Development Model for Technological Products

Freddy Alexander Díaz González
Centro de Investigación y Desarrollo Tecnológico de la Industria Electro
Electrónica e Informática (CIDEI)
Bogotá, Colombia

Jonattan Andrés Andrade Mayorga
Centro de Investigación y Desarrollo Tecnológico de la Industria Electro
Electrónica e Informática (CIDEI)
Bogotá, Colombia

María Ximena Cáceres Prado
Centro de Investigación y Desarrollo Tecnológico de la Industria Electro
Electrónica e Informática (CIDEI)
Bogotá, Colombia

Andrea Carolina Corredor Bedoya
Centro de Investigación y Desarrollo Tecnológico de la Industria Electro
Electrónica e Informática (CIDEI), acorredor Bogotá
Colombia

Abstract

This paper describes the model for Technological Product Development (TPD) that collects best practices of the methodologies used worldwide, such as Rational Unified Process for Systems Engineering (RUP SE) and Méthodologie de Conception des Systèmes Electroniques (MCSE), also, the Know-How of development companies of technology products in the electro electronic sector of Bogotá region. The TPD model focuses on the development of engineering systems, considering the use of documentary tools like Real Time Unified Modeling Language (UML RT) and System Modeling Language (SysML), to facilitate understanding of the content and improve communication within the workgroup. Additionally, the model features a pyramid-shaped architecture that guides the development and documentation of the process, and incorporates the PDCA cycle to ensure quality processes and continuous improvement.

Keywords: development model, architecture, engineering system, PDCA cycle, modeling languages

Introduction

Based on the characterization of the development companies of electro electronic sector in Bogota region, conducted by the *Centro de Investigación y Desarrollo Tecnológico de la Industrial Electro Electrónica e Informática* (CIDEI) were identified the following shortcomings and problems associated with the development of machinery and electro electronic equipment: rework for lack of a structured methodology, lack of documentation of the projects implemented, lack of traceability in the requirements requested by stakeholders, delays by staff turnover and lack of structured test protocols to ensure quality developments. Currently there are several models for the development of engineering systems, such as the *waterfall model, spiral model and the V model*, which are used by formal methodologies such as *Rational Unified Process for Systems Engineering* (RUP SE) and *Méthodologie de Conception des Systèmes Electroniques* (MCSE). These methodologies are the most used to develop engineering systems around the world, however, do not include all areas of knowledge that can contain a technological product or specific conditions of the Colombian environment.

As a solution to the problem, is proposed a model to develop engineering systems involving areas such as software, electronics, mechanics, electrical, industrial and industrial design. This model is considered within a quality approach that ensures compliance with the requirements raised by stakeholders and also presents tools for document development as modeling languages.

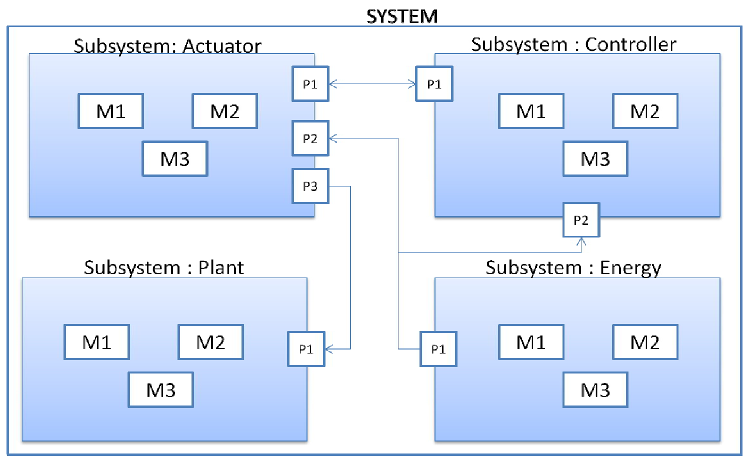
This article presents the definition of systems engineering applied to the development of technological products, the main development models currently used, and identifies aspects that hinder their applicability in the locale and finally presents the developed model with its components.

1. Engineering System

Engineering systems can be described by their structure, function and purpose; they must consider the needs of the customer, the operating environment, communication interfaces, maintenance and operating personnel capabilities, these features should be correctly reflected in the design of the system (Kossiakoff, Sweet, Seymour, & Biemer, 2011). Technological products currently required domestically combine the traditional engineering disciplines, such as software, electronics, mechanics and electrical, sometimes also requires knowledge of industrial design. Due to the multidisciplinary condition of these systems, it is necessary to use development models and modeling languages that allow properly link the areas involved in the design and development phases.

Figure 1 shows the composite structure diagram containing the subsystems of a typical engineering system that allows specifying the relationship between subsystems, modules and components, through the connections between the ports (P1, P2, etc.). Which vary according to the specific conditions of development.

Figure 1. Composite structure diagram of the typical engineering system



To guaranty proper performance of the system, each subsystem must meet the requirements set individually and in combination with other related subsystems, therefore, each unit of the system cannot be designed in isolation and then assembled. Additionally, the design and implementation of the system must consider the partial information required by each stakeholder (known as views of architecture) and how it should be documented.

The development of an engineering system includes the following knowledge:

- Main or theoretical, represent a formal approach to the solution of the problem and are generally employed in new situations and / or in structured environments
- Practical or empirical, represent the accumulated experience, which has allowed the development of standard operating policies for well-structured problems
- Perspective or user, represent the opinion held regarding future directions and realities in the technology under consideration

2. Systems Development Model

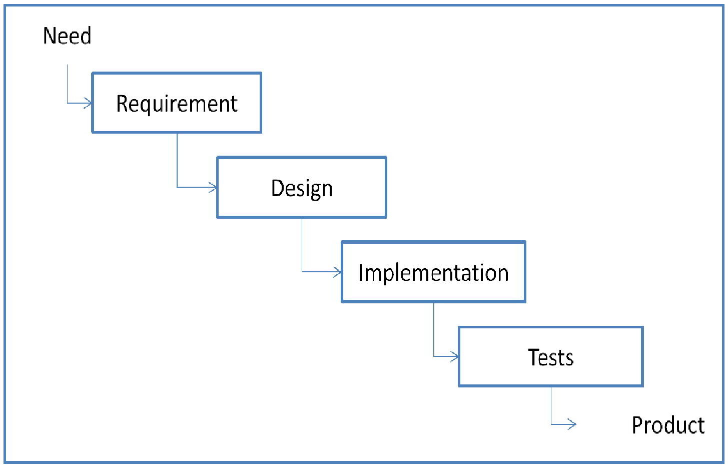
During the project execution it is important to establish the steps to transform customer needs into a product or prototype. To facilitate this process, there are some development models, used mostly for software development.

The three most popular models are the waterfall model which poses a series of sequential steps, where the output of one becomes the input of the next (S. & Murugaiyan, 2012). The spiral model raises product development through incremental approximations (CoFluent Design, 2009). Finally the V model, proposes a focus on design verification, associating each development stage to one testing stage. Each model has unique features that allow them to be used in different types of project.

2.1 Waterfall Model

It contains a series of steps that include the entire product development sequentially, from the definition to its operation. As shown in Figure 2, the result of each stage is tested and verified before proceeding with the next, when a stage is completed, it is closed and the next is started.

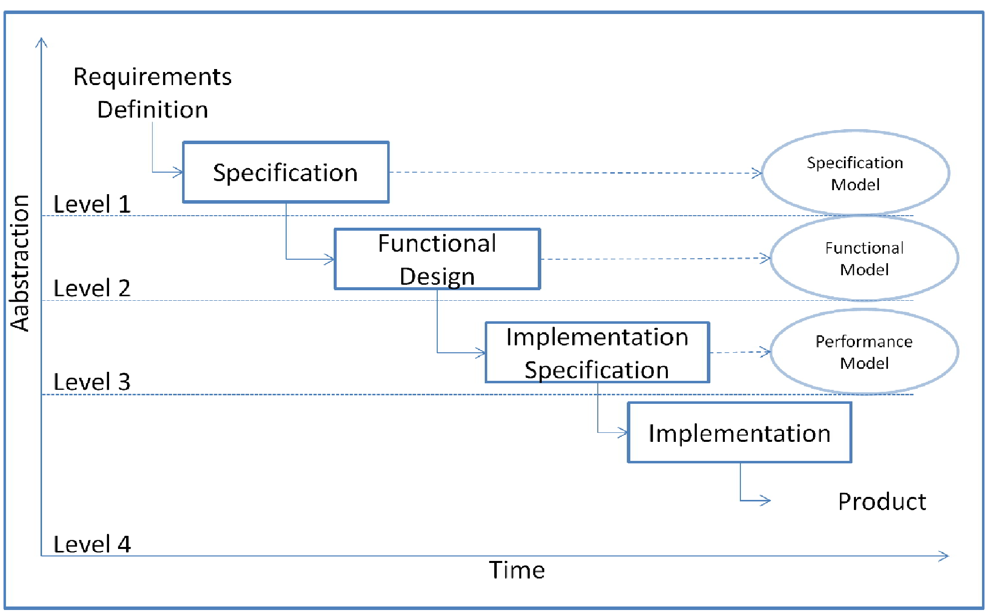
Figure 2: Waterfall Model



Source: Based on: J. P. Calvez, *EMBEDDED REAL-TIME SYSTEMS a Specification and Design Methodology*. New York: John Wiley & Sons, Inc., 1993.

One advantage of this model is that the documentation should be performed upon completion of each stage and before proceeding with the next, ensuring the quality of the process. However, there are not contemplated contingency or corrective actions in case of late identification of requirements in the development process. One methodology that uses the waterfall model is the methodology of design of electronic systems (MCSE), owned by the Co-fluent group. The process is based on a top-down development and contains a series of steps that generate a deliverable at each stage. In some special cases, designers can work simultaneously in several steps, considering the dependency between steps. MCSE consists of four phases and a preparatory step, which allows traceability of design, capitalization and intellectual property reuse in all stages of the process. Figure 3 presents the stages of the method and the level of design abstraction over time.

Figure 3: MCSE Phases



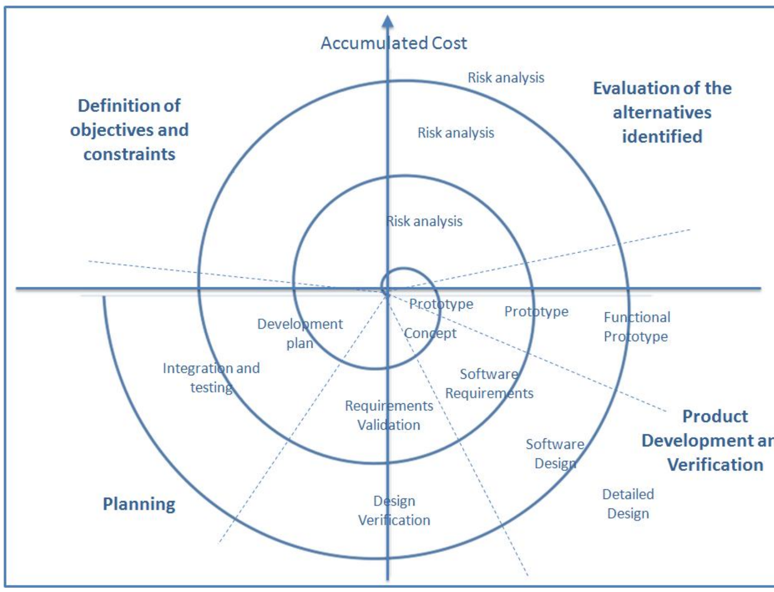
Source: Based on: J. P. Calvez, REAL-TIME SYSTEMS. New York: John Wiley & Sons, Inc., 1993.

Additionally, MCSE uses the System Modeling Language (SysML), developed by OMG, this is considered an extension of UML, focus on the development of systems engineering and contains graphical tools to describe the mechanics, electronics, software and electrical areas; in a statical and dynamical way. One of the most important tools is the requirement diagram whish establishes the requirements and specifies dependencies graphically (OMG, 2006).

2.2 Spiral Model

The incremental spiral model raises an iterative sequence in which each of the activities is repeated cyclically increasing its scope at each iteration. The stages in this model are represented in each of the quadrants of Figure 4 and are: Planning subsequent phases, defining objectives, alternatives and constraints, solution assessment and finally, risk analysis development and product verification.

Figure 4: **Spiral Model**



Source: J. P. Calvez EMBEDDED REAL-TIME SYSTEMS a Specification and Design Methodology. New York: John Wiley & Sons, Inc., 1993.

One methodology that uses the spiral model is Rational Unified Process for System Engineering (RUP SE) of IBM, this covers areas of knowledge and electronic engineering, software engineering and systems engineering. This methodology is directed to development of engineering systems with emphasis on software design. The RUP SE management has an iterative approach, where the intensity of the repeats varies according to the characteristics of the project and the spiral structure is conserved, facilitating the construction of an incremental prototype.

RUP SE proposes strategies to manage communication between the development team, applicable for both small workgroups to large multidisciplinary teams. One strategy is the definition of views that are a mainstay of description of the development, among them there are: business view, computing, engineering, and process information. The views are described through the language modeling Real Time Unified Modeling Language (RT UML) that facilitates the documentation throughout the process (Rational Software Corporation, 2002).

RT UML is based on UML, presents tools for describing systems with constrained time, containing hardware and software components (Powel Douglass, 2011). This language raises the use of diagrams to document the development of the system, which are:

- Composite Structure Diagram: illustrate statically the composition of the system to be developed
- Time Diagram: represent objects that change its state in a given period
- Use Case Diagram: represents the overall system functionality

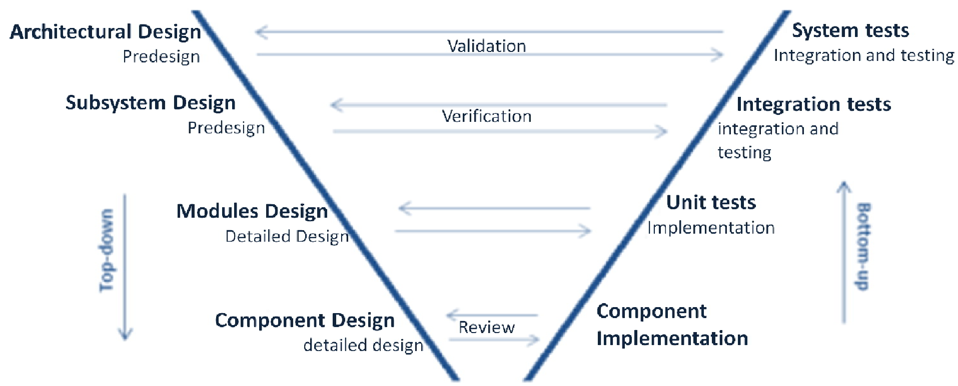
Currently engineering systems involve various areas of knowledge, including mechanical engineering, electrical engineering, electronic engineering, software engineering, industrial engineering, industrial design, among others.

These areas contain constraints that must be considered in the development process due to their nature and are not fully compatible with spiral model, because development of Machinery and electro electronic equipment requires a gate, where decisions are made to repeat an activity or continue with the next, which can hardly be iterative. For example, the implementation of mechanical and electrical components can only be started once the design is complete and it is a phase which should be done only once in order to avoid overruns in materials and machining.

2.3 V Model

This model is focused on "Verification and Validation" at different levels of abstraction (CoFluent Design, 2009). Uses the top-down strategy for design and Bottom-up for the implementation, which means that while in the detail level of the system increases on the design, on the implementation is decreased; tests and design are implemented parallel. This model is appropriate for developments that must be constantly validated in project implementation; however, the V model does not specify neither documentation tools nor design tools for the development of multidisciplinary technology products. Figure 5 shows the V model.

Figure 5: V Model



Source: Based on: J. P. Calvez, *EMBEDDED REAL-TIME SYSTEMS a Specification and Design Methodology*. New York: John Wiley & Sons, Inc., 1993.

3. Model of Technological Products Development

The model of technological products development is based on international methodologies for the development of engineering systems such as *Rational Unified Processes for System Engineering* (RUP SE) and *Méthodologie de Conception des Systèmes Electroniques* (MCSE), which use their own development models to develop products that involve specific knowledge areas such as software engineering and electronic engineering, however, are not recommended for the development of systems with mechanics, electrical or industrial design, because of the constraints and special conditions resulting from these areas are not considered on these methodologies. Additionally, this model is part of Methodology for development of machinery and electro electronic equipment, which was developed by the *Centro de Investigación y Desarrollo Tecnológico e Informático* (CIDEI).

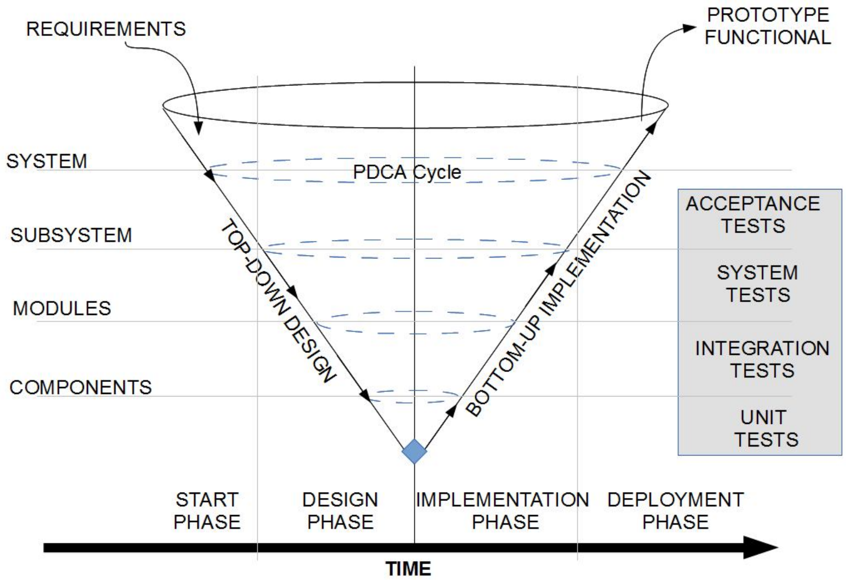
Initially, it identifies aspects that must be considered in a development model, through the characterization of the development companies of the electro electronic sector in Bogota Colombia, made by CIDEI; in which the following criteria were established to develop engineering systems: incremental implementation using quality standards, design by levels of abstraction, planning the testing protocol according to abstraction levels defined in the development, the definition of indicators to establish decision points to manage the development process and the use of tools to document the projects executed.

Figure 6 presents the proposed model, which focuses in two stages: Design and Implementation. The design stage follows the top-down strategy and starts with the identification of project requirements, then, from an analysis, identifies constraints and system-level requirements, subsystems, modules and components, and finally proposes a design solution for each level of abstraction. The Implementation stage follows the bottom-up strategy whose philosophy is to develop first the smaller parts of the product (called components in the model), and build incrementally modules and subsystems, integrating them, in order to assemble the functional prototype that will be delivered to the customer. This stage begins once the design is complete; executing tests during the implementation, which are directed by the Process approach and quality management.

Additionally, this model proposes the planning, implementation, verification and correction of the tests for each level of abstraction. Below are the components of the testing protocol that includes this model:

- Unit Tests: check of each component and module
- Integration Tests: include the verification of the integrations between components, modules and subsystems
- System Tests: it is focused on verifying compliance with the requirements set at the beginning of the project
- Acceptance Tests: Verify the customer satisfaction with the delivered product, through the needs analysis expressed at the beginning of the project.

Figure 6: TPD Model



3.1 Model approach based on processes and quality management

The TPD model poses work organization through processes, where the activities and tasks associated with the development, should specify not only their inputs and outputs but also the sequence that will be executed. The aim of proposing a development model focused on processes is to ensure the effectiveness of quality management systems, which are designed to fulfill the requirements set by the client and the rules governing the product. This quality management is emphasized on:

a) Understanding and meeting the requirements
b) The need of considering processes in terms of added value
c) The obtained results of process performance and effectiveness
d) Continuous improvement of processes based on objective measurement (Icontec, 2008)

The core of the process approach is the "continuous control", which corresponds to PDCA or Deming cycle, which proposes the following activities within each process:

- Planning: is to establish the necessary tasks to meet the objective of the process, where should be established the test that will verify compliance and control points that indicate the progress of each task.
- Do: the planned processes must be run, considering the specified standards
- Check: performs a monitoring and control to the process, through the established indicators; once the task is complete, the implementation is verified through the planned tests
- Act: Once the tests are complete, errors or non-compliance of the implementation must be identified in order to be corrected in the product and in the process, ensuring the quality of development and continuous improvement of the processes

3.2 Architecture of the model for technology products development

The TPD model has an architecture that structures the development of machinery and electro electronic equipment and allows the process documentation in an organized manner.

An architecture is a set of elements that allow cover all necessary aspects of a product, summarizes the project documentation and contains the information necessary to understand the system and its operation (CoFluent Design, 2009) (Kruchten, 2004).

Pyramidal architecture proposed in this model is based on the 4 +1 RUP architecture and the three views architecture of MCSE methodology. Four views are proposed to describe the product through different angles in order to provide targeted information to each stakeholder, including the development team.

Figure 7 presents the proposed architecture, which contains three base views: Logic, Physics and Process, and a top view of services, from which others are derived. The following describes each of the proposed views:

- **Logical View:** contains information related with the subsystems, modules and components of the product, which describes their dependency and conceptual operation to generate models of behavior that can be analyzed by the project designers.
- **Physical View:** displays information related with the implementation of the product, as a description of the physical components, references, features, connection interfaces and their distribution in the enclosure. It is aimed at installers and maintenance personnel.
- **Process View:** describes the behavior, detailed communication and events, for each component, module, subsystem and system product. Once each unit of product is specified with a level of detail, the design can be started.
- **Services View:** Specifies the services provided by the product to the users and contains the requirements and needs expressed by the customer. It is aimed at customers, investors, company managers and contains relevant information to the development of base views.

Figure 7: Pyramidal Architecture

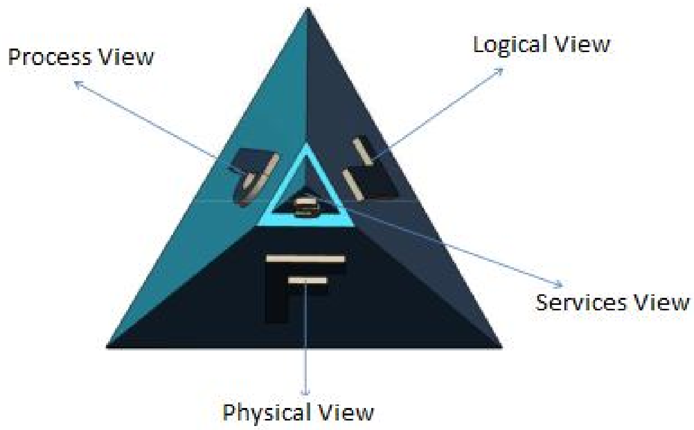
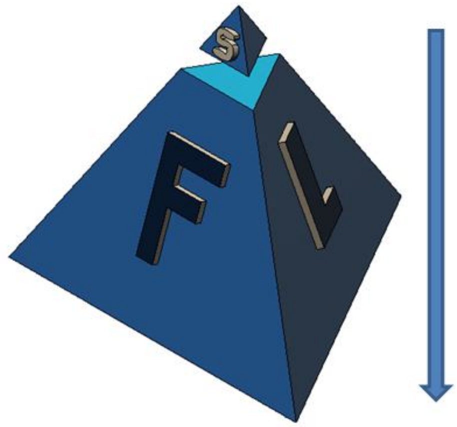


Figure 8 shows a front view of the pyramidal architecture, where it is stated that the top view is the basis for the development of others.

Figure 8: Front perspective of the pyramidal architecture



Once the content of the architecture is classified, according to the views presented, the TPD model proposes to document information through modeling languages *Real Time Unified Modeling Language* (UML RT) and *System Modeling Language* (SysML), with the aim of facilitating content understanding and improving communication within the working group. The combination of these languages provides a set of tools capable of describing an engineering system graphically and documenting the analysis and design phases.

Conclusions

The TPD model, guides the development of engineering systems, which, applied to a methodology that implements its strategies and guidelines, can be used for micro, small and medium enterprises for the development of technological products Testing protocol proposed in the TPD model, allows to optimize implementation phase reducing the time spent in the error detection and consequently the overall development

The pyramidal architecture proposed in the TPD model allows to communicate and to document the progress in an agile way, distributing the information according to the interests of each stakeholder The TPD model proposes the application of PDCA cycle in the macro processes, processes and subprocesses of product development in order to ensure compliance with product requirements and quality standards that govern it The TPD development model, provides communication between the engineering systems designers through the use of pyramidal architecture and the modeling languages as RT SysML UML

References

Castrillo Florez, N. (2010). Systems management: strategy, quality and knowledge. *Revista Vínculos Vol 7,* 51 - 59.

CoFluent Design. (2009). *The MCSE Methodology overview.*

Goncalves, M., Rodríguez, R., & Tineo, L. (2009). Adding fuzzy queries to software development. *Avances en sistemas e informática*.

Icontec. (14 de 11 de 2008). NORMA TÉCNICA COLOMBIANA NTC-ISO 9001. Bogotá, Colombia: Instituto Colombiano de Normas Técnicas y Certificación.

Kossiakoff, A., Sweet, W. N., Seymour, S. J., & Biemer, S. M. (2011). *Systems engineering principles and practice.* New Jersey: A JOHN WILEY & SONS, INC. PUBLICATION.

Kruchten, P. (2004). *The Rational Unified Process: An Introduction.* Addison-Wesley Professional.

Kuster, J. M., & Stroop, J. (2001). *Consistent Design of Embedded Real-time Systems with UML RT.*

OMG. (2006). *OMG Systems Modeling Language (OMG SysML).*

Powel Douglass, B. (2011). *Real-Time UML Worshop for Embedded Systems.* Newnes.

Rational Software Corporation. (2002). Rational Unified Process for Systems Engineering.

S., B., & Murugaiyan, S. (2012). *WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC.* International Journal of Information Technology and Business Management.

Sage, A. P., & Rouse, W. B. (2009). *Handbook of Systems Engineering and Management.* John Wiley & Sons.