbon has a participation 12% greater than Software. According to a report by the Spanish Embassy's Economic and Commercial Office, the 188 software companies in Colombia can be allotted into one of these 4 categories:

a) Software Developers
b) Information Product distributors and retailers
c) Internet service and Access providers
d) Hardware producers

3. RESULTS

Quality Models

Seven quality models currently used in other countries were analyzed.

CMI: Maturity Model for software development capacity, as you can observe in Figure 4.
The model [12] is phased according to the concept of maturity, which defines 5 levels or steps to qualify an organization’s maturity.

Level 1: Initial
Companies that don’t base their development on defined processes and don’t apply project management techniques. It refers to chaotic environments with heroic programming, whose results are unpredictable and depend exclusively on the people.

Level 2: Repeatable
Defines organizations that apply project management techniques, even if they don’t have defined processes.

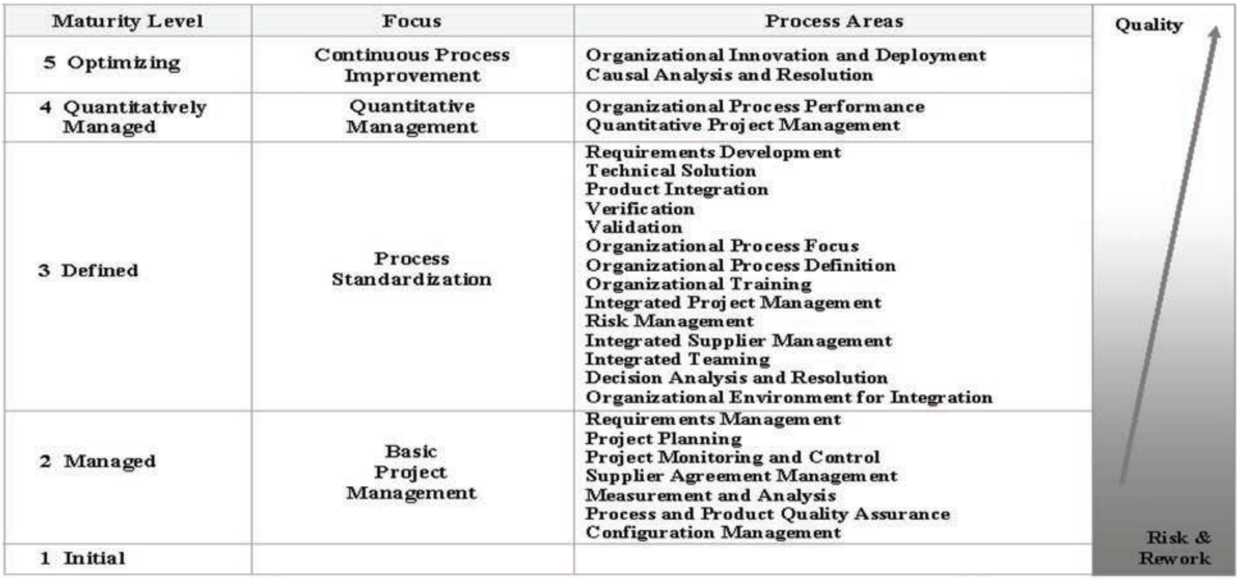
Level 3: Defined
Organizations that have precise, defined processes they execute regularly. Companies at this level examine the experience they have in the projects they carry out and apply the lessons they have learned to improve their processes.

Level 4: Arranged
In the 4th level of maturity, you have companies that have perfected their analysis of the projects they have carried out, and have institutionalized them as processes that measure their development ability quantitatively, so that they can predict results and evaluate improvements in an objective, measurable way.

Level 5: Optimized
Organizations in Level 5 have continual improvement processes that have been thoroughly defined and applied, and these processes are continually nurtured with the information the organization receives in the processes described in Level 4.

This model is both a guide to improve the institution and a set of criteria to evaluate its current level. While CMI centers these two dimensions of maturity on organization, CMMI introduces another dimension: process capacity, which can be used to guide improvement activities and to evaluate organizations.

CMMI: CMMI (Capability Maturity Model Integration) Product Suite [13] seeks to continually improve the process and product quality of an organization, and it leads the organization towards this goal by establishing maturity levels: Initial, Arranged, Defined, Quantitatively Arranged & Optimized.



Maturity Level	Focus	Process Areas
5 Optimizing	Continuous Process Improvement	Organizational Innovation and Deployment Causal Analysis and Resolution
4 Quantitatively Managed	Quantitative Management	Organizational Process Performance Quantitative Project Management
3 Defined	Process Standardization	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Integrated Supplier Management Integrated Teaming Decision Analysis and Resolution Organizational Environment for Integration
2 Managed	Basic Project Management	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
1 Initial		

Fig. 4. CMI System

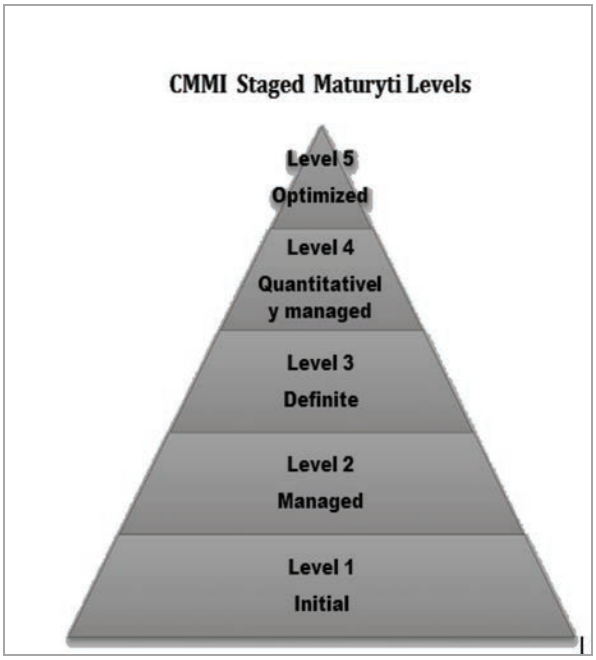


Fig. 5. Analysis of the application of the CMMI Software Quality Model.

These levels allow the organization to describe a path of evolution, which is ideal for companies that wish to improve its development and maintenance processes for its services and products. As you can observe in Figure 5, the levels can also be the result of evaluation activities. These evaluation activities can be applied to small companies or to groups within a bigger company (such as a group of projects or an area of the company.

CMMI has two improvement methods. One allows the company to increasingly improve the individual processes from the area the company selected. The other method allows the company to improve a set of related processes by treating the increasing successive levels of the process.

Level 1.
Initial: The process is carried out, and some previously identified aspects of the products are transformed.

Level 2.
Repeatable: The process is executed in the same controlled way.

Level 3.
Defined: The process is defined and always executed in the company.

Level 4.
Arranged: Process execution has been institutionalized in the company, and has an objective, quantitative system for measuring its capacity.

Level 5.
Optimized: Organizations with a Level 5 maturity have defined, continual improvement processes that they nurture with the quantitative information described in Level 4.

MoProSoft
MoProSoft [14] states that the Mexican software industry intended to apply this norm to its Software Industry, which is mostly comprised of small and medium companies. It is a model based on the best practices observed internationally. It intended to be simple and easy to apply, and to be adopted without major costs. It was also conceived to help companies achieve good evaluations in other quality models such as CMMI (Capability Maturity Model Integration).

The MoProSoft Model is directed towards software development and maintenance areas, as you can observe in Figure 6.

Superior Direction Category (DIR)
A process category concerned with Superior Direction practices, related to business management. It provides guidelines for Management levels and its feedback is the information generated by them.

Management Category (GER)
A process category that covers process management practices, projects and resources, according to the guidelines of the Superior Direction Category. It provides the necessary elements for the Operations Category, receives and shares the results with the Superior Direction Category.

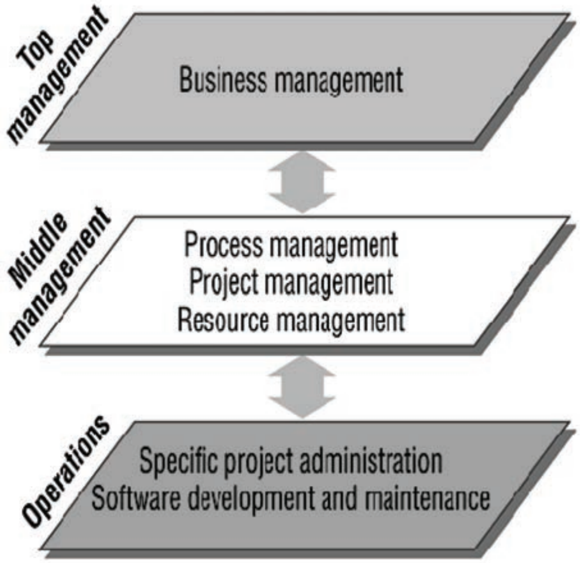


Fig. 6. General structure of the MoProSoft Model

Operations Category (OPE)
A software development and maintenance process category. It carries out activities according to the elements provided by the Management Category and turns information in to them. The MoProSoft Capacity Levels are taken from the ISO-IEC 15504 norm.

Level 0
Incomplete: The process is not implemented, or it doesn't achieve its purpose. Products or process results are not easily identified.

Level 1
Completed: The purpose of the process is achieved on general terms, though it is not rigorously planned or carried out. There are some identifiable products that evidence the achievement of the project's goals. The process is managed the deliverables are the result of specific, planned, continuous procedures, with quality, time, and resource requirements.

Level 2
Arranged: The process is arranged and the deliverables are the result of specific, planned, continuous procedures, with quality, time, and resource requirements.
The goals of the process are easily identified, process performance is planned and supervised, and the responsibilities and authorities for each process are assigned and communicated.

Level 3
Established: A process is carried out and managed using a defined process based on the best practices for Software Engineering. In this level, the basic elements of a standardized process are determined, including adjustment guides, determining sequences and the interaction of the standardized process with other processes. There is also a quantitative understanding of process capacity and the improved ability to predict and manage performance.

Level 4
Predictable: It has defined processes that are implemented within the control limits established to attain previously defined process goals.

There is also a quantitative understanding of process capacity and the improved ability to predict and manage performance. In this level, the process' information needs are established according to applicable business aims (the quantitative aims of the project).

Level 5
Optimized: The process is optimized according to the business' current and future needs. Quantitative performance and efficiency goals are established according to the organization's objectives. Optimization is carried out by studying and adopting new, innovative ideas or with new technological products that enhance the defined process.

IT_MARK

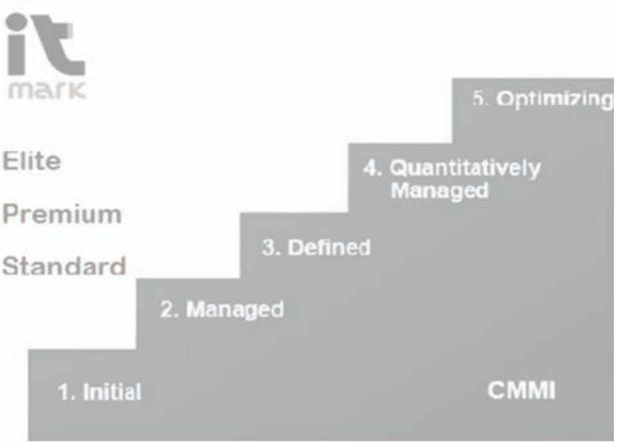


Fig. 7. IT_MARK Quality Model

The main objective of the IT_MARK Quality Model [15] is to provide a quality seal to small and medium software companies. It also seeks to improve the organizational effectiveness and the organization's market success through the improvement of its processes, as can be observed in Figure 7. The service identifies strengths and weaknesses, and discovers what areas may be improved according to business objectives. It is the first International Quality Model designed specifically for small and medium companies. It is a scalable model that applies to small and micro companies in the Information and Communication Technology sector. It has three perspectives:

A. General Management of the company, according to the 10-squared model, which studies 10 process areas such as strategy, commercial, financial, product and service definition, market understanding, marketing, etc. to obtain an exhaustive vision of the company.
Each of these categories takes into account 10 aspects (some of them critical) that have something to do with the company's state of development: Seed, Start-up, Development or Expansion.

B. IT_MARK's Information Security, based on norm ISO/IEC-17799:2005, defines several levels: the organization's security, responsibilities, legal requirements (such as the protection of personal data, etc.), and the necessary security controls that are required, has made it necessary that security management become a standardized process within the organization. Continuing improvement of the Information Security Management System.

C. Software and System Development Processes and the core of the model are based on CMMI.

IT_MARK
is a Quality Model based on the CMMI. Its levels are either Elite, Premium or Unused. It is the first International Quality Model designed specifically for small and medium companies. It is a scalable model that applies to small and micro companies in the Information and Communication Technology sector.

Systemic Model for Quality (MOSCA)
This model [16] is based on:
✓ Quality Model for a software product with a Systemic focus.
✓ Quality Model for a software process with a Systemic focus.

Some of the characteristics that this model takes from other Quality Models are: internal and contextual aspects, such as the model's partial quality, and Callaos' System Quality. This model integrates the above models to guarantee guality. Its levels are the following:

Level 0
Dimensions: The internal aspects of the process, the contextual aspects of the process, the internal aspects of the product, and the contextual aspects of the product, are the four dimensions of the model prototype. Only a good, balanced interrelation between them guarantees the global systemic quality of a company.

Level 1
Categories: Eleven (11) categories are taken into account: six (6) relate to the product and the remaining five (5) relate to the development process. This division does not mean that these aspects are not related; it is only used to identify what area or sub-model they belong to.

Level 2
Characteristics: Each category has an associated set of qualities which define the key areas that need fulfillment to achieve, insure, and control the quality of the product and the process. The characteristics for each product category include a series of process characteristics.

Level 3
Metrics: A series of metrics are proposed for each characteristic, and they measure quality systematically.

Characteristics of the software development process have a direct impact on product categories. The quality of the finished product is greatly influenced by the quality of its development process. However, each category in the product model depends specifically on certain characteristics of the process model, which would mean that if all these characteristics were completely fulfilled during the product development, the product will fulfill at least a some of the quality levels.

Process characteristics influence:
1) Software Product Functionality.
2) Software Product Reliability.
3) Software Usability.
4) Software Sustainability.
5) Software Product portability.

PSP (Personal Software Process) TSP (Team Software Process)

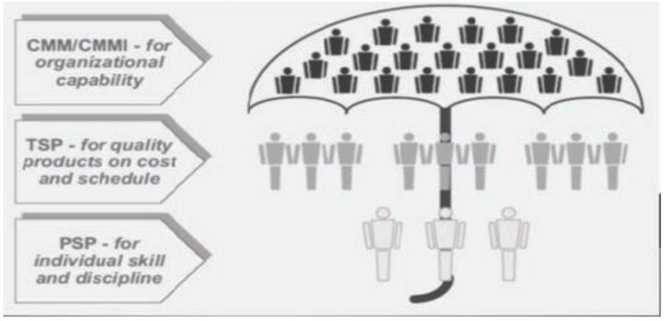

CMM/CMMI - for organizational capability
TSP - for quality products on cost and schedule
PSP - for individual skill and discipline

Fig. 8. PSP (Personal Software Process)/TSP (Team Software Process)

TSP/PSP [17] is a model designed to help software engineers do their job better. It shows how to apply advanced engineering methods to their daily tasks, offering them detailed planning and estimation methods, teaching them how to optimize their time and increase their performance, as you can observe in Figure 8.

Level 1
Personal measures: It includes stages PSP 0 and PSP0 1. In stage PSP 1, the development personnel take elementary measures concerning current processes. They estimate the development time and the number of defects in the programming. Task PSP0 1 standardizes the coding and coding evaluation processes.

Level 2
Personal Quality: It incorporates stages PSP1 and PSP1.1. PSP 1 makes reference to personal planning. It introduces the estimation of resources and software size. In stage PSP1.1, the schedule is defined and the tasks are planned.

Level 3
Cyclic personal process: it refers to stages PSP2 and PSP2.1. PSP2 consists of personal quality. During this phase, the coding is revised to correct mistakes. Stage PSP2.1 introduces techniques for design specification and verification.

Level 4
Continuing improvement [18]: it refers to stage PSP3, which combines multiple processes from PSP2.1 in a cyclic way, to uphold larger scale development projects. (Hayes 1997).

Characterization of current quality models through the analysis and selection of the best practices that can be applied to the proposed model.

The following chart shows a comparison between the processes used by some current models, the type of processes they are used in, and the types of processes they are not used in.

The CMMI Quality Model specifically takes into account the maturity of the organizations in their software production processes. On the other hand, PSP is a series of organized and disciplined practices for time management and the personal productivity improvement of the programmers or software engineers in development tasks and system maintenance. It is aligned and designed to be used by organizations with CMMI or ISO 15504 process models.

Similarly, TSP integrates high-performance software development teams by following a set of structured processes that indicate what to do in every stage of the development process to construct an organized finished product. The MOSCA quality model focuses on internal and contextual product aspects and internal process aspects.

Finally, the IT_MARK model studies the technical and business processes and it is specifically designed for PYMES in the IT sector to measure their Excellence in Information Technologies. We can also say that it is a key service designed for PYMES which helps position them through a continuing, sustainable improvement.

A comparison between the characteristics of PYMES' Processes and the Quality Models that were analyzed is shown in Table 1.

Process Characteristics	CMMI	PSP	TSP	MOSCA	IT MARK
Measurement and Analysis	□	✗	✗	✗	
Cause Analysis and Solution	✓	✗	✗	✗	✗
Configuration Arrangement			✗		
Analysis Decision and Solution		✗	✗	✗	✗
Integrated Management Project	□				
Measurement and Analysis	□	✗	✗		
Organization Innovation and Deployment	□	✗	✗	✓	✗
Organization Process Definition	□	✓	✓	✓	✓
Focus on Organization Processes	□	✓	✓	✓	✗
Organization Process Performance	□	✗	✗	✓	✗
Organization Training	□	✗	✗	✓	✓
Project Monitoring and Control	□	✗	✓	✗	✓
Project Planning	□	✗	✗	✓	✗
Product process and Quality Assurance	□	✗	✗	✗	✓
Product Integration	□	✗	✗	✗	✓
Quantitative Project Management	□	✓	✓	✓	✗
Requirements Management	□	✓	✓	✓	✓
Development Requirements	□	✗	✗	✗	✗
Risk Management	□	✗	✗	✓	✗
Provider Management	□	✗	✗	✓	✓
Solution, Validation and Verification	□	✗	✗	✗	✓
Defect Prevention	✗	✓	✓	✗	✗
Change Management Technology	✗	✓	✓	✓	✗
Change Management Process	✗	✓	✓	✓	✗
Qualitative Software Management	✗	✗	✗	✓	✗
Training Program	✗	✗	✗	✗	✗
Product Engineering Software	✗	✓	✓	✓	✗
Group Integration Coordination	✗	✗	✓	✗	✓
Paired Revision	✗	✓	✓	✗	✗
Quality Control Software	✗	✗	✓	✓	✓

4. DISCUSSION

The selected processes for the creation of the Quality Model were taken from other models after analyzing the activities and characteristics of the PYMES from which they were selected:

- Management Category
- Resource management
- Process monitoring and control
- Project management
- Organizational training
- Software development and training
- Organizational process approach
- Requirement management
- Changes management

Operations Category

- Revision
- Group coordination and integration
- Solution, validation, verification
- Finished product

The model was elaborated with these processes, seeking to become a new resource that can be applied and used for a better use of time and resources, creating more profitable products.

PYME Software Project

The implemented model is shown in Figure 9.

5. CONCLUSIONS

Through this research project, several characteristics and relevant aspects of PYMES have been identified, which will help make decisions that favor the service provision ability of a PYME.

Every small company must take certain guidelines into account to improve personal, economic and business performance.

A research project is a long, arduous process that helps reinforce what was learned during the university course.

6. ACKNOWLEDGEMENTS

This article is the result of a research project named “Quality Model for software developing PYMES located in the Metropolitan Area of the Aburrá Valley”. The authors thank the institution and Politécnico Jaime Isaza Cadavid Project Evaluation Committee for their valuable contributions, which helped to continuously improve this project.

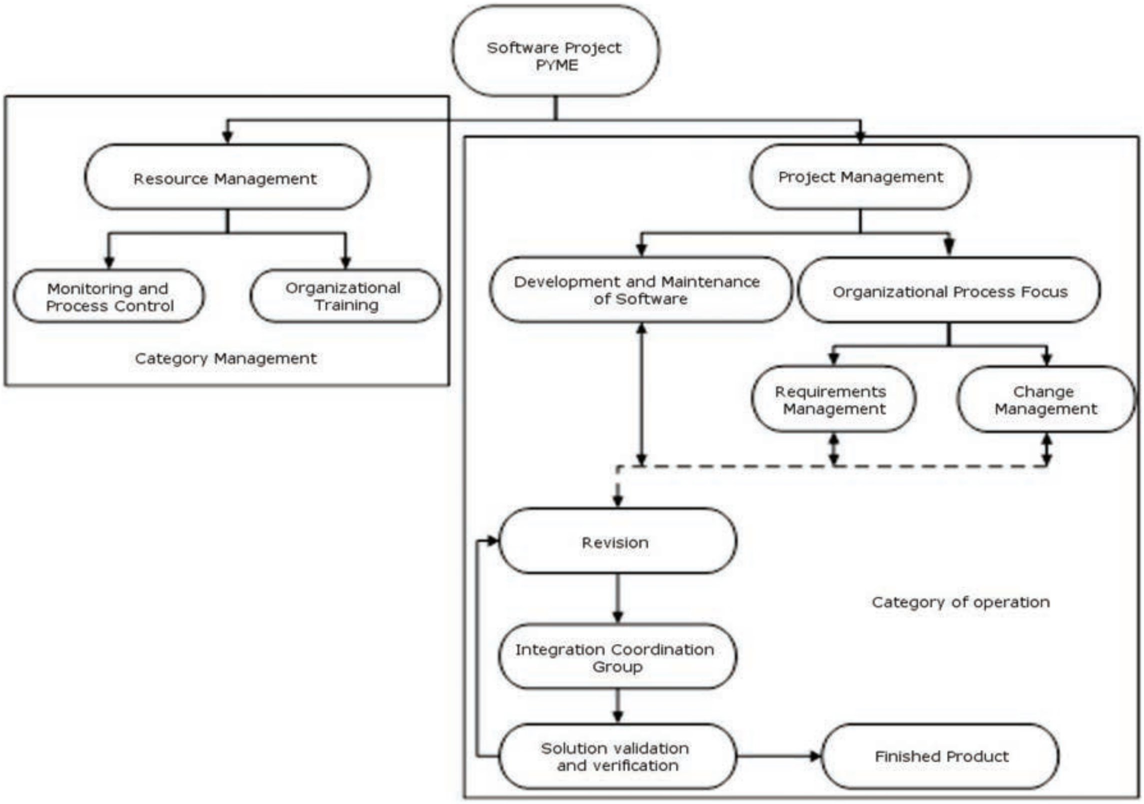


Fig. 9. Implemented model.

7. BIBLIOGRAPHY

[1] BECK, Katia. "Extreme Programming Explained. Embrace Change", Pearson Education, 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.

[2] BOEHM, B. y TURNER,R. Balancing Agility and Discipline, Addison Wesley, 2004.

[3] CÁCERES, P.Y. Marcos, E. Procesos agiles para el desarrollo WEB. Departamento de Ciencias Experimentales e Ingeniería
Universidad Rey Juan Carlos. Madrid. España. 2001.

[4] DÍAZ, José Ramón. Las metodologías ágiles como garantía de calidad del software REICIS.

Revista Española de Innovación, Calidad e Ingeniería del Software. Disponible en: http://www.redalyc.org/articulo.oa?id=92217181006 [fecha de consulta: 17 de septiembre de 2014].

[5] NAVARRO, José Manuel; GARZÁS, Javier. Experiencia en la implantación de CMMI-DEV v1. 2 en una micropyme con metodologías ágiles y software libre. REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software, vol. 6, no 1, 2013.

[6] YAGÜE, Agustin; GARBAJOSA, Juan. Comparativa práctica de las pruebas en entornos tradicionales y ágiles. REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software, 2013, vol. 5, no 4. Disponible en: http://revistas.ojs.es/index.php/REICIS/article/view/ 1489. [fecha de consulta: 17 de septiembre de 2014]

[7] LETELIE, Patricio; PENADÉS, Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).Disponible en línea. [Citado el: 17 de septiembre de 2014].

[8] SAMAMÉ SILVA, Jaime Humberto. Aplicación de una metodología ágil en el desarrollo de un sistema de información. 2013. Disponible en: http://tesis.pucp.edu.pe/repositorio/handle/123456789/5044. [Citado el: 17 de septiembre de 2014].

[9] Highsmith J., Orr K. "Adaptive Software Development: A Collaborative Approach to Managing Complex Systems". Dorset House. 2002.

[10] Highsmith, J. "Agile project management. (2009). Disponible en: < http://books.google.com.co/books?id=VuFpkztwPaUC&printsec=frontcover&hl=es#v=onepage&q&f=false>. [fecha de consulta: 17 de septiembre de 2014]

[11] HURTADO J, PINO, F. Y VIDAL, J. Software Process Improvement Integral Model: Agile.SPI.Technical Report SIMEP-SW-O&A-RT-6-V1.0.University of Cauca. Colciencias. 2006.

[12] ISO, ISO/IEC 15504-2:2003/Cor.1:2004(E). Information technology -Process
Assessment- Part 2: Performing an assessment. International Organization for Standardization, 2004.

[13] INTECO, Estudio sobre la certificación de la calidad como medio para impulsar la Industria de desarrollo del software en España, INTECO, (2008). Disponible en: http://www.inteco.es/Calidad_del_Software/estudios_e_indicadores/publicaciones/calidad. [fecha de consulta: 10 de Junio de 2014]

[14] Introducción al proceso personal software Sm. Disponible en: <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r35043.PDF>

[15] MARTIN, R. "Continues Care vs. Initial Design". Disponible en: www.objectmentor.com/resources/articles/Continuous_Care.pdf. 2002.

[16] PESSAGNO, Leslibeth, et al. Modelo de calidad para herramientas FLOSS que dan apoyo al modelado de procesos del negocio. REICIS: Revista Española de Innovación, Calidad e Ingeniería del Software, 2013, vol. 4, no 2. Disponible en: http://revistas.ojs.es/index.php/REICIS/article/view/ 1456. [fecha de consulta: 17 de septiembre de 2014].

[17] MONSALVE, Luis. Calidad de los Productos Software. Disponible en: http://www.inf.udec.cl/~revista/ediciones/edicion1/lmonsalve.PDF. 2002.

[18] MURUA, juan. Metodologías para el Desarrollo del Software. 2005.