Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2013

HUGO RAFAEL MENDES TAVARES

Metodologias de desenvolvimento de sistemas embedded

Development methodologies for embedded systems

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2013

HUGO RAFAEL MENDES TAVARES

Metodologias de desenvolvimento de sistemas embedded

Development methodologies for embedded systems

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Pedro Nicolau Faria da Fonseca, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, coorientação do Doutor Paulo Bacelar Reis Pedreiras, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho aos meus pais e irmão, pelo amor que me dão, pela educação que me deram, e por tudo o que me proporcionaram ao longo da vida. Pelo apoio e confiança depositados e, sobretudo, por me fazerem sentir muito feliz e amado.

o júri

presidente

Prof. Doutor José Luís Costa Pinto de Azevedo
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Mário Jorge Rodrigues de Sousa
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor Pedro Nicolau Faria da Fonseca
Professor Auxiliar da Universidade de Aveiro

Agradecimentos

Ao meu orientador, Professor Doutor Pedro Fonseca e ao meu coorientador, Professor Doutor Paulo Pedreiras, por todo o apoio, acompanhamento prestado e sobretudo pelos sábios conselhos. De igual modo, quero agradecer a todos os funcionários pertencentes aos quadros da Exatronic pelo acolhimento na empresa. Um agradecimento em especial à Engª. Sónia Martins que sempre me acompanhou e apoiou no necessário durante todo o período de desenvolvimento de trabalho para esta Dissertação.

A todos os meus amigos por estarem sempre presentes nesta caminhada, pela motivação e coragem prestados, procurando acompanhar todo o meu percurso, não só pessoal, como profissional.

Um agradecimento especial à minha família, pela constante atenção, amor e confiança, que semearam em mim o gosto pelo saber, proporcionando todas as condições para evoluir como pessoa. Por me terem oferecido a possibilidade de seguir o meu percurso de vida, sem o seu apoio e exemplo não teria chegado a esta fase.

palavras-chave

Sistemas embutidos, sistemas de gestão da qualidade, normas, metodologias de desenvolvimento, revisão de código, métricas, unit testing.

resumo

O presente documento aborda algumas metodologias de desenvolvimento de projetos e de sistemas embutidos, com enfoque em algumas técnicas para melhorar os produtos e serviços de modo a satisfazer as necessidades dos clientes.
Ao longo dos últimos anos, os requisitos funcionais de sistemas compostos por *software* registaram um aumento extensivo, devido ao avanço de várias tecnologias usadas em dispositivos. Num mundo cada vez mais competitivo, o nível de procura para o desenvolvimento de sistemas de gestão mais eficientes com o objetivo de garantir vantagem competitiva também aumentou. De forma a competir em mercados exigentes, as organizações precisam de adotar estratégias, visando a criação de valor das suas principais funções empresariais de modo a garantir a satisfação dos clientes. A competição pelo mercado exige às organizações que estas procurem alternativas para melhorar as suas metodologias de desenvolvimento.
Desta forma, foi organizado o necessário para o desenvolvimento de *software* embutido no âmbito do desenvolvimento de processos por forma a resolver os métodos atuais, analisando criticamente essas metodologias. Este trabalho foi também desenvolvido em ambiente organizacional, providenciado pela empresa Exatronic. No fim, obtém-se um conjunto de princípios para uma metodologia de desenvolvimento para este tipo de sistemas, com a possibilidade de ser aplicada às atividades da empresa.

keywords

Embedded systems, quality management sytems, standards, development methodologies, code review, metrics,unit testing.

abstract

Over the past few years, the functional requirements of systems comprised of software have increased extensively, due to the advancement of various technologies used in devices. In an ever increasingly competitive environment, the level of demand for the development of more efficient management systems as a means to achieve high levels of competitive advantage is also increasing. In order to compete in highly unpredictable markets, organizations need to adopt appropriate strategies, aiming at creating value out of their main business functions to guaranty high levels of customer service. Markets competition is driving organizations to find alternatives to improve their development methodologies.

The object of study focuses on a topic regarding the methodologies for project development and embedded development necessary to answer customer needs of products and services. The goal is to organize what needs to be done in embedded software development from the standpoint of development process by addressing current methodologies and critically analysing them. This work was also developed in an organizational environment, provided by the organization Exatronic. In the end, principles for a development methodology for this type of systems is obtained, with the possibility to be applied within the organization's activity.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ISO – International Organization for Standardization

NP – Norma Portuguesa

MME – Monitoring and Measuring Equipment

RDI – Research, Development and Innovation

QMS – Quality Management System

PDCA – Plan-Do-Check-Act

PM – Project Manual

C&D – Conceção & Desenvolvimento

I. Introduction

Over the past few years, the functional requirements of systems comprised of software have increased extensively, due to the advancement of various technologies used in devices. In an ever increasingly competitive environment, the level of demand for the development of more efficient management systems as a means to achieve high levels of competitive advantage is also increasing. In order to succeed, organizations need to develop strategies, aiming at assuring quality out of their main business functions to guarantee high levels of customer satisfaction. Markets competition is driving organizations to find alternatives to improve their development methodologies.

Issues arising from the increasing scale of development and increased complexity around these systems are posing a real challenge to organizations quality standards, since most organizations seek international and national certifications, such as the International Organization for Standardization (ISO), on all levels of management systems. These issues are mostly due to deficient development processes within the organization, in various aspects.

Exatronic is an organization specialized in investigation, design and development of innovative solutions of information, communication and electronic technologies. The organization intended to elaborate a study about development methodologies in the area of embedded systems in order to acquire valuable information for a possible integration into the organization's projects and continuously improve the processes of project development.

I.1 Objectives

This dissertation is presented in partial fulfillment of the requirements for the Master's degree in Electronics and Telecommunication Engineering at the University of Aveiro. The object of study focuses on a topic regarding the methodologies for developing embedded systems, still unclear by many organizations but, nevertheless, its benefits are quite desired.

The main objective of this research is to study various existing methodologies for project development and embedded development necessary to answer customer needs for products and services. The goal is to organize what needs to be done in embedded software development from the standpoint of development process by addressing current methodologies and critically analysing them. In the end, it is expected to obtain principles for a development methodology for this type of systems, with the possibility to be applied within the organization's activity.

For the student, the possibility of being received in a multidisciplinary team with several years of experience in developing electronic projects is particularly motivating.

I.2 Structure

Current dissertation is structured with four main chapters.

In the first chapter, the goal of the research is presented and its main features. This chapter describes the industrial context of the organization and the motives that led to the development of such research in the area of embedded systems. The chapter includes a brief introduction to the topic, a description of the main objectives to fulfill, and the structure of the dissertation.

The second chapter provides a brief overview of embedded systems. A historical perspective of quality and the evolution of how organizations address this subject up to modern days is also presented. This chapter also includes analysis to international and national standards, ISO 9001 and NP 4457 respectively, and finalises with an analysis of a project development method and a guide.

The third chapter starts with a revision on the organization's project development manual, entitled *Projects Manual*, which is applied within the *Conceção&Desenvolvimento* (C&D) sector. The goal is to verify how far the document is in compliance with the standards which the organization is certified and then propose changes, if required. The chapter follows up with a study on various methods, activities, practices and tools related to embedded software development in order to provide information and guidance for a possible implementation within the organization development processes.

The last fourth chapter discusses the main aspects of the research performed, its most critical conclusions and its limitations. Several other new opportunities for future research are proposed as potential elements related to this dissertation.

II. State of the Art

To systematize this chapter, a division by subchapters was made, each one corresponding to a main theme of analysis: Embedded systems, quality and quality management system, ISO 9001 and NP 4457 standards, PRINCE2 and ISO 21500 for project management.

II.1. Embedded Systems

Over the past few years, the functional requirements of systems comprised of software have increased extensively, due to the advancement of various technologies used in devices. This fact has led the development of embedded software to also expand in scale in various areas such as industrial automation, avionics, car industry, medical devices, energy production and many more.

Embedded systems are generally unnoticed by most people during their daily life, however, people may be constantly surrounded by dozens of embedded systems.

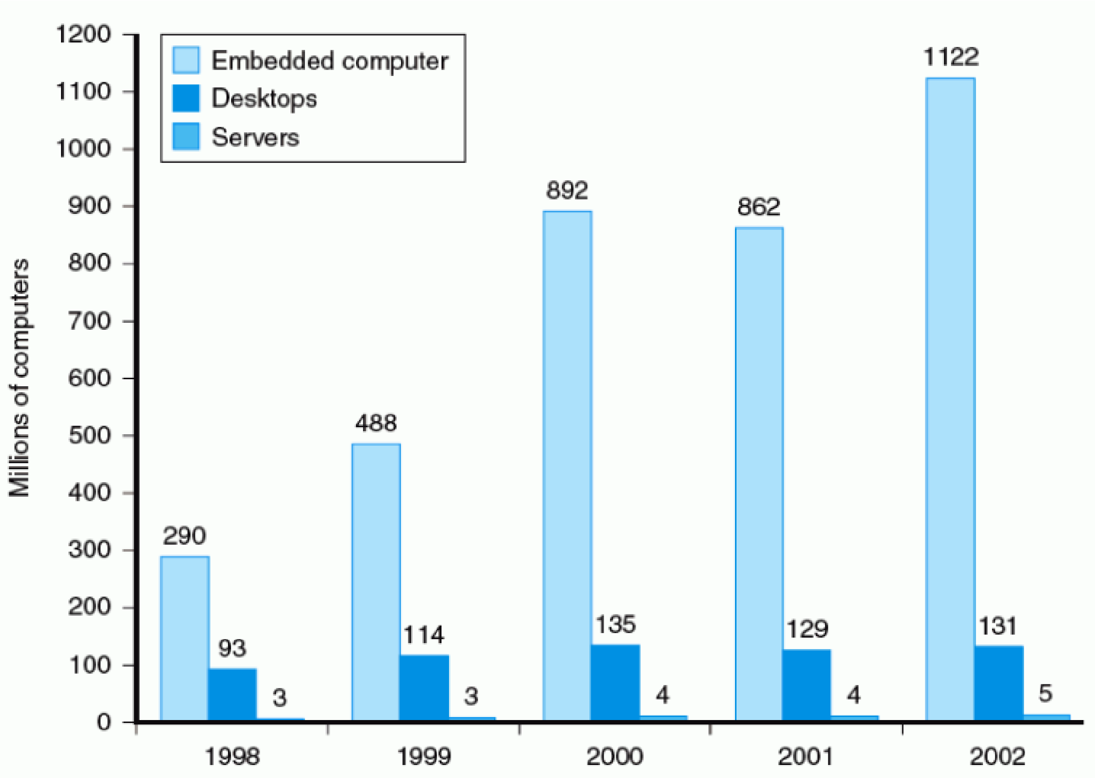

Figure 1 – Comparison of sold units between 1998 and 2002 of the three main computer systems (adapted from [1]).

The figure 1 shows the evolution of the main computer systems sales since 1998 to 2002. Besides being the class with the highest number of sales, the embedded systems have an annual growth rate higher than the other classes, which shows an increasingly tendency to sell these systems.

The development of these systems shows particular complexities, for which there is still no clear and universal solution. The embedded systems are normally based on microcontrollers that perform a set of functions embedded in a larger system, that is, systems that may also include electrical, mechanical and others elements, apart from the computational section.

When available, user interaction is done through specific interfaces; in other words, unlike computer based systems, they don't have a screen and a standard keyboard. They also differ from computer base systems because they are not generally designed to be configured or programmed by the common user. There is a wide variety of examples regarding embedded systems such as television controls, mobile phones, alarm systems and blood pressure monitors.

These systems are used in process control and applications management, in areas such as people safety or goods, which often have strict timing requirements to execute the task they were made for. In most cases, the input data of an embedded system is obtained through the use of sensors. Regarding the development of embedded systems, aspects related to the area, power consumption, development and production cost are often considered essential due to the high number of units built and the reduced or limited price for the target system. For this reasons, the resources and system performance should be only those strictly necessary for the system to perform its function properly.

Formerly, the scope of embedded development was somewhat limited and the developers did not require to be so conscious about development process to build the software required to be embedded in the final product. But with the increase in complexity and scale of software development, the development of embedded systems is bringing particular problems, since it gathers components from different development processes. As a result, more and more attention is given now to improve the development process to address these emerging problems.

The existing methodologies, which are mainly focused on software development, must be adopted based on:

- The difference in development speed and the different time cycles of the various components of an embedded system;
- The fact that there is a wide range of choices for the physical realization, both for the device and its interfaces, terms that are normally absent from the software development (the interface, the physique architecture and the base technology are almost always defined in advance and are considered immutable)

II.2. Quality and the Quality Management System (QMS)

Nowadays, quality is quite in the centre of attentions, which might suggest that this is a recent topic. However, it is possible to remit the origins of this growing concern to the decade 50 of the XX century [2], although it is reasonable to say that the concern in products quality is already present in ancient civilizations: Egyptian, Greek or Roman [3].

The quality, whether applied to a product or service, has many sides and they all are oriented to customer needs. These needs manifest in various ways, and it is up to the organization decide which quality dimensions are to be prioritised. An organization that is committed to improve its quality, should start by making the following steps: formulation of a quality vision, creation of teams for quality and quality planning, and finally, implementation of the plan [4].

II.2.1. Quality

Quality is a conditioning factor for the organizations life and it can somehow be a little confusing to define. The concept about quality can be so extensive and varied that even people related to it don't always share consensus, and they may go through several answers such as: perfection, consistency, cost reduction, fast delivery, compliance between policies and procedures, providing a useful product, performing properly at the first time and customer's satisfaction. In the end, quality is a bit of everything, hence, it is very important to be aware of all the perspectives about quality in order to evaluate the role it plays in the organization [5].

Deming, one of the great icons of the quality theme, defines quality as 'continuous improvement' that aims for an ultimate goal of zero defects. This, in practice, may not be economically viable. He also states that quality can only have the required attention if top management provides a lot of support and that quality improvement is the best way for an organization to remain competitive and alive in the market [6].

The following figure presents a brief historical evolution of the quality concept, showing evidences that the concern about quality exist since the beginning of 20th century and it is connected to the human nature.

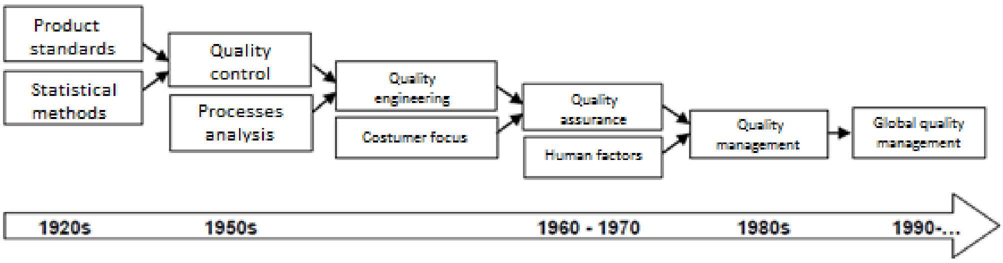

Figure 2 – Evolution of the Quality concept (20th century).

Competitive advantage is one of the aspects associated with quality and may result in the ability of an organization to achieve market superiority. This competitive advantage is obtained, very often, making use of quality, namely through the following points [5]:

- Product quality is important to generate profit in a given business;
- Organization that offer product/services with quality have, generally, big market shares and the process to insert products/services in the markets is easier;
- Quality is associated to high return investments;
- High quality of products usually leads to higher prices;

Like Deming, other names have excelled in the area of quality over the decades. Among them are Juran, Crosby, Feigenbaun and Ishikawa.

II.2.2. Quality Management

Juran, one of the pioneers in quality, associated competitive advantage to quality, although technology is also very commonly associated to it, especially in the literature. However, reality proves that technology innovations all alone are not enough to assure organizational competitiveness. Most of the times, quality is only used to bring recognition to organizations, but it should be managed as a fundamental resource to assure competitiveness.

During the 60s, in Europe, competitiveness entailed a strategy that, although quantitative, was oriented to larger markets, where products and services were already important.

Around the 80s, markets were showing a slow expansion and people started to realize that faster production and lower prices were not the solution for competitive advantage, which required a change in the current strategy. Organizations started to bet in quality production, which led to a qualitative strategy. Nevertheless, the costs related to quality were unknown and it was difficult to prove the contribution of quality in organizations. On the other hand, it is hard to comprehend if the organization competiveness is affected by the lack of quality in their products/services [7] or if the quality in organization is affected by the lack of competitiveness and poor performance by them, as states Longo [2].

It is understandable, in this context, why one of the most investigated areas in quality management is the impact it has in competitive edge. In general, organizations switched from a quantitative strategy to a qualitative strategy, and product/service differentiation was now related to quality. Market needs resulted in substantial changes in quality management and are the main factors for this strategy change.

Since then, an increasingly tendency to monitor quality was noticed, since the contractual stage till the use of the product/service. This complete life cycle management requires each function to attend a portion of the global quality objective.

It is possible to resume the evolution of quality control shapes, since the 2nd world war until current days, in the following figure:

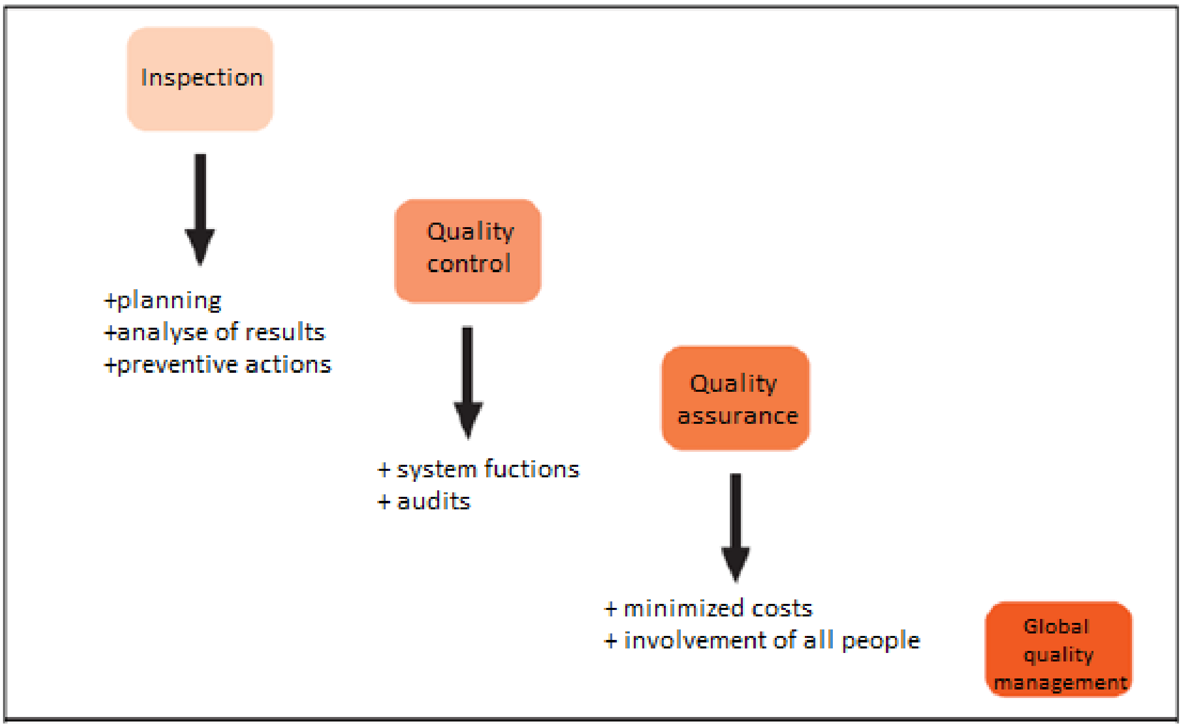

Figure 3 – Quality evolution since the 2nd world war till today (adapted from [8]).

II.2.3. Quality Management Principles

In order to understand the concept of quality and its management, the International standard ISO 9000 [9] defines eight quality management principles, established along the last decades and are based on the knowledge and experience gained by several scholars in this area:

1. Customer Focus

“Organizations depend on their customers and therefore should understand current and future customer needs, should meet customer requirements strive to exceed customer expectations.”

An organization that does not do this is most likely an unsuccessful organization. It is very important to comprehend what customers want in order to satisfy their needs and also be aware of how customers perceive the organizations performance.

2. Leadership

"Leaders establish unity of purpose and direction of the organization. They should create and maintain the internal environment in which people can become fully involved in achieving the organization's objectives."

It is up to the leaders to provide direction to an organization, to lead their people to a common goal, the success of the organization. They must develop a culture and environment in which people are committed to satisfy customer needs.

3. Involvement of people

"People at all levels are the essence of an organization and their full involvement enables their ability to be used for the organization's benefit."

Everyone contributes to the success of an organization, every role has a part to play in it and top management should make that clear.

4. Process approach

"A desired result is achieved more efficiently when activities and related resources are managed as a process."

Organizations should implement a process approach within their quality management system. By adopting this, everything that an organization do will always follow a process of identification, evaluation and definition which leads to more predictable results, lower costs and much more.

5. System approach to management

"Identifying, understanding and managing interrelated processes as a system contributes to the organization's effectiveness and efficiency in achieving its objectives."

The way that an organization manage must be consistent, this requires the organization to understand what a management system is.

6. Continual improvement

"Continual improvement of the organization's overall performance should be a permanent objective of the organization."

If there is an opportunity to perform better then that's how it should be performed. An organization must continually improve their products, processes and systems as their efficiency and effectiveness.

7. Factual approach to decision making

"Effective decisions are based on the analysis of data and information."

All decisions must be based on facts. This requires a program to gather data and analyse that data.

8. Mutually beneficial supplier relationships

"An organization and its suppliers are interdependent and a mutually beneficial relationship enhances the ability of both to create value."

An organisation must work with its suppliers wherever possible, creating clear and open communications in order to ensure reliable and defect-free delivery of supplies. Also create relationships that balance short-term gains with long-term gains so that both can benefit.

II.2.4. Quality management systems

Taking into account the evolution of the quality concept and the globalization of the markets, any organization that pretends to be competitive and differentiated from others, must resort from new ways for management in order to surpass new challenges.

This new needs promoted the development of systems capable of assuring continual improvement, higher involvement from top management, and the creation of effective and efficient quality management. Thereby, Quality Management Systems (QMS) appeared, in which the term system "conveys the idea that several elementary components, independent, interact and form a coherent whole with a common goal" [7]. The union of quality management to system implies the creation of a concept that can be defined as

“Set of organizational measures capable of transmitting maximum trust that a certain acceptable level of quality is being achieved with minimum cost” [7].

With the implementation of this management system, organizations will define resources and responsibilities for the area of quality management, therefore, making quality management viable, effective and competitive.

The QMS shows, in a clear and effective way, how organizations have to deal with different areas, highlighting:

- Results of data analysis from processes and products measurements;
- Results of customers evaluation;
- Results of internal and external audits;
- Treatment of non-conformities and non-conform product;
- Treatment of internal and external complaints; and
- Initiation, monitor and verification of the corrective, preventive and improvement actions efficiency.

Decision making is one of the most important steps in any process, and the decision of implementing a QMS is a good example. Before any organization decides to implement a QMS, organizations must be sure if they can implement a QMS capable of evolving in the long term. Nevertheless, the implementation of this system does not grant by itself customer satisfaction regarding the organization’s products/services quality. Since this is a dynamic system, product and services continual improvement must be centred at customer needs; organizations that best perceive this philosophy will able to maintain their products updated and will generate return from the investment made in the quality system.

Being one of the natural results of the history of quality evolution and an increasingly complexity on quality assurance, the QMS seek to contribute in the achievement of certain objectives:

- Systematic approach to all activities that can affect quality;
- Highlight preventive activities rather than trust in inspections;
- Provide clear evidence that quality was met.

With these three big objectives in mind, the QMS must be faced by those who are involved, as an aid in problem solving, rather than something that is bureaucratic or difficult.

Making a link between the QMS and the International standard ISO 9001:2008, it is possible to say that this standard is used as the base for developing and implementing a QMS. According to the standard, an organization is not obliged to include all of its products and services in their QMS, or contemplate the processes used to create those products and services. However, an organization must be aware that, in order to implement a QMS, it must define the process to be controlled, how they should be measured and how often, and transform the current process approach in an integrated process approach.

An integrated process approach brings multiple advantages to an organization such as: the involvement of all people, a management based in a system, continual improvement and a better focus on the customer, avoiding miss communication between departments in order to provide a product/service according to the need and desire of the customer.

II.3. ISO 9001:2008

II.3.1. Introduction

The International Organization for Standardization (ISO) is the biggest non-governmental organization that produces and edits international standards. This organization consists of a network of standardization institutes from 162 countries, where each country is represented by a member. The headquarters of ISO is in Geneva, Switzerland. This organization connects the public with the private, since many of the institute members are part of a country governmental structure, or are controlled by the government, while others are entirely private, having been set up by national partnerships of organizations from different sectors. Facing this reality, ISO is consensual when reaching for solutions that meet specific business needs, as well as other areas of society in general.

The ISO 9001 standard belongs to the ISO 9000 series of standards and its purpose is for the certification of Quality Management Systems (QMS). The certification recognizes the efforts of any organization to ensure conformity of their products and/or services, customer satisfaction and continuous improvement and was reviewed in 2008.

The following advantages [10] are almost guaranteed to any organization that obtains the certification:

- Improved internal organizational;
- Increased customer satisfaction;
- Increased employee satisfaction;
- Improved competitive position;
- Increased productivity;
- Cost reduction;
- Increased opportunities in specific markets;
- An internationally recognized QMS;
- Ensure quality to customer and gives confidence to the management;
- Allows it to be audited by a specialized third party entity;

The eight principles of quality management [11], which have been already presented in II.2.3, are the basis of a QMS and underlie the requirements of ISO 9001.

II.3.2. Process model

This International Standard adopts a process approach when developing, implementing and improving the effectiveness of a QMS. This type of model provides a step by step control over the linkage between the individual processes within the system of processes, as well as over their combination and interaction. This means that an organization is capable to identify and manage linked activities by the process inputs and outputs.

The model of a process-based quality QMS is represented in Figure 1, below:

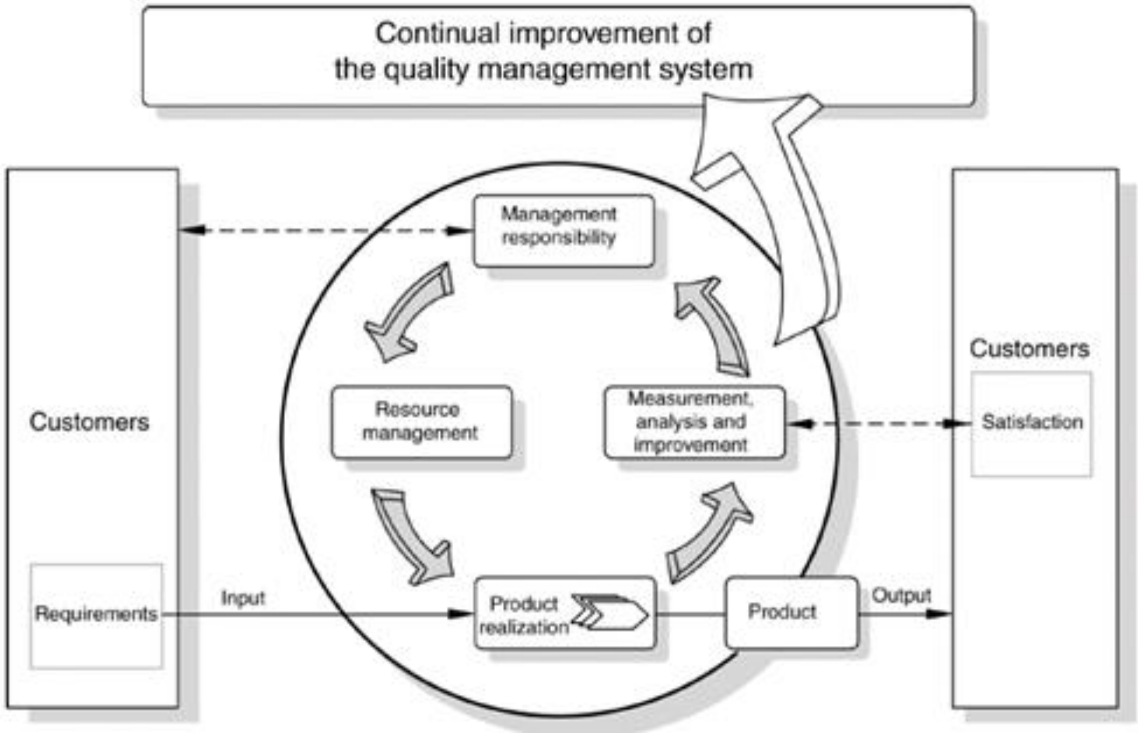

Figure 4 – Model of a process-based quality management system [12].

It is based on a "Plan-Do-Check-Act" methodology, which helps the organizations to create, implement, monitor and measure their own processes in order to obtain results that fall within the organization requirements and continuously improve performance, adopting the respective most appropriate action.

This cycle is intended to make the processes of an organization more clear and agile in any functional area of the given organization, ensuring advantage over competitors and success in its business.

II.3.3. Development and implementation of a QMS

In order to develop and implement a QMS, the management must commit itself in doing so. Initially the management should make a diagnosis of the current state of the organization, as well as an analysis of its strategy and the various motivations that arise around such project.

II.3.3.1 Quality Management System

As outlined in the standard, the organization shall "establish, document, implement and maintain a quality management system and continually improve its effectiveness in accordance with the requirements of this International Standard" [12].

In order to address this requirement, the organization should make a detailed analysis of their work process, transcribing it to paper and adapting it to the Standard requirements. The procedures of the organization should be developed through a more detailed model while the records should be more consistent and accurate, facilitating the interpretation of the documents by their users.

- Documents requirements: The organization should be based on a "documented management system" in order to ensure results and the consistency of the activities and processes. Only the necessary documentation that supports the management of processes should be developed. The standard at this point defines a list of documents that the organization should include in the QMS and gives an increased focus to the quality manual and to the control of documents/records. The table below contains a summary of all the required records by ISO 9001 as well as their corresponding section.

Table 1 – Required records by ISO 9001 [13].

Section	**Required record**
5.6.1	Management review.
6.2.2 e)	Personnel education, training, skills and experience.
7.1 d)	Evidence that the realization processes and resulting product meet requirement.
7.2.2	Results of the review and actions arising from the review.
7.3.2	Design and development inputs.
7.3.4	Results of the reviews and any necessary actions.
7.3.5	Results of the verification and any necessary actions.
7.3.6	Results of validation and any necessary actions.
7.3.7	Results of the review of changes and any necessary actions
7.4.1	Results of evaluations and any necessary actions arising from the evaluation
7.5.2 d)	Processes requirement where the resulting output cannot be verified by subsequent monitoring or measurement.
7.5.3	Unique identification of the product, where traceability is a requirement.
7.5.4	Customer property is lost, damaged or otherwise found to be unsuitable for use.
7.6 a)	The basis used for calibration or verification in cases where no measurement standards traceable to national or international measurement standards exist.
7.6	Validity of the previous measuring results when the equipment is found not to conform to requirements.
7.6	Results of calibration and verification.
8.2.2	Internal audits and their results.
8.2.4	Evidence of conformity with the acceptance criteria as well as the person(s) authorizing release of the product.
8.3	Nature of nonconformities and any subsequent actions taken, including concessions obtained
8.5.2	Results of action taken.
8.5.3	Results of preventive action taken.

II.3.3.2 Management responsibility

From this clause of the standard, the requirements and documents are described in a more detailed manner and adapted to the organization practices. This means, it is noticeable a difference between the management commitment with quality, the level of orientation between the QMS and the customer, the quality policies and objectives of each organization and the level of communication between various levels within the organization.

- Management commitment: The standard states that top management shall demonstrate its commitment to the development and implementation of the QMS, i.e., the organization must show evidence of compliance with the clauses of this normative reference by providing the necessary resources to establish a quality policy and quality objectives as well as communicating to all its members the importance of meeting customer requirements.

- Customer focus: Being one of the principles of quality management, this should be promoted through the involvement of top management in ensuring that the organization defines customer requirements and ensures that the organization is going to meet them in order to enhance customer satisfaction.

- Quality policy: This part only intends the organization to define some key points in its quality policy declaration. The quality policy should be appropriate to the organization, should include a commitment of good QMS practice and it must be communicated and understood throughout the entire organization. It must be continuously reviewed in order to keep appropriate with the organization performance.

- Planning and management review: The point "Planning" of the Standard states that an organization should propose goals to achieve in a given period of time, which must always demonstrate continuous improvement and shall implement goals, actions and methodologies to achieve them. The organization must also ensure that when the QMS is being planned, it is conducted in a way that meets customer's requirements and ensures that the QMS integrity is maintained when changes are planned and implemented in the QMS. The standard also requires a periodic review of the QMS and this is one of the most revealing actions of commitment by the administration, as well as the use of and usefulness of it. This action seeks to ensure that the QMS remains suitable, adequate and effective.

II.3.3.3 Resource management

Compliance with this section ensures two fundamental objectives of a QMS. These objectives are the implementation and maintenance of the QMS, in order to continuously improve its effectiveness and increase customer satisfaction. Resource management goes through the planning, processing, maintenance and training of human resources, infrastructure and working space.

II.3.3.4 Product realization

This section of the standard refers to all processes that deal directly with the design and development of the product. Thus, an organization should ensure proper planning and process development for product realization. This planning should encompass all the activities related to the supply chain of a product.

- Planning of product realization: Ensure that the processes associated with product realization consolidate the quality objectives envisioned by the organization which should include product requirements, identification and good use of resources, verification activities, validation, monitoring, measurement, inspection and testing; and maintenance records showing evidence of its correct organization. The result of this planning describes the resources and activities required to plan and execute the work and check if it meets the requirements and expectations. Customer-related processes: When interacting with the costumer the organization should focus at the methodologies for determining the product requirements, its review and the communication with the client. Regarding the determination of requirements, the organization must take into account the requirements specified by the customer (including delivery activities and after-sales service), the requirements that the customer does not declare but are necessary to the functionality of the product, and the statutory and regulatory requirements related to the product. As for the review, the organization must ensure that product requirements are defined, the contract requirements or orders that are different from the ones agreed are resolved, and assure that the organization has the ability to meet the defined requirements. In the communications field, the organization has to ensure effective mechanisms to communicate with customers at all stages of production and supply of the product;

- Design and development**:** This section deals with the design and development of a product to be supplied. The needs and expectations of the customer are defined based on the expected functionality, performance and necessity and it is up the organization to translate that into a product. Hence, the following main activities are:
    - Planning**:** Regarding the design and development planning the organization shall identify stages, people and/or groups responsible for them, as well as its inputs (requirements, etc...) and outputs (product specifications, etc...) including revisions and verifications and at some points, analysis and decisions;
    - Revision**:** The organization shall review the suitability and effectiveness of the adopted solutions in order to meet the objectives that were previously defined. When the process of review comes to an end the organization must take decisions regarding the future of the project;
    - Verification**:** At this stage, the organization should confirm if the objectives have or not been achieved and elaborate an analysis of the design processes. At the end of this activity there should be, as output, decisions regarding the viability of the process;
    - Validation**:** The last step of this section is the validation, where the client and/or organization examines whether the product meets the previously

defined requirements. Ensure that the product is suitable for its intended use, in real or simulated space. The validation is usually performed by the use of a prototype; and

- Control of design and development changes: Changes in the design and development should be identified, reviewed, verified, validated and finally approved before implementation. The possibility of interaction and exchange between the different components of the product should be made known and the effects of any changes in the design and development should be documented.

- Purchasing: Within this clause, the standard demands three fundamental items, which in turn should take into account some important points**:**
    - Purchasing process: This section requires the organization to establish an adequate control over their purchasing process in order to guarantee that only compliant products are acquired. Therefore it is necessary to evaluate suppliers and then select them based on their ability to supply compliant products. The responsibilities and methodologies of conducting such activities must be carefully controlled, based on tracking activities that are planned, implemented and monitored;
    - Purchasing information: The purchase information should describe the product specification and the required purchasing requirements. The organization shall ensure that the specified purchase requirements are appropriate prior to being communicated to the supplier. All purchased products by an organization must be accompanied with information describing that same product in order to validate the compliance of the product when it is received; and
    - Verification of purchased product: The organization should identify and carry out inspections or other activities to ensure that the product complies with the purchase requirements. The nature of these activities depend on the importance of the purchased product and on the supplier trust level.

- Production and service provision: This point intends to focus in the production of products. The organization shall plan and carry out the production and service provision under controlled conditions, where these conditions are the information of the product characteristics, work instructions, use of proper equipment, monitoring and measurement, implementation of monitoring and measurement activities, and development of methodologies for delivery and post-delivery. The Standard also imposes some important requirements beyond product control and service provision:
    - Identification and traceability: At this point it is demanded that each product has a unique code within the organization, and is correctly identified (verification status, inspection, etc...) anywhere the organization;
    - Customer’s property: Ensure that the organization protects customer's property when this property is made available for incorporation into the organization's product or for use in the organization’s product realization processes. If in any case customer's property is damaged or broken, the organization should report the event to the client and keep records of it; and
    - Product Preservation: In this field, the Standard intends that all shipped and stored products are properly identified, packaged, protected and handled in order to keep their requirements intact and compliant.

- Control of Monitoring and Measuring Equipment (MME): The organization shall ensure that any device used for monitoring or measuring the conformity of the product is able to provide valid results. The calibration of the measuring equipment should only be performed when it is necessary to provide valid results on the compliance of the product when related to the specified requirements. The MME are in some cases software or may include software functionalities, which must also be subject of control, being necessary to validate the software edition in use. It is legitimate to control the software with monitoring or measurement functionalities at least once, if it is assured that there is no change in the controlled edition or neither in the infrastructure where it is operated (hardware). The outcome of each MME calibration or verification should be analyzed by the organization against criteria previously defined and consistent with the tolerances of the product specifications that were controlled by the devices. Records of these analyzes should be kept. Each MME should only be released for use when such analysis demonstrates the suitability of the equipment.

II.3.3.5. Measurement, analysis and improvement

In this section, the organization shall plan and implement the processes for monitoring, measurement, analysis and improvement that are necessary to demonstrate compliance with the product requirements, the QMS and the continuously improvement of its effectiveness.

- Monitoring and measurement:
  - Customer satisfaction: The organization shall monitor information relating the customer perception of how the organization operates. This enables the unfavorable results or trends to trigger corrective action and continuous improvement, while the favorable results can be used to promote further improvement in the product;
  - Internal audit: This section serves to ensure more extensive checks of the effectiveness of processes and procedures at planned intervals. That is, the organization must have (or contract) trained people in audits in order to check the activities and procedures of the organization, without making any kind of performance evaluation to the personnel. According to the Standard, audits should be scheduled, planned and their records kept, which demands the existence of a documented procedure;
  - Monitoring and measurement of processes: The organization must implement evaluation and measurement methodologies for the QMS processes in order to analyse its performance. These methods should include indicators that show the state of aptitude of the processes when

comparing with the previously planned results, in case of negative results the organization must initiate preventive and/or corrective actions; and

  - Monitoring and measurement of product: This requirement is intended to ensure that all product characteristics are monitored and measured (and the results recorded and kept) in their productive cycle, in order to verify that they meet the requirements of the product idealized.

- Control of nonconforming product: The organization shall ensure that actions are taken at all phases of product realization so that the nonconforming product does not follow the normal process. This entails the existence of a documented procedure, in which all the treatment that is given to the non-conforming product is documented in order to prevent it from being sent to the customer;

- Analysis of data: The organization must not only determine, collect and analyze information on the QMS and its processes but also analyze these data in order to identify trends and opportunities for improvement, including preventive actions; and
- Improvement:
  - Continual improvement: The Plan-Do-Check-Act cycle of this standard is highlighted in this section as this indicates organizations the idea of continually improving the effectiveness of their QMS. This continuous improvement must be understood and demonstrated by the organization through its quality policy, its quality objectives, audit results and the analysis of data from the QMS that is implemented throughout the organization;
  - Corrective and preventive action: The standard requires the existence of documented procedures for these two actions in order to ensure that the organization undertakes actions to prevent and/or eliminate the causes of nonconformities and so prevent and/or avoid its occurrence. These actions should be appropriate to the effects of the detected nonconformities or to the effects of the potential problems.

II.3.4. Conclusion

The international standard ISO 9001 is used for the certification of quality management systems and plays a key role in the development of strategies for the organizations that adopt it. This increased awareness of customer importance and increasing business to business competitiveness, has lead organizations to invest in the QMS as a source of competitive advantage and continuous improvement of its products and/or services, and consequently reshaping its image within the markets.

II.4. Portuguese Standard NP 4457:2007

II.4.1. Introduction

The Portuguese Standard NP 4457:2007 aims to define the requirements for an efficient Research, Development and Innovation (RDI) management system, enabling organizations that adopt it to define a RDI policy and achieve their innovation objectives.

The standard is based on a model of innovation, supported by interfaces and interactions between scientific and technological knowledge, knowledge about the organization and its functioning, the market and society in general.

This normative reference helps organizations to improve their performance, with emphasis on their RDI management system, as a mechanism to create knowledge and transform it into economic and social wealth.

This report intends to describe, in a brief way, the various stages and requirements to implement a RDI management system.

II.4.2. Process model

The Standard uses the interaction model chain as its model of reference, which is a model of innovation for the economy of knowledge. The design of the model is based on Kline and Rosenberg's diagram of interactions chain, known as the chain-linked model [14] which is an attempt to describe the complexities in the innovation process. The interaction model chain accommodates the concepts of the 3rd edition of the Oslo Manual [15] which contains guidelines for collecting and using data on industrial innovation and thereby, the model has particular emphasis on the concept of innovation.

The Oslo manual states that innovation is "the implementation/commercialisation of a new or improved product/service, or process, or a new marketing method, or a new Organizational method regarding business practices, on the organization of the working space or on the external relations" [15]. Focusing on the products line, it is considered a technological product innovation when a new product/service with improved performance characteristics is implemented. This includes significant improvements like technical specifications, components and materials, embedded software or other functional characteristics, and follows the same philosophy for process innovation.

The concept of research is associated with the discovery of new knowledge or to a better understanding in the scientific and technological scope.

The NP 4456 [16], Standard that establishes the terminology and definitions that are used within the Standards, identifies two strands of research:

- Fundamental or basic research;
- Applied research.

Fundamental research relates to activities that involve the generation of new knowledge about physical, biological and social phenomena while applied research is directed at solving specific technical problems.

As for development, it is defined as a set of systematic activities carried out based on the knowledge created by research, with the goal of creating new or significantly improved materials, products or services, marketing or organizational innovations.

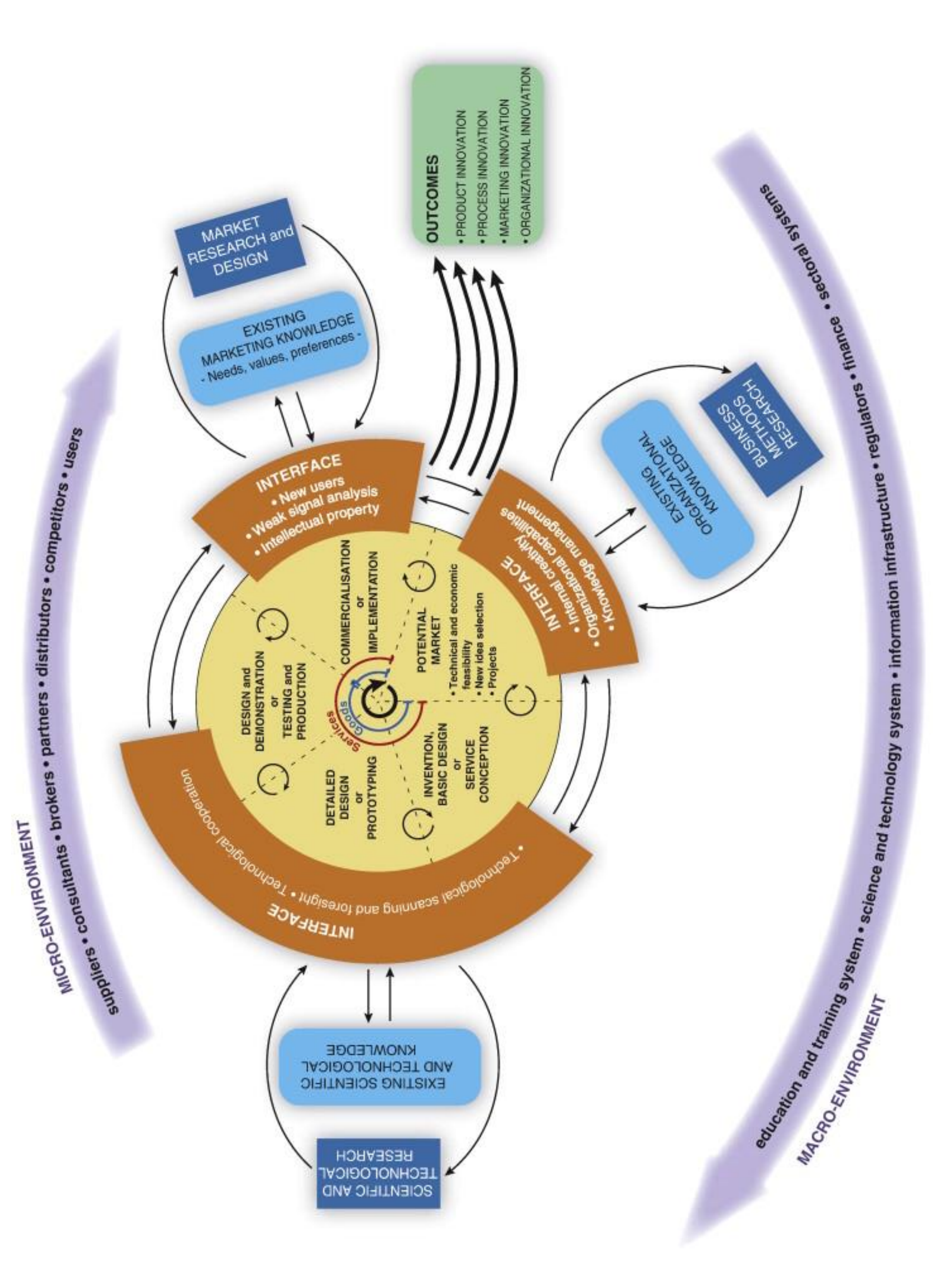

Figure 5 – Interaction model chain [17].

According to the interaction model chain above, innovation is the result of interactions between skills and scientific-technological knowledge, market and organization that exist in the organization and the skills and knowledge resulting from the Organization's external, macro and micro environments (education and training system, science and technology system, suppliers, customers, partners, etc...).

The RDI management system follows a Plan-Do-Check-Act approach, focusing on continuous improvement and outlining the requirements for the different phases of this cycle. The standard was aligned with other ISO management system Standards.

Advantages of implementing a RDI management system

The advantages obtained with the implementation of a RDI management system are observed in several areas and according to APCER, a Portuguese certification company, it allows organizations to [18]:

- Systematize their RDI activities to avail the internal "know-how";
- Establish goals and objectives that contribute to the control of resources associated with activities;
- Plan, organize and monitor RDI units;
- Improve the organizational image and competitiveness in relation to other organizations of the sector at national and international levels;
- Follow technological development in order to anticipate the market and identify opportunities for improvement;
- Integrate the management of RDI with other management systems implemented in the company;
- Establish the interaction of RDI with other departments and divisions of the organization;
- Obtain patented technology that allows its subsequent license to sale;
- Demonstrate to the public administration and all the organisms that evaluate RDI projects for possible funding, transparency of this activity in the organization;

- Monitor, identify opportunities for improvement and implement corrective actions, in accordance with the results obtained in their research, development and innovation activities.

II.4.3. Implementation of a RDI management system

II.4.3.1 General

At this point the Standard summarizes the main requirements for the implementation of a RDI management system. Briefly the organization must determine how it will meet the requirements, define the scope of RDI activities and document the activities associated with the management system RDI.

II.4.3.2 Management responsibility

The NP 4457 Standard assigns to top management the responsibility of ensuring that the RDI management system is established, implemented and maintained in accordance with the requirements of the Standard. The RDI policy is one of the many important tasks of top management and should be clearly conveyed to the whole organization in order to provide effective management.

II.4.3.3 Planning of the RDI

- Management of interfaces and the production of knowledge: At this point the Standard addresses the organization’s need to analyze their micro-environment and macro-environment in order to ensure the best interaction with the external environment, which can be exchanges of information, threats and opportunities that may be either technological or new customers. It also states the relevance of the activities performed within the organization that encourage creativity and the need to establish procedures to manage information;
- Management of ideas and evaluation of opportunities: The ideas are important to the appearance of RDI projects. This point states the necessity to establish a methodology for capturing, analyzing, evaluating and pre-selection of ideas as well as the respective evaluation of the potential market, technical and economic study and risk analysis;

- There are various ways of capturing ideas, some can spontaneously be born inside the organization, or after the analysis of data, the evaluation of results or even during a brainstorming reunion. It is up to the organization to elaborate an adequate method to register this ideas; and
- Planning and implementation of RDI projects: If an idea/opportunity is validated for implementation, the organization should start planning a project for it. The Standard requires the establishment of a project plan that includes provisions for the subsequent phases, as well as the content that it should refer. Citing the standard [17], the project plan should refer:
    - Description of the project and identification of the problem to solve, improvement possibilities, competitive advantage or expected benefits;
    - Planning of activities, including verification and validation activities;
    - Identification of the team, resources required and timelines;
    - Expected results, including milestones;
    - Method to control changes;
    - Documentation of the provisions relating to intellectual property.

II.4.4. Implementation and operation

II.4.4.1 Activities of RDI management

The Standard states that the organization has to identify the activities in the various areas of management that are necessary for the RDI process and list these areas. Relatively to these identified activities, the organization must ensure their proper execution, documentation and registration.

II.4.4.2 Competence, training and awareness

The organization has to ensure that personnel who carries out RDI activities have the necessary skills to do so and should, when necessary, provide training.

II.4.4.3 Communication, documentation and control of documents

The Standard restates that the internal and external communication must be efficient and clear, also enumerating the topics that need to be included in the documentation content of the RDI management system. For these documents, the organization shall establish, implement and maintain procedures in order to approve, revise and update these documents. It should also possess a list of all existing documents, internal and external, in order to provide an effective system to identify and control these documents. The same applies to records.

II.4.5. Evaluations of results and improvement

II.4.5.1 Evaluations of results

The evaluation of results includes the analysis of the values from the objectives that were defined by the RDI policy and the results from RDI projects.

The results reflect the real value of the organization's RDI system performance so, whenever deviations are detected, there must be an analysis and corrective actions must be implemented.

Regarding the procedures for documenting and assessing the RDI results, the standard lists several points that the organization has to ensure.

II.4.5.2 Internal audits

Audits are a management tool since they allow to evaluate the compliance status of the RDI management system with the standard requirements and with the methodologies defined by the organization.

It is up to the organization to define a program of audits and also ensure that actions are taken to eliminate non-compliance cases within the agreed timeframe.

II.4.5.3 Improvement

The NP 4457 states that "the organization shall continually improve the effectiveness of the of the RDI management system." Some tools for improvement are listed below:

- Use of RDI policy;
- Evaluations of results;
- Results of audits; and

- Internal and external communication.

II.4.6. Conclusion

The implementation of a RDI management system should be suitable to the organization, and it is expected to contribute for a better efficiency regarding the management of RDI activities and consequently to better results.

It is essential to be strict in interpreting the Research, Development and Innovation concepts when implementing a RDI management system. One should also avoid a bureaucratic management system and always focus on results.

It is important that organizations have the ability to adapt and are oriented to change, experimentation, share and try to learn from experience tolerating mistakes and taking risks consciously. These organizations are generally organizations where there is a culture of innovation and most assuredly the implementation of a RDI management system will be a success.

Innovation is a differentiating factor and its management assures several key aspects of the strategy of RDI. Ensures that an organization is open and does not waste innovative ideas, being able to prioritize, distinguish opportunities and carrying out the development of innovative projects. The organization should be focused on results so that the whole process of RDI is able generate products at the appropriate time.

II.5. PRINCE2

II.5.1. Introduction

PRINCE is a structured method for effective project management. It was established in 1989 by the Central Computer and Telecommunications Agency and was originally based on PROMPT, a project management method created by Simpact Systems in 1975. This method was developed for the UK government and has been used since then. Many organizations from around the world use PRINCE as a standard as well [19]. "PRINCE is a way of delivering projects and it is based on a set of processes, components and techniques that can be followed by any organization" [20]. It is also used for R&D projects, product development projects, marketing projects, business transformation projects and many more.

PRINCE states the following features [20]:

- Its focus on business justification;
- A defined organization structure for the project management team;
- Its product-based planning approach;
- Its emphasis on dividing the project into manageable and controllable stages;
- Its flexibility to be applied at a level appropriate to the project.

Regarding its scope PRINCE covers the entire project life cycle, i.e. the path and sequence through the various activities to produce the final product, covering mostly the management of the project and the resources involved in it. However it does not cover all of the relevant subjects of project management, which are the specialist techniques involved in the creation of the products and should be covered by other methods. This does not mean PRINCE ignores them, actually it must interface with them to gather information that is required for the covered areas in project management. The project management techniques and tools are not always the same to every project and they vary depending on its type and corporate environment.

The figure below shows where PRINCE fits into a business and project environment.

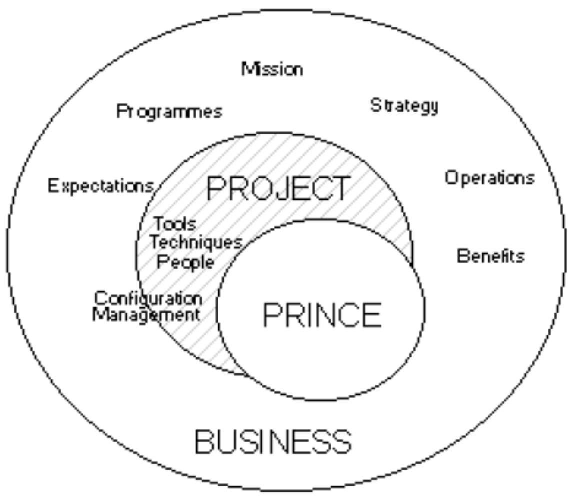

Figure 6 – The PRINCE relationship with Projects and Business [19].

It is easy to understand that PRINCE is mainly centered on the project, but projects also require some pre-preparation in order to start in an organized and controlled manner, PRINCE covers that as well.

II.5.1.1 Project management

Project Management deals with planning, delegating, monitoring and controlling the project. The role of the project manager is to achieve project objectives within the targets set for time, cost, quality, scope, benefits and risk. Project failures are all too common and the reasons are wide and varied. The manual states some common causes [21]:

- Insufficient product definitions at the start, resulting in the wrong product being developed;
- Lack of communication, which may cause a black cloud over the project;
- Poor estimation of duration and costs, leading to projects taking more time and costing more money than expected;
- Lack of control over progress so that projects do not reveal their exact status until too late.

Without a project management method, there is a high probability that those who are involved on a project will have different ideas of how things should be organized and when certain aspects of the project will be completed. The responsibility, authority and accountability for will also be unclear which creates space for confusion around the project. The success rate of projects without a management method is very low and by success it is meant that the project is completed on time and within acceptable cost.

Regarding PRINCE, it guides the project through a controlled, managed, visible set of activities to avoid the problems identified above in order achieve the desired results. The manual sates some principles of good project management [19]:

- A project is a finite process with a definite start and end;
- Projects always need to be managed in order to be successful;
- For genuine commitment to the project, all parties must be clear about why the project is needed, what it is intended to achieve, how the outcome is to be achieved, and what their responsibilities are in that achievement.

II.5.1.2 Benefits of using PRINCE

The manual states that with the use of PRINCE the organization has, as benefits, a controlled management of changes in terms of investment and return; active involvement of the users and stakeholders throughout the lifecycle of the project, which ensures that the products meet the business, functional, environmental, service and management requirements. The methodology features an approach that distinguishes project management from product development that it can be applied in the elaboration of any project in any industry. Project managers using PRINCE are able to use a defined structure for delegation, authority and communication and divide the project into stages for more accurate planning [19]. Thus, all risks will be reviewed and analyzed and there will be a natural systematic for risk management.

From the manual, PRINCE provides projects with [19]:

- A controlled and organised start, middle and end;
- Regular reviews of progress against plan an against the Business Case;
- Flexible decision points;
- Automatic management control of any deviations from the plan;
- The involvement of management and stakeholders at the right time and place during the project;
- Good communication channels between the project, project management, and the rest of the organisation.

II.5.1.3 Overview of PRINCE

PRINCE is a process-based approach to project management where the processes define the activities to be carried out during the project. In addition, PRINCE describes a number of components that are applied within the appropriate activities. Figure 7 shows the components positioned around the central process model.

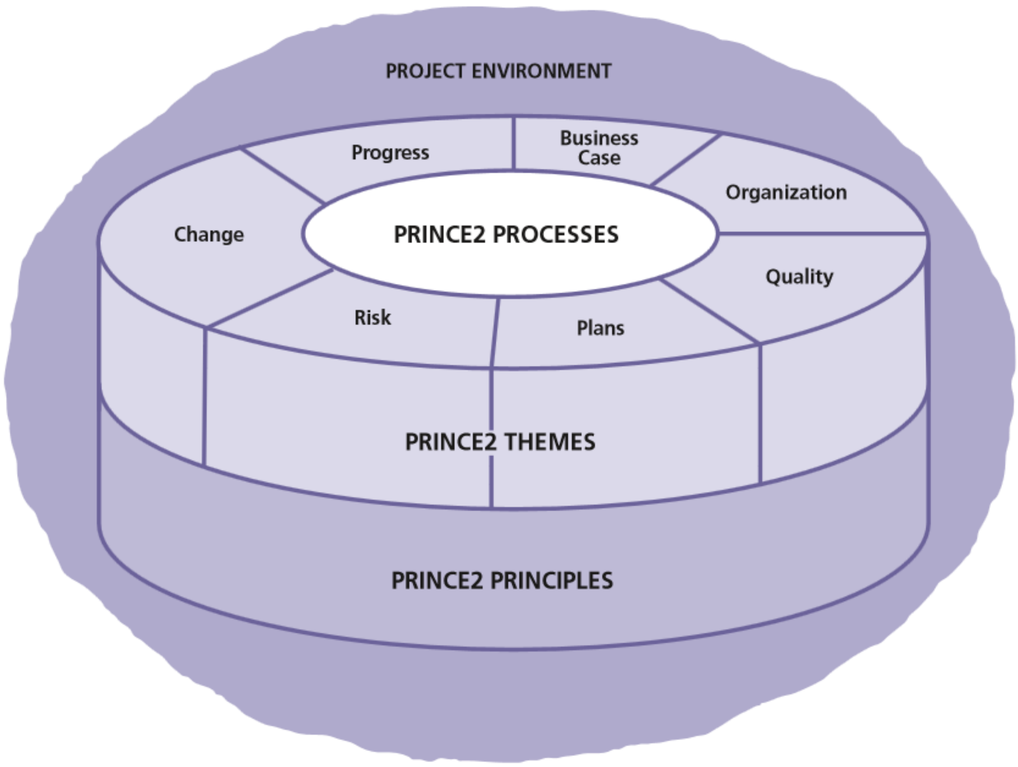

Figure 7 – Structure of PRINCE [22].

The above structure consists of seven distinctive project stages containing forty unique sub-processes, covering all the activities that must be executed throughout the project life-cycle.

Because PRINCE adopts a process-based model, the various project management products are the inputs and outputs of each process. Each product is identified, defined and its delivery is controlled which enables the project to state the standard of quality to which each product must conform. Another technique that PRINCE describes is quality review and it is where the testing of the product comes in, which is an attempt to prove that the products are meeting their required quality standard [19].

Projects have to deal with change quite often and that's something that is not easy to predict. PRINCE takes risk into consideration during project management by incorporating a management of risk into the processes.

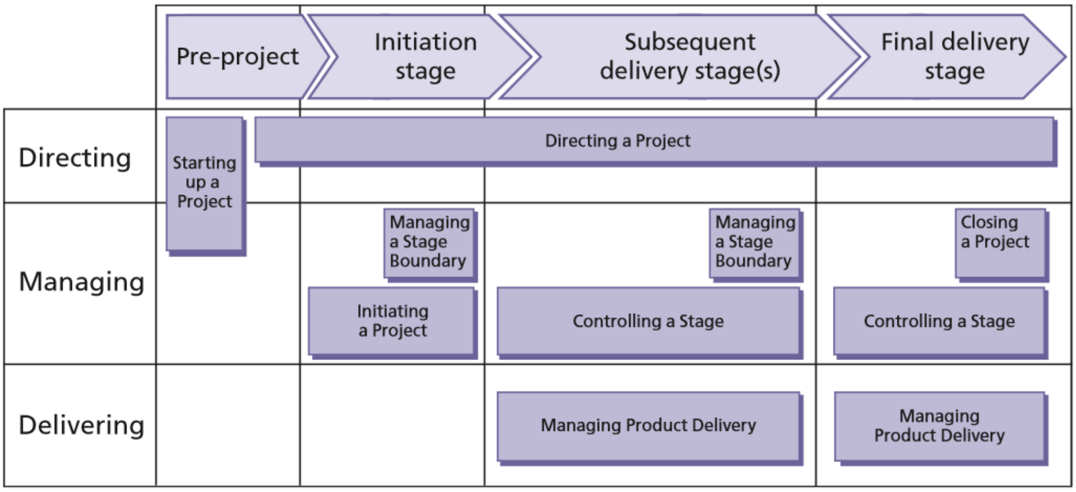

Figure 8 – The PRINCE Process model [22].

By looking at the PRINCE process model, it is possible to understand that the organization's program management is divided into a distinct set of stages. Each Stage has a defined set of activities with its inputs and outputs, which in the end shall result in a product. A Stage is considered completed when the agreed product is completed. Planning is a common process used by many of these processes and can only be done to a level of detail that is manageable and foreseeable. PRINCE divides the project into a number of management stages so the control over projects is easier and enables this control to be varied according to the business priority, risk and complexity involved.

PRINCE defines an Initiation stage that covers the planning and definition of the project in order to allow the management to review it before making any commitment to later stage(s) and their associated resources and costs. The key deliverable from this stage includes an overall Project Plan and defines baselines for the six project performance targets of time, cost, quality, scope, risk and benefits [23].

The PRINCE manual states that stage boundaries need to be appropriate to the particular project and gives some examples of how they may be chosen. They may be chosen according to [19]:

- The sequence of delivery of the products;
- The grouping of products into self-consistent sets;
- The natural decision points for feedback and review.

These activities are a must for any organization that intends any kind of quality certification and are mentioned in ISO 9001, an International Standard that sets the requirements for a quality management system.

II.5.2 Themes of PRINCE2

II.5.2.1 Business Case

The business case presents the mix of information used to judge whether the project is (and remains) desirable, viable and achievable, and therefore worthwhile investing in.

PRINCE defines a Business Case in order to keep the business benefits of the project. Throughout a project, the Business Case is reviewed and progress is measured against the defined benefits. Any deviations from the original Business Case must be controlled. The specification of products can also change during the project and need to be controlled as well in order to maintain the organization's objectives throughout the project.

II.5.2.2 Organization

"One of the principles of PRINCE is that all projects must have a defined organizational structure to unite the various parties in the common aims of the project and to enable effective project governance and decision making" [23] and it is illustrated in Figure 9.

The PRINCE project management structure is based on a Customer/Supplier environment, i.e. there is a Customer who pays the organization to create something specified by him and a Supplier who will provide the resources and skills to create that something.

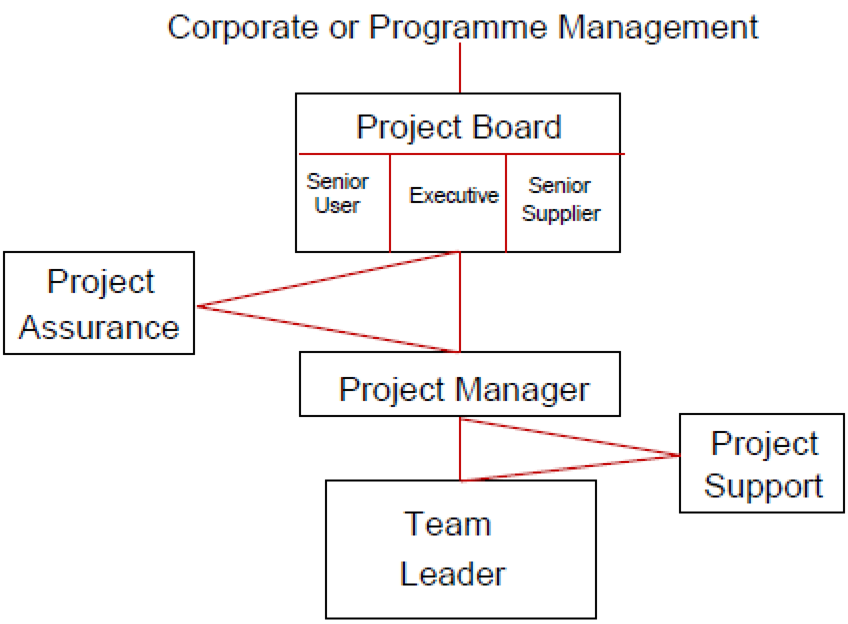

Figure 9 – Project Management structure [19].

This theme describes the roles and responsibilities in the project management team required to manage the project effectively. These roles can be allocated, shared, divided or combined according to the project's needs and the responsibilities for a role can also be moved to another role or delegated, but they should not be dropped.

The Project Manager or Project Board roles are exceptions and cannot be shared or delegated if they are to be undertaken effectively.

II.5.*2.3* Quality

PRINCE treats quality by focusing on products right from the start. This requires systematic activities to identify all the project's products, then define them in product descriptions and finally implement and track the quality methods employed throughout the project.

The first two activities are covered by Quality Planning and the last one is covered by Quality Control and Quality Assurance.

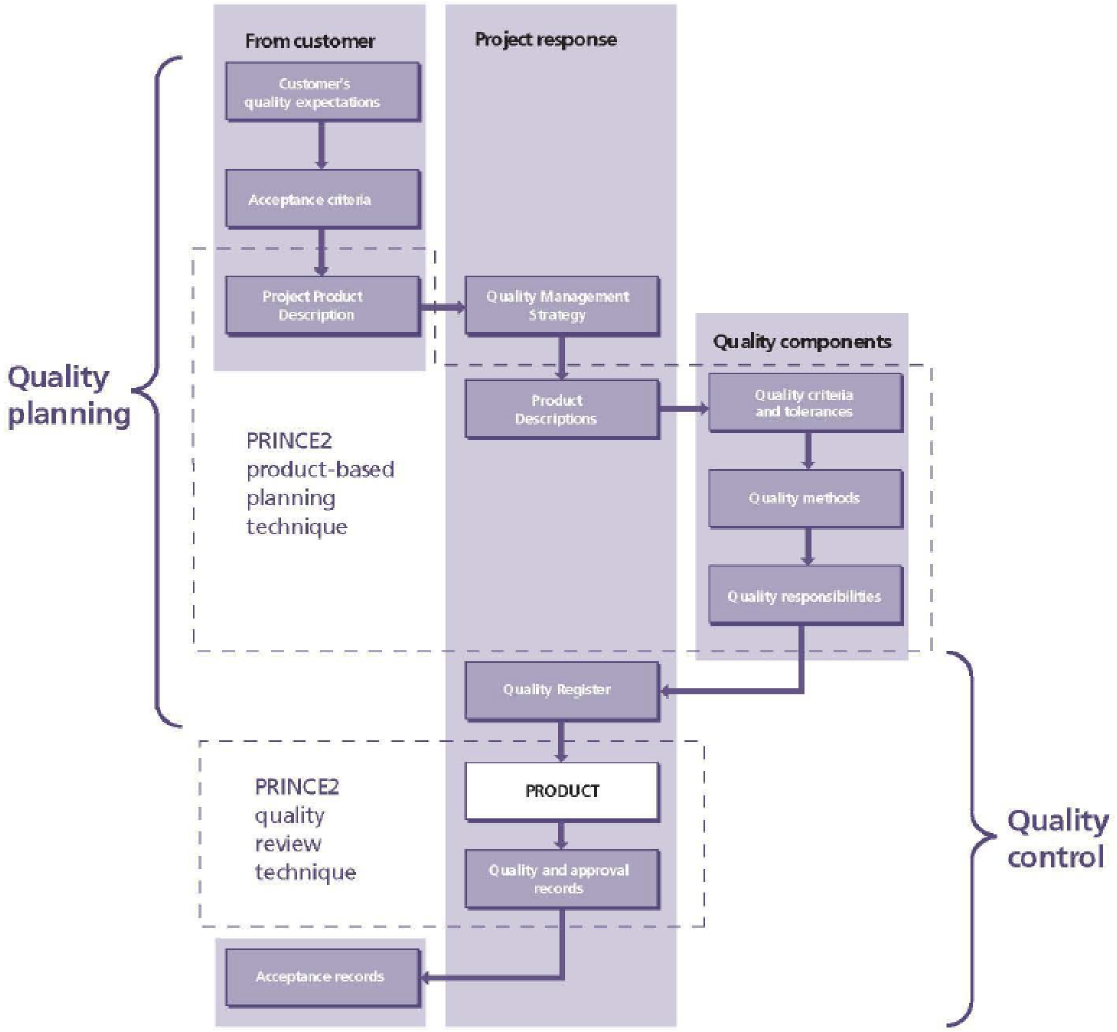

Figure 10 – Quality audit trail [23].

The PRINCE approach to quality can be summarized by the quality audit trail in figure 10. This theme provides an explanation for the outline development so that the quality attributes of the products are understood by all participants and then ensure that these requirements are subsequently delivered.

II.5.2.4 Plans

A plan is a document that describes how, when and by whom a specific target or set of targets is to be achieved. It is a design of how identified targets for deliverables, timescales, costs and quality can be met [23].

This theme complements the Quality theme by describing the steps required to develop plans and the PRINCE techniques that should be applied. In PRINCE projects proceed on the basis of a series of approved plans, which are matched to the needs of the personnel at the various levels of the organization. They are the focus communication and control throughout the project.

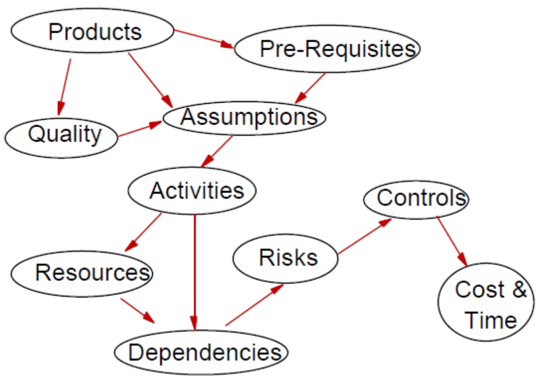

Figure 11 – The Components of a Plan [19].

Figure 11 shows the philosophy behind producing plans in PRINCE, starting from a list of the products to be produced and then defining the activities, dependencies and resources to deliver those products. Risks are then considered, followed by the addition of control points.

II.5.2.5 Risk

Projects typically entail more risk than stable operational activity which makes risk a major factor to be considered during the management of a project. Risk is defined as "the chance of exposure to the adverse consequences of future events".

This theme addresses how project management manages the uncertainties in its plans and in the wider project environment.

PRINCE's approach to the management of risk is based on OGC's (Office of Government Commerce) publication *Management* of *Risk: Guidance for Practitioners* which is based on a number of risk management principles. Some of the principles that fit the project context are [23]:

- Understand the project's context;
- Establish clear project objectives;
- Develop the project risk management approach;
- Define clear roles and responsibilities;
- Monitor for early warning indicators;
- Establish a review cycle and look for continual improvement.

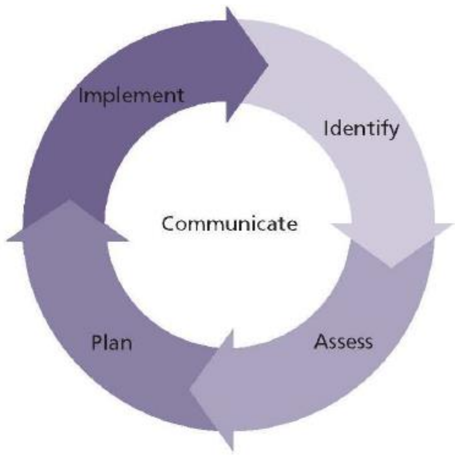

Figure 12 – The risk management procedure [23].

The Project Manager is responsible for ensuring that risks are identified, recorded and regularly reviewed. In order to manage risks in a disciplined manner, PRINCE define two disciplines:

- Risk analysis, which involves the identification and definition of risks, plus the evaluation of impact and consequent action;
- Risk management, which covers the activities involved in the planning, monitoring and controlling of actions which will address the threats and problems identified.

II.5.2.6 Change

Changes to specification or scope can easily ruin any project unless they are controlled. This theme describes how the project management assesses and acts upon issues that can doom the project.

The project's controls for issues, changes and configuration management are defined and established when the project is initiated and then reviewed and (if necessary) updated towards the end of each management stage.

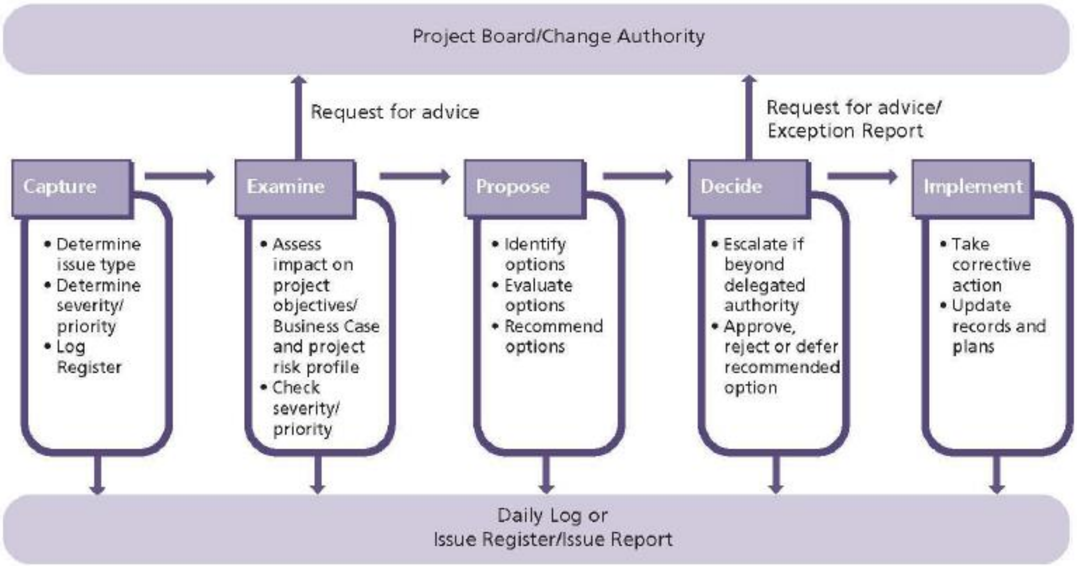

Figure 13 – Issue and change control procedure [23].

Figure 13 shows a common approach to dealing with requests for change, off-specifications and problems/concerns, which PRINCE adopts.

II.5.2.7 Progress

An organization must be able to "measure" the progress regarding its own planned objectives. This theme aims precisely in establishing mechanisms to monitor and compare actual achievements against those planned, in order to have ways to forecast the project objectives and the project's continued viability.

As stated above progress control involves measuring actual progress against the performance targets of time, cost, quality, scope, benefits and risk, and then using this information to make decisions and to take actions as required. PRINCE2 provides progress control through [23]:

- Delegating authority from one level of management to the level below it;
- Dividing the project into management stages and authorizing the project one stage at a time;
- Time-driven and event-driven progress reporting and reviews;
- Raising exceptions.

The project's controls should be documented in the Project initiation documentation.

II.5.3 Conclusion

PRINCE is a widely used project management method tailored to suit the project's size, environment, complexity, importance, capability and risk which some organizations adopt to run their projects and it is considered a standard for project management.

The PRINCE projects have continued business justification and are planned, monitored and controlled on a stage-by-stage basis focusing on the definition and delivery of products, in particular their quality requirements.

While PRINCE is quite strict when defining roles and responsibilities for projects, it allows some tolerances for each project objective, as long as they are controlled. This enables project teams to learn from previous experience.

II.6. ISO 21500

II.6.1 Introduction

ISO 21500 is an International standard that provides guidance on the concepts and processes of project management. In 2006, the United Kingdom requested a new standard for project management aiming for the upcoming Olympic Games in 2012, as their current standard was considered 'antique' and required revision [24]. Various project management associations were involved in the creation of the standard, which is now known as ISO 21500, and it is the first guidance on project management that is accepted and recognized by most international organizations as it provides a 'common language' in project environments.

There is no certification for ISO 21500 because it was written as a guideline and, therefore, there are no requirements yet. However, the standard is a reference for other project management standards, methods, and best practices, such as PRINCE2, Project Management Institute (PMI)'s guide for project management book (PMBOK), Agile and the International project management association Competence Baseline (ICB).

It is expected that, by bringing various project management practices together, organizations attain an increase in theirs project's performance in terms of costs, delivery times, and customer satisfaction. ISO 21500 can also be a supplement of the ISO 9001, thereby, it can be used as a reference in an audit.

II.6.2 Project management terms and concepts

In order to provide a 'common language', the standard has a clause named 'Terms and definitions' in which defines sixteen project management terms. The objective is to improve communication within the organization and also between organizations. Terms that are already defined in the standard lists of ISO definitions are not listed in the clause.

In the next clause of the standard, 'Project management concepts', it is described the most important concepts during the execution of most projects, providing some context about how projects should be performed.

These concepts are:

- Project;
- Project management;
- Organizational strategy and projects;
- Project environment;
- Project governance;
- Projects and operations;

- Stakeholders and project organizations;
- Competences of Project personnel;
- Project life cycle;
- Project constraints; and
- Relationship between project management and concepts and process;

For a better understanding of how project management concepts relate to each other, the figure below provides an overview of the relationships between these concepts.

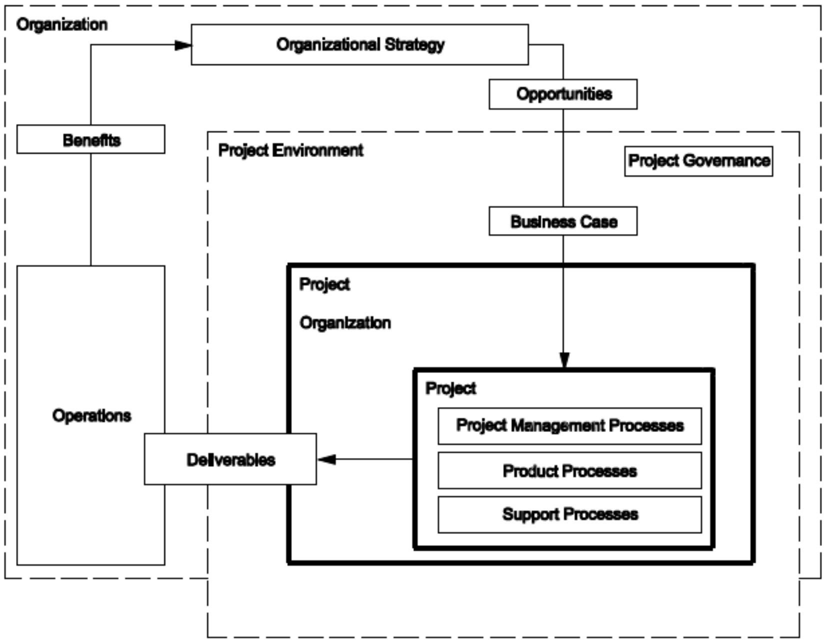

Figure 14 – Overview of project management concepts and their relationships [25].

The arrows represent the logical flow, connecting the concepts. Analysing the figure 14, it is possible to notice that strategies established by the organization may guide the development of opportunities that, once evaluated and selected, may lead to the creation of a business case. In order to accomplish these opportunities, projects are created to transform these opportunities into benefits. ISO 21500 [25] defines project as "a unique set of processes consisting of coordinated and controlled activities with start and finish dates, undertaken to achieve an objective." This objective will lead to the creation of deliverables that must meet specific requirements set by internal or external entities, and require regular supervision by the project management team. These deliverables are normally used by operations to realize benefits, benefits that can be the input to realize and further develop the organizational strategy.

II.6.3 Project management processes

The last clause of the standard is labelled as 'Project management processes' as it identifies some projects management processes that are recommended to use during a project and/or during project phases. Like other ISO standards, the identified processes are generic, and thus, they are appropriate for use by any organization/entity in any project.

Project management is defined in ISO 21500 [25] as "the application of methods, tools, techniques and competencies to a project. Project management includes the integration of the various phases of the project life. Project management is accomplished through processes." This requires coordination within the project management and also requires processes to be well connected with each other. The processes are defined in terms of the purposes they serve, the relationships among other processes, the interactions within the processes, and the primary inputs and outputs associated with each process.

The standard defines five independent process groups: Initiating; planning; implementing; controlling; and closing. Figure 2 shows the interactions between the process groups.

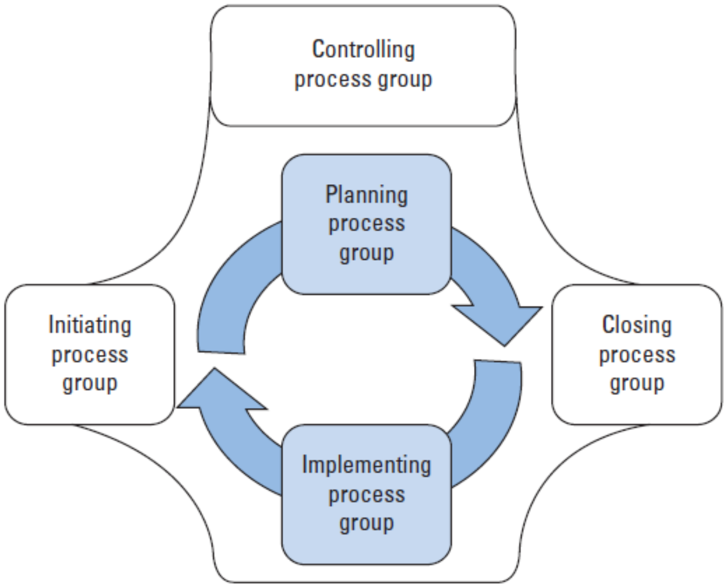

Figure 15 – Process groups interaction [24].

The Deming Circle (Plan-do-act-check) for continuous improvement can be recognized in the figure above, as it influence in the interaction between process groups is quite clear. Some processes may interact in ways that are not illustrated above. The processes are linked with inputs/outputs and, according to the standard, they are mostly project management documents. Nevertheless, all processes might not always be needed on all projects, as well all interactions between them.

I.6.3.1 Subject groups

ISO 21500 defines 10 project management themes, called subject groups: Integration; Stakeholders; Scope; Resource; Time; Cost; Risk; Quality; Procurement; Communication.

Each subject group consists of processes that can be applied in any project phase or project. There is a total of 39 processes divided among subject groups and ISO 21500 defines these processes in terms of purpose, description and primary inputs/outputs.

For an overview of the project management processes in ISO 21500, figure 3 summarises the processes cross-referenced to Process groups and Subject Groups, without representing a chronological order.

Table 2– Project management processes cross-referenced to Process Groups and Subject Groups [25].

Subject groups	Process groups				
	Initiating	Planning	Implementing	Controlling	Closing
integration	Develop project charter	Develop project plans	Direct Project Work	Control project work Control changes	Close project Phase of project Collect lessons learned
Stakeholder	Identify stakeholders		Manage stakeholders		
Scope		Define scope Create work breakdown structure Define activities		Control scope	
Resource	Establish project team	Estimate resources Define project organization	Develop project team	Control resources Manage project team	
Time		Sequence activities Estimate activity duration Develop schedule		Control schedule	
Cost		Estimate costs Develop budget		Control costs	
Risk		Identify risks Assess risks	Treat risks	Control risks	
Quality		Plan quality	Perform quality assurance	Perform quality control	
Procurement		Plan procurement	Select suppliers	Administer contracts	
Communication		Plan communications	Distribute information	Manage communications	

I.6.3.2 Processes

ISO 21500 presents process by process, showing the most common inputs and outputs of them. The standard warns that not every interaction between processes is shown and detailed in the document because in practice, some interactions may overlap with each other. Furthermore, the standard states that some "project-related processes may be accomplished external to the project's boundaries" [26] and recognizes that there is more than on way to manage a project, specifying the determining factors such as risk, size, timeframe, and more.

II.6.4 Conclusion

ISO 21500 is an International standard that provides guidance on the concepts and processes of project management. The standard provides a common language in project management as it references other project management standards. It can be used as a link between different project management, business processes and as a common reference between different methods, practices and models. This standard aims to increase project's performance in various terms such as costs, delivery times, and customer satisfaction.

II.7. PRINCE2 and the ISO 21500

This section compares PRINCE2 with the International standard, ISO 21500, and show where PRINCE2 meets the needs of ISO 21500. The International standard sets the overall framework and creates a 'common language' for project management in an organization while PRINCE2 is a methodology, defining how project management should be applied effectively in specific situations, always from the point of view of the customer.

These two standards don´t share the same structures and scope so it is not always possible to make direct comparisons. Nevertheless, they do share many project management principles, practices and activities and even share the same concept of phases/stages in terms of project lifecycle, outlining where each phase/stage of the project starts and ends. PRINCE2 does not include the procurement sections and the project manager's skills found in ISO 21500. The reason is because PRINCE2 believes that other standards have already developed that section and, therefore, there is nothing new to contribute.

Because PRINCE2 is written as a methodology, it has details of techniques and approaches which are not in the international standard. Areas such as project lifecycle management, business cases, exception management or health checks [26] are an example.

In the end, PRINCE2 meets the requirements of ISO21500 although some elements must be covered by other organizational processes such as document management, procurement and skills and competencies.

III. Development methodologies

Over the last few years, embedded software has grown rapidly both in scope and variety. With that, development tasks noticed an increase in complexity and organizations struggle to understand the best approach for them. This dissertation aims to provide solutions for the development process for embedded software that comply with International and National standards, improve current condition and promote efficient development of embedded software. It is expected that these solutions prove to be helpful among organization's projects, individual tasks, and especially within the development process of embedded software.

The development of software requires various types of organized activities in order to become a product. These set of activities will become part of the software development process. Various frameworks in software development process are already considered solid, such as ISO/IEC1207, which "establishes a common framework for software life cycle processes" and is also applicable for the software portion of firmware. However, most standards have been developed to be generic and consequently, organizations find it difficult to follow in case of embedded software development. Anyway, the development of embedded software must comply, in some way, with traditional development processes, since embedded software is still one type of software.

III.1. Revision of current methodologies

III.1.1. Exatronic's Projects Manual

The Projects Manual (PM) is a guide created by the organization for project management and defines procedures and activities for Conceção&Desenvolvimento (C&D) projects in order to give them a common structure. The PM's life cycle starts with the specification tasks and conception of the product and ends with validation tests of the complete product that were specified by the organization with the customer, and/or vice-versa.

Exatronic is currently certified with ISO 9001 and NP 4457. These standards are established as mandatory conventions and practices intended to be employed and enforced as stated, to ensure that the activities pre-defined within the scope of the objectives of each standard are performed correctly as documented, and the organization is periodically subject to audits with the means for checking whether the organization is fully compliant or not.

The objective of this section was to study the PM in order to verify if it covers all requisites imposed by the ISO 9001 standard, in order to reduce the organization's non-conformities and also improve the quality and productivity of the C&D work.

The table 1 from the ISO 9001:2008 section contains a summary of all the required records by ISO 9001 as well as their corresponding section. Sometimes it is required to keep other records that are not specified by the standard in order to be in accordance with it. It is up to the PM to make reference of all necessary records that are directly and indirectly required by C&D.

The analysis of the mesh between the PM and the standard was performed by following, sequentially, the clauses of the standard in order to guarantee that all relevant points were addressed.

The following clauses of the standard have been analysed:

- 5 – Management responsibility (only 5.6);
- 7 – Product realization;
- 8 – Measurement, analysis and improvement (Except 8.1).

The points that were excluded are 6.2.2, because it relates to human resources management, and point 8.1 which serves as an introduction and sets the main purpose of the specific clause and does not contain anything specific to the PM.

III.1.2. PM Analyses/Proposals

Clause 5 – Management responsibility

- Management review (5.6): The PM should be reviewed by top management in order to assure it remains appropriate, adequate and effective. This revision must have as basis the data analysis report, which at this moment only includes the *Lessons Learned* document, where the tendencies of the organization must be evaluated. The identified negative tendencies shall become object of analyses and measures must be defined to be implemented in order to reverse these tendencies. This point is not required to mention in the PM, but it is appropriate to define and mention a date for reviewing the PM.

Clause 7 – Product realization

- Planning of product realization (7.1): The PM addresses most part of this point in "Especificação de Requisitos". The PM specifies the quality objectives and product requirements and also the necessity of establishing processes and documents. Also states the necessity to provide specific resources for the product in the "Estudos Técnicos Prévios" document. Activities in this first phase also address the need to evidence that realization processes and the resulting product are in line with the requirements, in the "Ficheiro de Controlo" document.

  The required procedures for testing the final product are defined in the document "Especificação de requisitos" and it is reviewed later at the end of the phase.

  The first phase ends with the verification and validation of the requested requirements. Also requires the customer to approve the documents involved.

  The design processes are not planned in this phase and the viability of the product is performed by the business area of the organization.
- Customer-related processes (7.2): The determination of requirements related to the product is the most important part of the business, thus, the organization should obtain all possible information about the product with the customer;
- Design and development (7.3): Related to this topic, the standard states that the organization must plan these process, defining expected outputs and the stages to

achieve those outputs, including the appropriate control for each stage, the responsibilities and authorities. The phase two of the PM, labelled "Conceção", relates to the design planning and is divided by stages. There is no clear identification about revisions, verifications and validations on this stage. The PM identifies the person responsible for the design activities, as well as the inputs and outputs of each activity. There is no reference to any action related to change control. Changes should be reviewed, verified, validated and approved before being set for implementation[1];

- Purchasing (7.4): The PM does not cover this clause of the standard because the document is only directed for the C&D area. However, the organization must assure a clear communication between the C&D and Commercial area;
- Production and service provision (7.5): For the control and validation of the production and supply processes, the PM does cover most points, directly and indirectly, which is enough for this case. Product identification and traceability is not mentioned by the PM, although the organization practice it; and
- Control of monitoring and measuring equipment (7.6): There is no reference of MME in the PM, but the organization has another internal document that completes this point.

Clause 8 - Measurement, analysis and improvement

- Monitoring and measurement (8.2): Regarding the PDCA cycle, the PM does not define methods for monitoring the product in order to assure if it meets the requisites throughout the project life-cycle;
- Control of nonconforming product (8.3): The PM assigns to the person in charge of the activity to decide the action to be taken when an activity ends without success. Also states the possible results of that decision. It should, however, provide more information about procedures for nonconforming products, that is, product with defects or a product that does not meet the specified requirements;

[1] The last version of the PM, released after the work developed in the organization, addresses this issue by setting specific procedures for change control.

- Data analyses (8.4): The management must assure the transformation of data into appropriate information in order to take decisions, identify tendencies and improvement opportunities. The PM covers this point when it mentions the "Lessons Learned" documents. This document must be used to improve project procedures or even identify procedures that demonstrate capability to provide the expected results and use them in future projects and avoid mistakes already committed; and
- Improvement (8.5): The standard ISO 9001 states "the organization shall continually improve the effectiveness of the quality management system". The PM once again does not address specifically this point, although it is indirectly related with the "Lessons Learned" document, which is used for improving the procedures taken by the organization.

The analysis of the PM is now complete and some nonconformities, mostly omissions of procedures, with ISO 9001 were identified. In order to make the PM a better document regarding the subject matter, a set of proposals are now presented.

Proposal nº1 (Formal correction):

Suggestion of documents, structure and what should be included

- Characteristics of the requested product by the customer:
  - Project objectives;
  - Requirements related to the product:
    - Specified by the customer;
    - Not specified by the customer but required;
    - Statutory and regulatory;
    - Additional required by the organization;
  - Project boundaries and interfaces with the external world; (Add)
  - Specific requirements; (Add)
- Budget:
  - Initial estimative of the development cost;
  - Initial estimative of the final cost of the product;

- Project development schedule;
- Risk analysis;
- Pre technical studies;

A flowchart for each phase of the project for a simple, fast and effective way to demonstrate the PM procedures and the relevant structure for the product realization is highly recommended.

Proposal nº2 (Formal correction):

For the activities “Revisão e especificação detalhada” assure that:

- Product requirements are defined;
- Contract requirements or orders that are different from the ones previously expressed are resolved;
- The organization is capable to meet the defined requirements;

These items must be assured before the verification and validation of the customer’s requirements. A simple record stating the capability by the organization to fulfil these items is enough. The item related to the requirements verification and validation must be recorded as well.

Proposal nº3 (Substantial correction):

Perform a revision to the output documents/records before ending (validation) a stage and keep records of that revision. If problems are found and prevent the realization of the product in accordance with the procedure that was previously planned (requirements, resources, supply and more), it is required a change regarding those procedures. Regardless of the stage where the change is made, a study that evaluates the consequences of that change must be performed regarding all component parts of the product. The customer should be informed about the changes, and if those changes affect the requirements made by the customer, a new agreement should be held. All information about the problem must be kept.

Proposal nº4 (Substantial corrections):

Whenever the C&D is involved in a purchasing process, ensure that all information that C&D provides (material references and others), is correctly received by the commercial area to avoid buying products that were not desired, that is, clear communication is essential on this matter. Also, activities for verification and inspection of the product should be realized when this one is received as to ensure that the purchased product is in accordance with the specified purchase requirements.

Proposal nº5 (Formal correction):

Whoever is in charge for each phase, must verify the processes that are performed within that phase and review if the stipulated practices are being fulfilled. The outputs of each phase must also be reviewed.

The results of applying these methods should be somehow captured in order to monitor if the project management process is effective.

Proposal nº6 (Substantial correction):

The nonconforming product can be detected, in the worst case, after being delivered to the customer and, in any case, it must be addressed in the fastest possible away in order to avoid replications. Therefore, actions should be undertaken to eliminate the nonconformity, starting immediately by investigating the origin of this nonconformity.

As for exemple:

- Severe case: After detecting the source of the nonconformity, and that source is within the product realization process, the manager in charge should immediately stop the production in order to save resources, components, and more. A solution should already be prepared to retake the production where it stopped and study the extent to which the material can be used recoverable or salvageable.
- Less severe case: The source of the nonconformity is detected in a post-production phase, and there is no need to stop production.

When the nonconforming product is addressed, it must undertake re-verification to demonstrate conformity with the requirements. The project manager must be responsible for these actions.

III.1.3. Final conclusion

Project Manual is mostly in accordance with the standard regarding the points related to product realization. The clauses related to MME, suppliers and procedures for verification, monitoring, measuring and inspection for the very phases of the C&D are not clearly stated by the document, or not stated at all.

The clauses that are not related to the activities of C&D were excluded from this analysis.

III.2. Study of new methodologies

III.2.1. Peer Code Review

Code review is a software development technique in which the code is reviewed, line-by-line, by a software developer other than the author with the objective to identify defects or opportunities for improvement in the code. This technique is one of the most effective approaches for identifying bugs, typos etc… early in the code development lifecycle. When code review is integrated into the organization's development processes, it will not only increase effectiveness when finding defects but also bring benefits from the act of reviewing to that organization. These can be educational, auditable and historical benefits but will certainly depend on the technique used to perform code review and how well it is performed. The technique addressed by this document is the tool–assisted technique, a process where tools are used in all aspects of review.

In order to effectively review code, it is fundamental that the reviewers understand the purpose of the code to be developed. Ideally they should be involved in the design phase of the project, whenever it justifies doing so. This will help reviewers find defects and most importantly, where defects in the code could endanger most the success of the project. Other aspects that reviewers must be familiar with are:

- The code, i.e. the language(s) used, the features and issues of that language from a performance perspective;
- The context, i.e. what the product is meant to do, what kind of data is being manipulated or processed, and what would the damage to the product be if that data was compromised;
- The audience, i.e. the final users of the product. A product meant for humans will not work the same from the ones that are for machines;
- The importance, i.e. understand the consequences that might happen to the product if the code cannot perform as intended.

Therefore, talking with the developers and the project leader is an effective way to get started, even a short meeting is just enough for the development team to share some information. If somehow that basic information cannot be obtained in any way, then the reviewers will have to spend additional time examining the code in order to know how the code is supposed to work, which is ineffective.

When performing code review with a specialized tool, there are two types of code review: pre-commit review and post-commit. Pre-commit review is when the code is reviewed and consequently fixed before being sent, "commit", to any type of source control manager. Post-commit review is when reviews take place after the code has been committed to the source control manager. Any fixes that need to be made are then committed again later.

There are some advantages and disadvantages concerning these types of review but still, there is not a consensus between developers on which type of code review is 'the best'. In the end it is up to the organization to choose what is best for the project. Some advantages of pre-commit review [27] are that the quality of the code is met before being committed; ensure that the code committed to source control manager has been reviewed and when other developers use the code, they won't be affected by bugs, or at least this risk is minimized.

On the other hand, pre-commit reviews may delay further work of other developers on the code since they have to wait for a successful review. Also, by a mistake of who commits, the version of the committed code may not be the one that successfully passed the review.

Post-commit reviews advantages are that changes on the code can be tracked by other developers, and that they can continuously work on the code. As a downside, this type of code review can potentially decrease code quality in the source control manager.

Regarding what type of code review is being performed, the development process can stipulate additional tests (manually or automated unit or component tests). Hence, there should be any kind of information about which tests were performed before the code is reviewed. This can give reviewers and other personnel some reliability that the code work, also allows reviewers to suggest additional testing that may have been missed by the author.

The organization should stipulate some "good practices" for code reviewing in order to optimize it. A study [28] conducted by SmartBear, a company specialized in software tools for developing, testing and monitoring software, at Cisco Systems, aimed to develop a theory for best practices to employ for optimal review efficiency and value. The result highlights eleven best practices for code review:

1 – The developers should review fewer than 200-400 Lines of Code (LOC) at a time. It states that beyond that number of lines the ability to find defects decreases drastically.

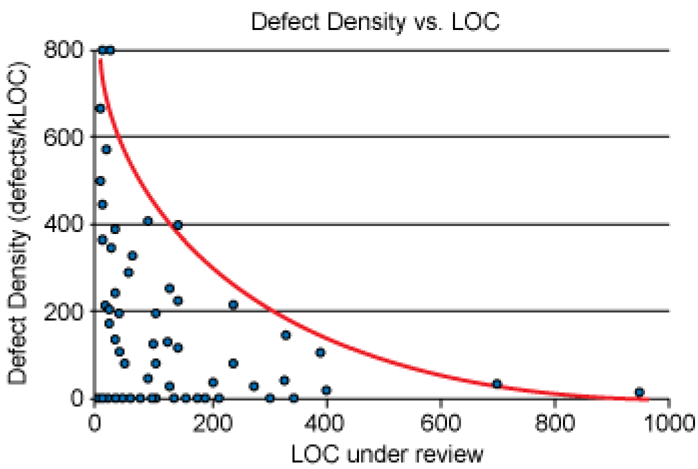

Figure 16 – Defects density vs. number of LOC under review [28].

The graph makes it clear that after 300 LOC under review, the defects density drops. This is mainly because the reviewer doesn't have time to effectively review that much LOC per sitting.

2 – The reviewers should aim for an inspection rate of less than 300-500 LOC per hour. This one is quite self-explanatory because the reviewer can't give the same attention to all the code. This is not a fixed parameter and should be adapted based on the reviewer itself.

3 – The reviewers should take the time for a proper, slow review, but not more than 60-90 minutes. While too fast is bad, too long is bad as well. The study states that reviewers should not review for too long in one sitting because activities that require people's concentration normally see their performance reduced after 60-90 minutes.

4 – The authors should leave notes for reviewers explaining the code. This study defends that developers should double-check their work and annotate it in order to guide the reviewer. While doing this, authors will uncover some of the defects before the review even begins, hence lowering the density of defects.

5 – It should be established quantifiable goals for code review and a system for capturing metrics in order to improve the processes. This is a good way to check if code review is achieving what is expected from it but also improve it.

6 – Use of checklists. The use of checklists is the best way to avoid omissions because it will remind the author or reviewer to confirm that errors and tests are handled. The study also states that this can be used as a personal checklist in where authors and reviewers can annotate the most common mistakes in order to prevent them to happen again.

7 – Verify that defects are actually fixed. While this might seem obvious, there are actually cases where tracking defects found during review can be difficult, especially in pre-commit code review or other code review techniques like via e-mail. There is not just a good way to do so. There are software tools that automatically track defects, which helps the reviewers to check if the bugs are fixed.

8 – Managers must adopt a good code review culture in which finding defects is viewed positively. This means that managers must promote code review "at a means for learning, growing, and communication" [28] and avoid defects to be seen as a bad thing. It is up to the managers to show that defects are in fact positive, it is an opportunity not only to improve the code but also make the author a better coder and shows that "the author and reviewer successfully worked as a team" [28]. Failing to do so might soar the whole team and the positive results of code review disappear.

9 – Watch out for the 'Big Brother' effect. Most developers assume that they are being monitored in some way, especially if a metrics system is implemented in the review process. While metrics are fundamental for continuous improvement, developers might not be so receptive about it and think that they are actually being measured. When this happens, "not only developers will be hostile to the process, but they will probably focus on improving their metrics rather than truly writing better code and being more productive" refers the study. Once again it is up to the managers to be aware of this effect and promote "the idea that finding defects is good, not evil, and that defect density is not correlated with developer ability", the study concludes.

10 – The Ego Effect. This is one of the positive aspects that code review can offer but can only be achieved when developers feel good with code review (reference to point 8 and 9) because it is a natural effect. Developers, by knowing that someone will be looking at their code, want to look good so tend to write better code and even review some before submitting it for review.

11 – Lightweight-style code reviews are efficient, practical, and effective at finding bugs. The last point highlights the advantages of using a "tool-assisted lightweight review process" by comparing it with formal or heavyweight processes. The results from the study showed that lightweight code reviews took only 1/5th of the time for formal reviews while finding the same amount of defects. The study also concludes that, from several methods to perform code review, the most effective one is with the use of a software tool, referring that a good tool "integrates source code viewing with a chat room" to ease the density of comments per line. Also integration with version control, the ability to track bugs and verify fixes is mentioned.

This study was performed by SmartBeart and highlights the best practices for code review, from a process and a social perspective.

III.2.2. Unit Testing/Integration Testing

Unit testing is an activity pertaining to software testing in which small testable portions of the code, also known as units, are individually and independently examined to ensure that it behaves exactly like it is expected to, under various conditions. This activity fits within the software development process, preferably before the act of code review and acting as a complement to code review by giving early feedback to reviewers that the code works. Unit tests can prevent regression, cases when small portions of code used to work but because they were revisited and thence changed, an inconvenient change can damage what was once operational. Another benefit of having a unit test program within the software development process is that it can provide implicit documentation of how the code is supposed to work. Some commercial software tools provide automated unit testing, some actually integrated with code review, but it can also be performed manually. Manual unit tests require more time and patience from the development team than automated unit tests, but the benefits of having a unit test program surpasses any inconvenience. Unit test can be used to test the functionality of software in the early stages of development, although they do not replace actual functional tests.

When developing embedded software, most of the software is not exposed to the common user, becoming hard to validate it through conventional tests. By performing tests after the software is developed, it becomes not only difficult but expensive (cost, time, etc…) to resolve defects or deviations from the original design. By testing portions of the software as they are developed is one of the best approaches to for this problem, thus, integrating unit tests within the development process is recommended. Each unit test creates a specific input scenario and then checks if it returns the expected value or action. Nevertheless, it may not be possible to test every input scenario and developers should be aware of that.

There are steps for conducting unit testing and how they are best structured. A guide [35], published by IPA/SEC in Japan, describes the standard types of work and best practices for easing the processes in embedded software development. Regarding the implementation of unit tests in the software testing stage, the guide states that it is important to prepare the "environment for implementing embedded software program units, and existing program units that are intend to be reused, among others". There are cases where units have already been developed and can be reused, hence it is important to check if they are still in the usable state. Once the environment is set, the following preparations should be taken into account when conducting unit tests:

- Preparation of test cases for the unit test. In order to verify if the software design can be implemented as defined in previous development stages, there must be some preparation regarding the functional test, and condition coverage test. This requires some preparation around the test data (specific inputs) required to run the test cases.
- Definition of testing criteria. This should include the criteria for determining the overall results of the unit testing activities, with procedures for every scenario.
- Preparation of test cases for verifying any implemented modifications. If a defect is detected, then procedures for determining a solution must be prepared based on the defect. Once the modification has been implemented to resolve the defect, test cases must be prepared as well in order to check the impact of that modification.

There is also a set of precautions that should be taken into account such as: which input values are to be used (boundary conditions or error conditions), contradictions of test cases, among others.

Once everything is prepared to conduct the unit tests, then the implementation of the program units may start. There are various properties that a unit test should have: thorough, repeatable and independent. It is very important that the unit tests are implemented in a way that verifies the results of the code, this is most important part of unit testing (less effective or less informative tests are normally called smoke tests). Regardless of the environment in which the tests are being executed, they should be able to work repeatedly, producing the same results. This is very important for the development team, as they become confident that unit tests will not detect bugs related to environmental differences. Unit tests have to be independent from other unit tests.

Finally, the next step is to conduct the unit tests. The procedures for each possible result should be already defined in the preparation stage so when the unit test results are known, those procedures are taken into action in a natural way.

Once all portions of the code are successfully tested, there is still the possibility of integration testing [36]. Integration testing is another software development process in which units are continuously combined into a component, an aggregate of two or more units, and then tested as a whole in multiple ways. Because each unit is tested before combined, any problems discovered in this phase are probably related to the interface among units, making it easier for the developer to analyse the problem.

There are three major approaches of carrying out an integration test, called the bottom-up method, the top-down method and the umbrella method [36].

- Bottom-up: This approach of integration testing begins with the test of the lowest-level units which are then combined, called utility modules, followed by tests of higher-level combinations of utility modules. With this approach, it becomes easier to account testing progress and utility modules are tested earlier in the development process. However, high-level logic and data flow are tested later;
- Top-down: Top-down integration testing is precisely the opposite of the bottom-up method, beginning with the test and integration of the highest-level modules and then with lower-level modules. This allows higher-level logic and data flow to be tested earlier in the development process but bugs are harder to find; and
- Umbrella: The umbrella approach combines both top-down and bottom-up testing. In this approach, bottom-up testing is done first, followed by top-down testing. The advantage of this approach is that it supports detection of limitations on an early release of the software product. It also helps minimize the disadvantages of the bottom-up and top-down approach.

The integration testing process finalises when all modules are grouped together and tested. This activity fits within the testing phase of the software development process and if done right, it may reduce the risk of defects being found by the client. Like every process, it is recommend to keep the entire process documented.

Unit testing comes at the expense of investing time to write a suite of tests early in development. However, as the project grows and the development team gets familiar with it, unit testing is a huge addiction to the software development process as it will help developers to improve the overall quality of the code and reduce the density of defects found when the code is submitted for review.

III.2.3. Metrics

Most products of the software process have many properties or features that might be useful to measure. These measurable properties and features are known as metrics and can be found in code, test documentations, design specifications and almost everything that is an output of the software process. Code review is no exception and can be a great source of metrics. Citing Paul Goodman [29], he defines software metrics as "The continuous application of measurement-based techniques to the software development process and its products to supply meaningful and timely management information, together with the use of those techniques to improve that process and its products." This basically means that by creating a process to measure metrics, it can lead to a better understanding and prediction of software projects so whoever is in charge can make more informed decisions and intelligent choices. When metrics are being obtained from the act of code review, the difficulty to obtain those metrics depends on the technique used for code review. With the use of a tool–assisted code review, the process of measuring becomes easier because some tools are already designed to collect metrics.

There are a lot of possible software metrics, some more relevant and useful than others, which leaves the organization to determine which metrics are right for them. Quoting Watts Humphrey's [30] on this matter, he says "There are so many possible measures in a complex software process that some random selection of metrics will not likely turn up anything of value." It is critical for the organization not only to define the right metrics but also knows how to measure those metrics. A good mechanism for this process is to adopt Basili and Rombach's Goal/Question/Metric paradigm [31] that is illustrated below.

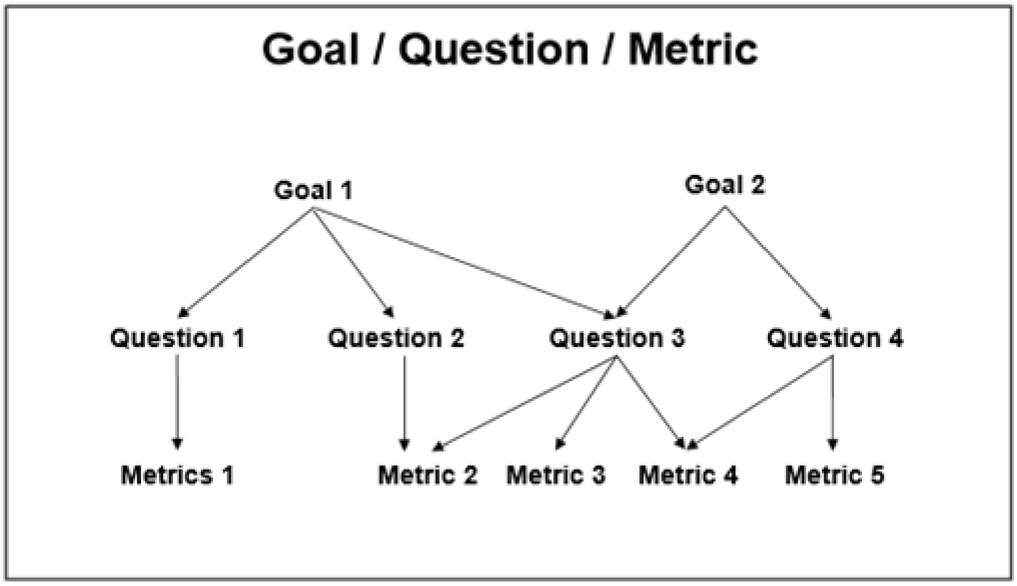

Figure 17 – Goal/Question/Metric Paradigm [31].

Initially, the organization needs to set goals for projects, then formulates questions that can be answered by measuring data and lead to the success of that goal. By defining a goal-based measurement program, the organization will be able to efficiently set measurable goals. If in any case a metric is being produced but not used, it may as well be discontinued because producing it is a waste of time.

The organization should define who will be responsible in collecting the data for the metrics. Normally the personnel more interested are part of the software development team (Programmers) and the ones responsible for reviewing/testing the code. Not only who collects data, but all personnel involved in tasks that involve data management must understand which metrics are behind that data and understand the purpose and usefulness of it. The management should support problems and issues related to data collection because data collection activities require time and resources. Providing a training program might be a good practice to ensure that the personnel collecting the data understand what to do and when to do it. It is recommended that procedures are established and documented, as part of the preparation for the training program. These training programs do not require to be long, an hour can be enough to provide simple collection mechanisms.

The use of a training program promotes a common understanding about when and how to collect data and decreases the time that support staff spend in answering the same questions over and over.

When the process to identify metrics is completed, the organization has to decide what to do with the results. According to ISO/IEC 15939, a standard that defines a measurement process applicable to software engineering [32], decision criteria are the "thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result". This means that it is required to have decision criteria in order to obtain guidance, which will then help interpret the measurement results.

"Data collection must be objective, unambiguous, convenient and accessible" [33] and the most efficient process to do so is creating an automated process for reporting and delivering metrics. The main objective in the end is to have a process that collects data accurately so the measurement program can be accomplished.

Regarding code review, the produced metrics can also be used to measure the progress of a development team. It can pinpoint areas where the development practice is weak, enabling the ability to address that weakness and where the development practice is strong. Metrics can also be recorded to check the performance of the code reviewers, although it is not recommended, and the review process itself.

The figure below shows a simple code review process and how metrics can be integrated in it.

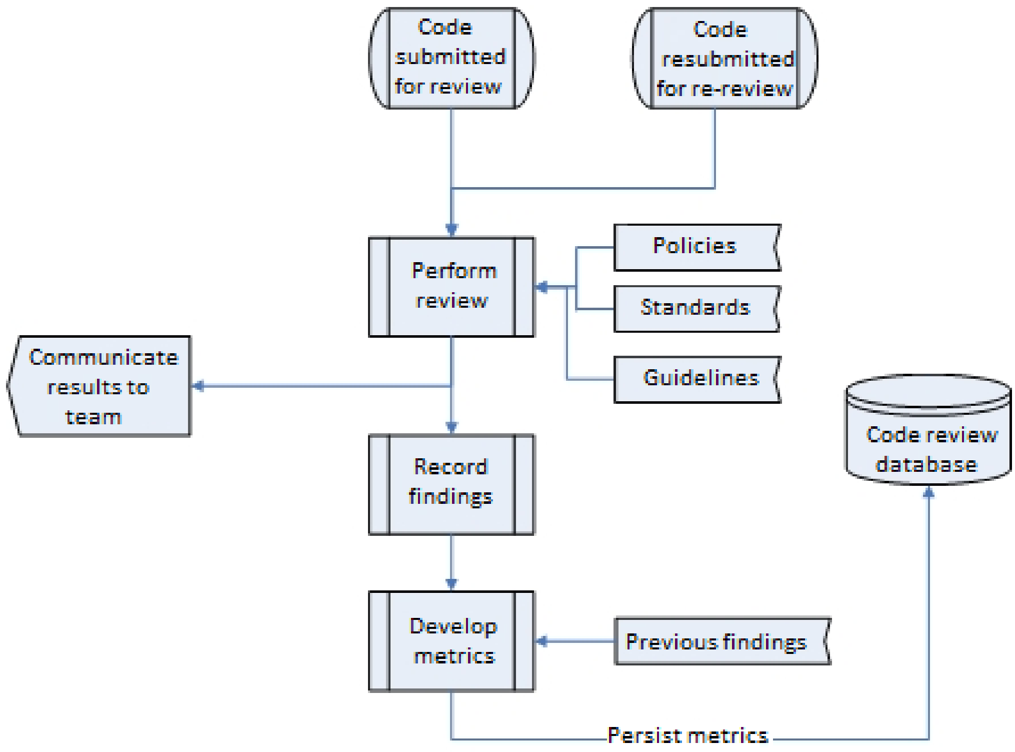

Figure 18 – Code review/metrics process (adapted from [34]).

The figure illustrates a simple measurement process, where the process of obtaining metrics is directly related to the process of review.

Software metrics are not as easy as it seems because few standards cover this topic [33]. For a successful measurement process, it is essential that the organization create their metrics program based on the organization's goals so the personnel will work to accomplish what they believe to be important. This also helps with communication, which is the key for success. Continuous improvement inside an organization is fundamental as it is covered by most quality related standards, thus a well-documented metrics program can help the organization in that matter.

III.3. Implementation of the new methodologies

III.3.1. Code review Tools

This clause provides a brief look at the currently available code review tools. The tools are divided among two categories, commercial and non-commercial tools, but regardless of the one used, they all serve pretty much the same purpose – retrieve the most recent changes to source code and flag them as needing review.

III.3.1.1 Commercial software

As the name suggests, commercial software requires the user to pay a certain amount of money to use it. A list of commercial tools will presented below, along with their most relevant features. As tools differ in functionalities and features, so does the price and depending on the manufacturer, the payment process can either be by acquiring a timeless license or by having a monthly/annual fee. The tools considered in this section may differ in how information is displayed due to specific restraints found when searching for information. As for example, some manufacturers provide their products price information while others require customer specific identification.

- Parasoft [37]: Parasoft's products are focused in three areas:
  - Development testing;
  - Application Programming Interface, Cloud, Service Oriented Architectures, Composite App testing; and
  - Service Virtualization.

  Concerning the development area, Parasoft offers distinct products for C/C++, Java and .NET code testing. These products provide static analysis, peer code review, unit testing and runtime error detection. For embedded and cross-platform development, Parasoft offers a "spin-off" of the C/C++ product and it can be used in both host-based and target-based code analysis and test flows. This products integrate with Parasoft's Concerto reporting system, which provides an interactive web-based dashboard. Basically it is a development management tool for software that allows teams to track project status and trends based on the test results and other key process metrics.

- SmartBear [38]: SmartBear offers a different product line than Parasoft, although they share a common area: software development. The manufacturer offers a wide range of software development products such as:
    - Collaborator – Code, documents and other project deliverables review;
    - ALMComplete – Application Lifecyce Management;
    - AQtime Pro – Performance Profiling;
    - Automated Build Studio – Software release management;
    - SoapUI Pro – API testing; and
    - DevComplete – Software Development Management.

  Regarding *Collaborator*, it offers a wide set of features such as: asynchronous reviews, user threaded and contextual chat, real-time chat, workflows, defect tracking and management, metrics and reporting. It also enables the review of other project deliverables. In terms of pricing, SmartBear provides two different licenses: the "named user" that can be acquired for $489 or the "concurrent user" that goes for $1499. Both licenses are timeless.

- Atlassian [39]: This manufacturer offers a wide variety of products and most of them is capable to integrate with each other, as long as Atlassian's core product is owned, *JIRA*. This tool is focused on project and issue tracking and can have its functionalities expanded to other Atlassian products. Regarding code quality products, the manufacturer offers *Crucible* for peer code review; *FishEye* for browsing and searching code; *Clover,* a tool that covers java and groovy code; and finally, *Bamboo,* which is focused on continuous integration. Crucible alone lacks in features when compared to Collaborator or Parasoft's products, but if Crucible and FishEye are both acquired, some functionalities that are found in FishEye can be expanded to Crucible and the user gets additional features such as diff view, in this case. The price of *Crucible* can go from $10 (5 users) to $8000 (100+ users). Although the license is timeless, software maintenance (support and product updates) will only be provided for 12 months unless it's renewed. The price for renewal is the same as the one paid for the license chosen. There is also a “market store” with free and payable add-ons for Atlassian products.

- Klockwork [40]: Klockwork does not offer as many products as the others manufacturers. Currently they have two products, *Klockwork Insight* and *Klockwork Cahoots*. *Klockwork Insight* combines on-the-fly analysis, drag & drop build reporting, and cross-project impact analysis. It covers C/C++, Java and C# code. The other product, *Klockwork Cahoots*, is a code review tool and it features customizable reports, code and other textual document reviews, real-time notifications and more. The license for *Cahoots* can be obtained for $49 for one year or $199 for three years per user. This prices are limited time launch prices and might be object to change.

- BeanBAG [41]: BeanBag *RBCommons* is another tool for code review and it differs from the others because of the payment method, having a monthly fee instead of a lifetime license. This tool offers the most basic features you want in a code review tool, as example, it is possible to create review requests and generate reports. The fee varies according to the number of users and repositories requested. The “starter edition” allows the use of 10 users and 3 repositories for $29/month, the “medium edition” ups to 25 users and 20 repositories for $99/month. There is still the large and enterprise editions with doubles the users and repositories offered and are priced for $179/month and $499/month respectively.

III.3.1.2 Non-commercial software

This section lists some of the non-commercial tools for code review. Most of the tools can’t compete with the commercial tools, in terms of features and functionalities, but some may be good options if an organization pretends no more than just a tool to perform simple code review. The use of non-commercial tools can also be a way for organizations to implement and test code review in their development process without compromising funds.

- Rietveld [42]: This tool is an open source project that uses Google Apps and it is a web-based code review. This tool is mostly used on Python projects but it can be used to code review any language. Rietveld supports Git, Subversion, Mercurial and Perforce repositories.
- Phabricator [43]: Phabricator is an open source collection of web applications that was created at Facebook. This tool offers code review, bug tracking and more. It also has a wiki and an active community. Works with Git, Subversion and Mercurial repositories.
- AgileReview [44]: AgileReview is a code review tool that uses the Eclipse IDE. This tools offers the very basic features of code review.
- Codestriker [45]: Codestriker is an open-source web application, written in Pearl, which supports online code reviewing. This tool offers configurable metrics and supports document review. Can be integrated with CVS, Subversion, Perforce and more.
- Review board [46]: Review Board offers a web-based interface with broad browser support for managing review requests and reviewing code. It is also possible to use command line tools for the review request submission process.

III.3.1.3 More than just code review

As mentioned before, current available commercial tools have more than just a feature or functionality. As a matter of fact, companies might use this to attract customers to buy their products by offering a wide set of techniques to improve code quality. Likewise, companies have different approaches on how they sell their products. Some offer the whole package like Parasoft and you get code review, unit testing among many others functionalities for a considerable price, while other companies such as Atlassian offer various products that integrate with each other making it feel "customizable". Some companies are more focused on embedded development than others and offer products with functionalities like unit testing which is more focused on meeting safety-critical standards.

LDRA is an example of a company more focused on embedded software. It provides automated analysis and testing tools for applications in which people's lives depend on [47]. This company offers a wide range of products such as TBVision, a tool that provides transparency into source code, quality metrics, etc. TBrun is LDRA's unit test tool and provides a complete integrated framework for the automated generation and management of unit tests. Most LDRA products are able to integrate with themselves.

III.3.1.4 Code review - Features comparison

The table 3, in the following page, shows an overview of the features from the tools that were covered.

By analyzing the table is possible to realize that most tools support integration with Subversion and Git. The fact that these version controls systems have plugins available for most IDEs, providing support for them is a desirable feature.

All commercial tool have their own interface while most free tools have theirs based on web IDEs. This is just a matter of refinement and mainly due to the availability of resources that a company has for this purpose.

Regarding the Post and Pre commit options, that really depends on what an organization wants. Offering support for both types is clearly an advantage over other tools but most commercial tools do have them. Regarding free tools, only Phabricator and Review board provide that feature.

Table 3 – Comparison matrix of the different Code Review tools.

Features / Products	Price	Side by side comparison	Metrics Report	Diff files	Pre commit review	Post commit review	Interface	Integration with
Parasoft	Not provided	Y		Y	Y	Y	Stand alone	SVN; Git; CVS; and more
Collaborator	Named user: $489 Concurrent user: $1499	Y	Y	Y	Y	Y	Stand alone	SVN; Git; CVS; and more
Crucible	10 Users: $10 100+ Users: $800	Y(Fish eye)	Y	Y(Fish eye)	Y	Y	Stand alone	SVN; CVS; Preforce
Cahoots	One year: $49 Three years: $199	Y		Y	Y	Y	Stand alone	SVN; Git; CVS; and more
RBCommons	10 users: $29/month 25 users: $99/month	Y		Y		Y	Stand alone	SVN; Git; CVS; and more
RietVeld	Free	Y		Y			Web	SVN
Phabricator	Free	Y		Y	Y		Web	SVN
Agile Review	Free		Y				Eclipse IDE	SVN
Codestriker	Free	Y	Y	Y			Web	SVN
Review Board	Free	Y		Y	Y		Web	SVN

One of the most important features for a code review tool is having support for *diff* files and most commercial tools offer this except for Atlassian *Crucible*, which requires the purchase of another Atlassian product called *FishEye* to make it available. Even most free tools support diff view, *Agile review* is the exception which is clearly a disadvantage.

Another important feature is the possibility to have metrics about the accuracy of the code review process, the performance of the code reviewers and more. This feature is not so common among commercial and free tools and that gives an edge over the other products.

Any type of code review is better than none, so either it is a paid or a free version, an organization should always choose one that fits what they do and what they need, in order to achieve competitive advantage.

III.3.4. Unit Testing

The unit test software selected for testing embedded code is called *CppUTest* and is written in C++ but can also work for C. This testing framework is used by many C programmers around the world [48]. It supports multiple OS platforms and is especially used in embedded development. The software is quite simple to set and use.

Tests run as part of an automated build using *make*. When no errors are found, the test output looks like this:

-> OK (2 tests, 2 ran, 0 checks, 0 ignored, 0 filtered out)

The output is minimal when all tests succeed. The interpretation of the summary line is quite simple:

- *Tests* is the total number of test cases;
- *Ran* is the total number of test cases that ran;
- *Checks* is a count of the number of condition checks made. (Condition checks are calls such as CHECK_EQUAL( ).);
- *Ignored is* a count of the number of tests in ignore state. Ignored tests are compiled but are not run; and
- *Filtered out is* a count of the number of tests that were filtered out of this test run.

When a test fails, the output looks like this:

-> Errors (1 failures, 2 tests, 2 ran, 1 checks, 0 ignored, 0 filtered out)

The failure reports the line of the failing condition check, the name of the test case, and the reason for failure. Also, in the summary line, there is an additional parameter that counts the number of failed tests.

IV. Conclusions and future work

The findings from this study show that by having a certified quality management system, an organization will notice an increase in theirs operational effectiveness, hence, benefit from a competitive advantage over direct competitors for the market. By adopting a policy of quality, the organization will not only increase customer satisfaction but also increase opportunities in specific markets and give confidence to the management. This study was based in the ISO 9001:2008 standard for the requirements of a QMS and, in order to obtain positive results from a QMS, it is fundamental to contemplate several requirements such as: solutions must be adequate to the organization's characteristics; true commitment from top management to maintain a QMS; involvement of all collaborators so they all feel responsible for the QMS; and having clear objectives. By the fact a QMS is based on objective evidence, organizations must keep in mind that:

- what is currently being made should be documented;
- do what is documented; and
- keep records of what has been done.

The study also contemplates the NP 4457:2007 standard for the requirements of a RDI management system. Organizations, by adopting an innovation strategy, can also exceed other market players by introduction of a completely new or notably better products or services. Only those who invest in their RDI department can achieve such statuses.

From several project management methodologies, this study covers PRINCE2 and ISO 21500, which is actually a guide that provides insights on the concepts and processes of project management, recognized by most international organizations. The benefits of being in accordance with such methodologies yields several opportunities to formulate appropriate strategies and restructure current activities to serve customers, aiming to attain the same or even better results at lower costs, by improving the overall development performance and business profitability. By organizing the development processes to be carried out by the people who are involved in embedded software development, the organizations will be able to:

- Verify that all the activities are carried out without fail, and re-examine the activities that may seem unnecessary;
- Set the framework for conducting elaborate work to achieve the functionalities required by the customers and/or market, ensuring that they are reliable and of quality;
- Ensure collaboration between teams allocated with specific jobs;
- Define the information that is used within activities (e.g.: information regarding the work products, deliverables, names of these activities, and more).

Most methodologies are generic and should be adapted to the reality of each organization.

This dissertation combines an analysis of several activities and the empirical data present in this study provides a strong indication that code review and unit testing can be a powerful tool to embedded software developers. Implementing an integration testing process, a metrics system, among many other measures are some of the options that management can accomplish. These processes should be used as means for learning, growing, communication and improvement. Hence, managers must adopt a culture where finding defects are viewed positively and avoid, by any means, the use of these activities for controlling and measuring the personnel who uses them.

If correctly adopted, these activities will improve the entire development process, finding bugs early, improving code quality, sharing experience and knowledge, reducing development time are just some of many improvements that developers and management will benefit. Various tools for practical implementation were studied. Among those tools, stand out *Phabricator* and *CppUTest*.

When planning the development project, it is necessary to predict the time and period taken from the start of the development until the final product is shipped out from the manufacturing site.

There is certain information that must be taken into account, prior to determining what types of work are required for the development such as: the characteristic features of the system and software that is to be developed; the special conditions of the development projects; and the conditions of the organizations involved in the development. The information provided in this dissertation is intended to cover the development and testing work deemed necessary in developing embedded software. Depending on the characteristics of the software to be developed, some of the activities and insights provided in this dissertation can be omitted.

Although these activities were partially implemented within a test case, provided by *Exatronic*, it was not enough to visualise and understand the impact of these activities. In order to corroborate what was studied in this dissertation, it would be necessary to apply this methodologies within a practical case and analyse how people would adapt and deal with these tools, processes and, most importantly, analyse the impact that these methodologies would in fact bring to the overall development process.

V. References

[1] Hennessy, D. (2005). Computer Organization and Design – The Hardware/Software Interface. San Francisco, CA, USA. 3rd edition: Morgan Kaufmann Publishers.

[2] Longo, R. (1966). Gestão da Qualidade: Evolução Histórica, Conceitos Básicos e Aplicação na Educação. Available: http://www.dcce.ibilce.unesp.br/~adriana/ceq/Meterial%20complementar/historia.pdf [Accessed: Feb. 20, 2013].

[3] Sousa, A. (2007). A relevância da informação na gestão pela qualidade: Contabilidade e Gestão. Available at: http://www.otoc.pt/downloads/files/1232466483_artigo-3.pdf [Accessed: Feb. 21, 2013].

[4] Gomes, A. (2004). A evolução do conceito de qualidade: dos bens manufaturados aos serviços de informação. Vol 002: Cadernos de Biblioteconomia Arquivística e Documentação Cadernos BAD.

[5] Evans, J., & Lindsay,W. (2008). The Management and Control of Quality. 7th edition. Canada: Thomson Coporation.

[6] Costa, B. (2007). Sistemas Gestión de la Calidad en las Empresas Portuguesas: Implantación, Impacto y Rendimiento. Sevilha: Universidade de Sevilha: Tesis Doctoral. Dirección de Empresas y Gestión de Marketing.

[7] Pires, A. (2004). Qualidade: sistemas de gestão da qualidade. Lisboa: Sílabo.

[8] AEP - Associação Empresarial Portuguesa. (2003). Manual Pedagógico PRONACI Qualidade.

[9] ISO 9000. (2005). Quality management systems - Fundamentals and vocabulary. Switzerland: International Organization for Standardization.

[10] ISO 9001. (2008). Quality Management Systems – Requirements. Switzerland: International Organization for Standardization.

[11] ISO 9004. (2005). Managing for the Sustained Success of an Organization – A Quality Management Approach. Switzerland: International Organization for Standardization.

[12] CT 80 (APQ). (2008). Sistemas de gestão da qualidade - Requisitos. NP EN ISO 9001:2008.

[13] APCER. (2010). *http://www.apcer.pt.* Available at: http://www.apcer.pt/index.php?option=com_smartformer&Itemid=482 [Accessed: Feb. 16, 2013].

[14] Kline, S., & Rosenberg, N. (1986). "An overview of innovation." In R. Landau & N. Rosenberg (eds.), The Positive Sum Strategy: Harnessing Technology for Economic Growth. Washington, D.C.: National Academy Press, pp. 275–305.

[15] OCDE. Manual de Oslo. "Diretrizes para coleta e interpretação de dados para i novação". Available at: http://www.oei.es/salactsi/oslo2.pdf [Accessed: March 10, 2013].

[16] NP 4456. (2007). Gestão da Investigação Desenvolvimento e Inovação (IDI). Terminologia e definições das actividades de IDI. Available at: http://www.ipq.pt /custompage.aspx?pagid=4050 [Accessed: March 12, 2013].

[17] NP 4457. (2007). Gestão da Investigação Desenvolvimento e Inovação (IDI). Requisitos do sistema de gestão da IDI. Available at: http://www.aip.pt/?lang=pt &page=inovacao/consultoria.jsp [Accessed: March 12, 2013].

[18] APCER. (2007). Associação Portuguesa de Certificação. Available at: http://www.apcer.pt/index.php?option=com_content&view=article&id=141%253 Anp-4457&catid=10&Itemid=60&lang=pt [Accessed: March 25, 2013].

[19] CCTA. (2013). The Unique Online Guide to Running PRINCE2 Projects. Available at: http://www.fee.unicamp.br/ieee/PRINCE2.pdf [Accessed: Aug 17, 2013].

[20] Murray, A. (2007). Everything you need to know about PRINCE2 in less than on thousand words! Available at: http://www.best-management-practice.com/gem pdf/PRIN CE2_White_Paper_v3.pdf [Accessed: Aug 17, 2013].

[21] Turley, F. (2010). The PRINCE2 Foundation Training Manual. Availble at: http://www.mgmtplaza.com/elearn/dirBooks/PRINCE2-Foundation-Training-Manual.pdf [Accessed: Aug 30, 2013].

[22] A. Murrey. (2011). Management of Risk: Guidance for Practitioners. Available at: http://www.best-management-practice.com/gempdf/prince2_in_one_thousand_words.pdf [Accessed: Aug 11, 2013].

[23] OGC. (2009). Managing Succesfull Projects with PRINCE2: The Stationery Office.

[24] Zandhuis, A., & Stellingwerf, R. ISO 21500: Guidance on project management – A Pocket Guide. Van Haren Publishing.

[25] ISO. (2012). ISO/DIS 21500: Guidance on project management.

[26] Buttrick, R. Prince2 and the National and International Standards. The Stationery Office. Availabe [Online] at: http://www.best-management-practice.com/gempdf/PRINCE2_National_Standards.pdf?goback=%2Egmp_96642%2Egde_96642_member_198686783 [Accessed Oct, 10, 2013].

[27] Devart. What is Pre-Commit and Post-Commit Review?. Available at: http://www.devart.com/review-assistant/learnmore/pre-commit-vs-post-commit.html [Accessed: Apr. 17, 2013].

[28] SmartBear Software. (2011). 11 Best Practices for Peer Code Review. [Online]. Available at: http://support.smartbear.com/pdf/11_Best_Practices_for_Peer_Code_Review.pdf [Accessed: Apr. 19, 2013].

[29] Goodman, P. (1993). Practical Implementation of Software Metrics. McGraw Hill, London.

[30] Humphrey, W. (1989). Managing the Software Process. Addison-Wesley, Reading.

[31] Basili, V., & Rombach, H. (1988). The TAME project: Towards Improvement-Oriented Software environments. In IEEE Transactions in Software Engineering.

[32] ISO/IEC 15939:2002. International Standard, Software Engineering – Software Measurement Process.

[33] Westfall, L. 12 Steps to Useful Software Metrics. Available at: http://floors-outlet.com/specs/spec-t-1-20111117171200.pdf [Accessed: Set. 19, 2013].

[34] Open Web Application Security Project (OWASP). Code Review Metrics [Online]. Available at: https://www.owasp.org/index.php/Code_Review_Metrics [Accessed: Apr. 20, 2013].

[35] IPA/SEC. (2012). ESPR, Embedded System development Process Reference Guide. Japan. Available at: http://www.ipa.go.jp/files/000028853.pdf [Accessed: Oct. 07, 2013].

[36] MICROSOFT developer network, Available at: http://msdn.microsoft.com/en-us/library/aa292128%28v=vs.71%29.aspx [Accessed: Oct. 07, 2013].

[37] PARASOFT. [Online]. Available: http://www.parasoft.com/jsp/capabilities/code_review.jsp?itemId=554 [Accessed: Apr. 22, 2013].

[38] SMARTBEAR COLLABORATOR. [Online]. Available: http://smartbear.com/products/software-development/code-review [Accessed: Apr. 21, 2013].

[39] ATLASSIAN CRUCIBLE. [Online]. Available: http://www.atlassian.com/software/crucible/overview [Accessed: Apr. 21, 2013].

[40] KLOCWORK INSPECT. [Online]. Available: http://www.klocwork.com/landing/software-code-reviews-software/index-v1.php [Accessed: Apr. 21, 2013].

[41] RBCOMMONS. [Online]. Available: https://www.rbcommons.com/tour/ [Accessed: Apr. 21, 2013].

[42] RIETVIELD. [Online]. Available: http://code.google.com/p/rietveld/ [Accessed: Apr. 21, 2013].

[43] PHABRICATOR. [Online]. Available: http://phabricator.org/ [Accessed: Apr. 21, 2013].

[44] AGILEREVIEW. [Online]. Available: http://www.agilereview.org/ [Accessed: Apr. 21, 2013].

[45] CODESTRIKER. [Online]. Available: http://codestriker.sourceforge.net/ [Accessed: Apr. 21, 2013].

[46] REVIEW BOARD. [Online]. Available: http://www.reviewboard.org/ [Accessed: Apr. 21, 2013].

[47] LDRA. [Online]. Available: http://www.ldra.com/ [Accessed: Apr. 27, 2013].

[48] Grenning, J. (2011). Test Driven Development for Embedded C. The Pragmatic Programmers.