Technologies and Information Systems Division

Castilla-La Mancha University

PhD Thesis

Measurement Framework for the Definition of Software Measurement Programs in SMEs: MIS-PyME

PhD Student: Ms. María Díaz Ley

Supervisors: Dr. Mario G. Piattini Velthuis

Dr. Félix Óscar García Rubio

Technologies and Information Systems
Division

Castilla-La Mancha University

PhD Thesis

Measurement Framework for the Definition of Software Measurement Programs in SMEs: MIS-PyME

PhD Student: Ms. María Díaz Ley

Supervisors: Dr. Mario G. Piattini Velthuis

Dr. Félix Óscar García Rubio

To Domingo and Pablo Guhl, my main motivation for finishing this work

To my family, for their loving support. A family that I am proud to belong to.

Acknowledgments

I would first like to thank my supervisors Félix García and Mario Piattini for their professionalism and their continuous efforts. They have been an excellent model to follow. I also owe them my gratitude for giving me the marvellous opportunity to work with their research group (Alarcos).

I truly appreciate the work of Tamara Mata with the MIS-PyME Tool and I thank her for joining this research project.

I thank all the members of the Alarcos research group for their cooperative spirit and their dedication, two of the main things required to obtain great results.

I am profoundly grateful to Salvador Fernández and Luis de Reyna for being the first to sincerely trust my professional skills.

I thank Roberto Moriyón for helping me and for channelling my research efforts towards this particular area.

I am grateful to Jesús Mayoral, José Luis Sánchez, and Juan José Muñoz for giving me the chance to develop this research project at STL.

I would also like to thank Guillermo, Lucía and Luis Miguel for reviewing this text, for being so special and for their yearlong friendship.

I could not possibly forget all the great teachers I have had throughout the years who have, in one way or another, helped me to achieve this goal: Aurora, María Antonia, Román, María José, Mateo, Esperanza, Sagrario, Amado, Esteban and especially, José Luis.

I thank my friends: Ana, Ángela, Annalisa, Atocha, Bárbara, Carmen, Cristina, David, Elena B., Elena C., François, Gloria, Hugo, Merche, Pascal, Patricia, Yaiza for their tremendous support all these years. They are one of the best parts of my life.

I would really like to thank Aula 4, Los Lebowski and especially NUR for bringing music into my life.

And finally, gratitude to my colleagues and friends from STL (and Félix) for each of the wonderful moments we shared together on a daily basis. They have made my work days happy.

Index

Table of Contents

Index of Figures

Index of Tables

Abstract

The successful implementation of software measurement programs is not an easy task. Software measurement program implementations sometimes fail, and this failure is frequently caused by the fact that measurement programs are not usually well defined.

The definition of measurement programs is even less encouraging in Small and Medium Enterprises (SMEs). Generally, SMEs do not have enough resources to promote serious measurement program initiatives; training is more difficult and software measurement knowledge is especially poor in this context. Unfortunately, the major methods and frameworks that support measurement programs –such as Goal Question Metric (GQM), Goal-Driven Software Measurement(GQ(I)M), Practical Software and Systems Measurement (PSM) and ISO/IEC 15939 are not suited to this kind of companies and literature provides few cases of measurement program implementation which could be used as a source of reference for good practices through which to successfully perform these activities.

In addition, one important reason for the failure of measurement program implementation is that the maturity of companies as regards measurement has not been taken into account during the definition phase. Nevertheless, companies wish to define measurement programs which are adapted to the measurement maturity of the company and improve their measurement maturity. Few measurement assessment models exist, and those that do are neither familiar nor are they applied in industry. Fewer still are integrated into a methodological framework for the definition and implementation of software measurement programs.

In this research, a methodological framework for the definition of software measurement programs called MIS-PyME is developed and validated. MIS-PyME is desgined as a lightweight methodology which takes into account the maturity and the limitations of the company. MIS-Pyme Framework provides a methodology which consists of a measurement program definition process, a set of roles and three support modules to ease the definition of the measurement program. In addition, MIS-PyME Framework provides a measurement maturity model used to define measurement programs adapted to the measurement matutiry of the company and to assess the measurement maturity of companies and identify software measurement improvements.

This thesis shows the related state of the art, its motivation and its research principles. The methodological framework and its valiation are also presented. The motivation of the research is supported by a Case Study in which a popular software measurement program methodology (Goal Question Indicator Metric - GQ(I)M) was applied in order to study its suitablilty in SMEs. MIS-PyME was validated by performing two Case Studies in the software development and maintenance department of a medium-sized company. The aim of the first Case Study was to understand the suitability of MIS-PyME (which was adapted to the company's maturity) for defining and implementing measurement programs in SMEs, along with understanding the benefits obtained by using MIS-PYME in contrast with the previous experience of using GQ(I)M. The aim of the second Case Study was to understand the suiatbilty of MIS-PyME Measurement Maturity model for assessing the measurement maturity of software development SMEs and for identifying measurement improvements.

Abstract

1. Introduction

This chapter explains both the subject matter and importance of this thesis. The hypothesis and goals, the sponsored research projects and the structure of the thesis are also described.

1.1. Background and Motivation

Software measurement is becoming increasingly important in software engineering. Software measurement is an essential tool for software control and management in any organization, especially in large-scale ones. The Project Management Body of Knowledge (PMBOK) (IEEE, 2003) defines five groups of processes: Initiating, Planning, Executing, Controlling and Monitoring, and Closing (see figure 1-1). Measurement occupies a relevant place in the planning, controlling and monitoring of processes, in which it is the main activity.

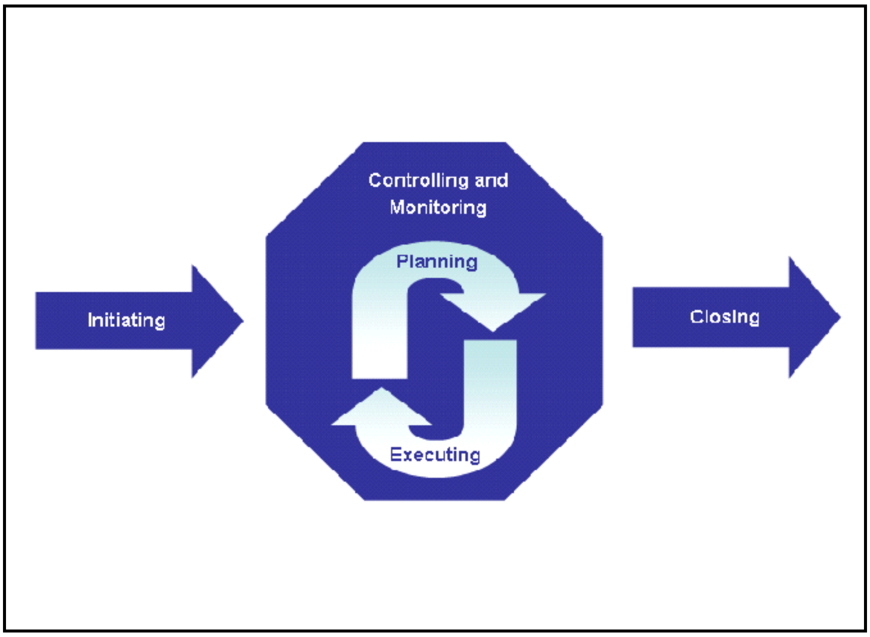


Figure 1-1. Processes Groups in PMBOK

Organizations understand the need to control and monitor their projects by means of measuring and taking actions in order to meet project goals. Certain measures are used to monitor schedule and work progress, the effort allocation vs. effort planned, etc. The quality and development processes should also monitor and control the quality of the product being developed. In production, attributes such as the ease of maintaining the product, its reliability, efficiency, usability, etc. are some aspects of the product that must be controlled when developing and maintaining software.

Measurement is also essential for controlling and improving the process (see Figure 1-2). Attributes of the process such as its effectiveness or efficiency and its cost should be measured in any organization that develops software. Once its current state has been quantitatively characterized it is possible to change the process and then determine whether an improvement has occurred. Collecting and interpreting well-defined data provides the organization with the necessary information to make well-founded decisions for controlling the process and improving it (Brijckers and Differding, 1996).

Managing projects and software development, and controlling and improving processes in the organization by means of measurement results leads to increased productivity, quality, and reduced cycle time, all of which make a company competitive in the software business (Daskalantonakis, 1992).

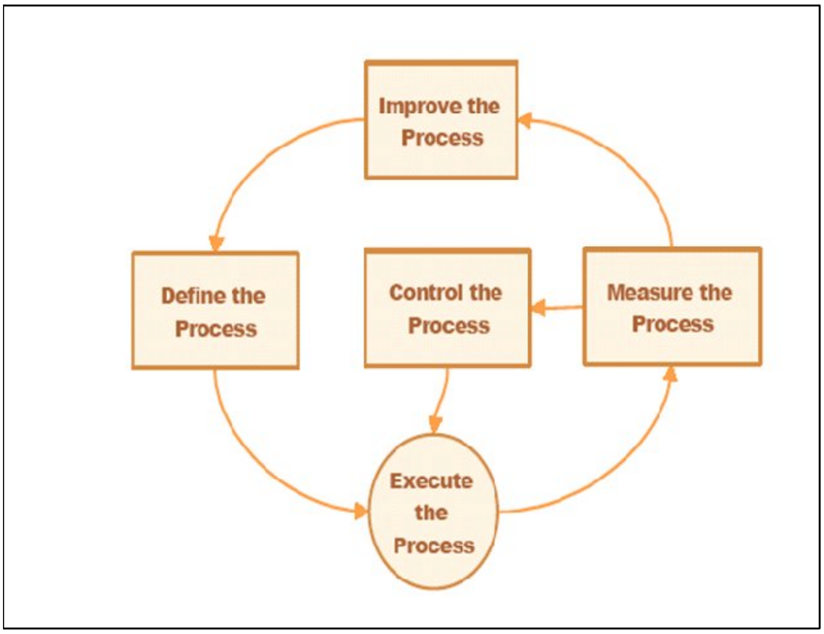


Figure 1-2: Key Process Management Responsibilities (Florac and Carleton, 1999)

The importance of software measurement is more evident when we study the “Guide to the Software Engineering Body of Knowledge” (SWEBOK) (Abran et al., 2004b). In this guide software measurement is present throughout all the key areas. Additionally, more importance was given to software measurement when Abran et al. (2004a) suggested that a new Software Measurement KA (Key Area) be included in the SWEBOK, in order to consider Software Measurement as a key area itself in the world of Software Engineering.

Most software organizations are aware of the importance of measurement programs and have accordingly started to work in this field, but in spite of the benefits derived from measurement and the fact that measurement is applied in various areas, this has proved to be a complex and difficult undertaking in the field of software. Some studies show that 50 to 80% of measurement programs do not continue beyond their second year (Rubin, 1991) and that two in three metric implementations fail (Pfleeger, 1999a), and these results are not encouraging. One of the main reasons for this unsuccessful rate is that there is poor software engineering knowledge in companies and especially in the software measurement area. Effective software measurement requires a great deal of information and models, and decisions which must be documented. This is therefore a particularly difficult task for those people who do not have extensive experience with software measurement (Briand et al., 1996). Poor software measurement knowledge may lead companies to define measurement programs with pitfalls, some of which are as follows:

- Trying to achieve the "best" measurement goals and not those that the company is able to successfully implement.
- Defining measures whose input data cannot be easily obtained, or posing data collection and analysis mechanisms which are insufficient or poorly organized (Selby, 2005).
- Defining measurement programs without taking into account the opinion and needs of the workers who are most interested in the measurement program. Measurement must be popular among all measurement stakeholders and must be integrated in the corporate culture of an organization in order to succeed (Selby, 2005).
- Defining measurement programs without clearly identifying their main purpose.
- Measurement analysis and interpretations are not performed by the most appropriate people.
- Feedback from the measurement analysis and interpretation of the results is not performed appropriately for all the interested parties.
- Extra effort is allocated to measurement data to back up decision-making when there are project tight deadlines. As a result, there is a very high risk that no data at all, or otherwise incomplete or inaccurate data, are obtained (McGarry et al., 1998).

As a result of these undesired practices, certain people have bad feelings with regard to measurement and they do not trust in it. In addition, they view this activity as a way in which to unfairly evaluate their work and some resistance still remains among technical people, probably owing to inner fears that measurement is and will be about people and not processes (Buglione and Dekkers, 2006). In addition, some stakeholders judge measurement programs to be too exhaustive when compared with their benefits. The link between an investment in measurement programs and the business return is attenuated and complex (Briand et al., 2002)

Small organizational units are just as likely to be confronted with demands for credible evidence of their ability to deliver quality products on time and on budget as large, multinational organizations are. Similarly, managers in small settings are equally or even more likely than their counterparts in larger organizational units to have to make well-founded business decisions about process improvement and the adoption of technology, and must have the wisdom to take new business opportunities. Implementing serious measurement programs is therefore even more important in small organizational settings (Goldenson et al., 2005). However the tendency towards failure in the successful implementation of measurement programs is particularly outstanding in the context of Small and Medium Enterprises (SMEs) (Gresse et al., 2003), the reason being that the factors which characterize these companies usually become the cause of the problem. Generally, SMEs do not have enough resources to promote serious measurement program initiatives; training is more difficult and software measurement knowledge is especially poor in this context. The work of (Kasunic, 2006) shows how measurement practices are not as popular as in medium or large companies: the use of software measurement increases as the size of the organization increases.

SMEs have an additional handicap: the existing methods and frameworks that support measurement programs such as Goal Question Metric (GQM) (Solingen and Berghout, 1999), Goal-Driven Software Measurement or Goal Question Indicator Metric (GQ(I)M) (Goethert and Siviy, 2004; Park et al., 1996) , Practical Software and Systems Measurement (PSM) (DoD, 2000) and ISO/IEC 15939 (2002) are not specially adapted to small and medium-sized companies. SMEs require a better adaptation of these methodological frameworks in accordance with their small dimensions in terms of divisions, resources, projects, objectives etc. These companies require measurement programs which are easy to understand and implement, as they cannot normally use large amounts of resources in carrying them out.

In conclusion, the potential impact of measurement on the practice of industrial software development has not yet been fully exploited (Metzker, 2003), particularly in SMEs since there is the need to adapt the apparent useful measurement program methodologies for large companies to the special properties and limitations of SMEs. In addition, SMEs represent the main software industry sector. (Richardson and Wangenheim, 2007) expose the fact that SMEs represent 85% of the technological sector in China, India, Finland, Ireland, Brazil, Canada, Hungary, etc. (ESI, 2007) exposes the fact that 85% of the companies in the technological sector in Europe are of a very small setting of between 1 and 10 employees.

In accordance with the previously identified issues, the aim of the thesis is to propose a methodological framework through which to define measurement programs in the SME context and which are adapted to the measurement maturity of the company. As Pfleeger (1999b) points out, one of the main problems when defining measurement programs is that they are not adapted to the maturity of the company. This framework was called MIS-PyME and has been the main result of this research.

1.2. Hypotheses and Objectives

Our research focuses on the Measurement Program Area and, more precisely, on the Establishment of Measurement Programs for SMEs. Measurement programs are the result of organized initiatives which include selected and suitable methodologies and approaches aimed at implementing specific measurement processes in organizations. Measurement processes: describe the indicators and required measures according to the information needs, describe a set of activities aimed at collecting these measures, build and analyze the indicators, communicate results and improve the measurement process indicators. The aim of the measurement process is to obtain effective and useful information related to the measurement goals.

Hence, the hypothesis that we state for this research is:

Are the existing methodologies for defining and implementing measurement programs suitable for SMEs?

The objective of our research is, therefore:

To define a methodological framework in order to support small and medium-sized enterprises in establishing measurement programs. This methodology should be a lightweight methodology and it should take into account the maturity and the limitations of the company.

The achievement of this main objective will be based on the achievement of the following partial objectives:

- To carry out an in-depth study in the Measurement Programs Area, which is tantamount to studying the known standards and methodologies that support the definition of the measurement programs and to study the approaches and tools in bibliography which complement the standards and models indicated above.
- To study the bibliography describing experiences in the implementation of measurement programs in companies so as to discover how the measurement program was implemented and what the results were.
- To study the suitability of the above standards and methodologies for supporting SMEs in the definition of measurement programs.
- To thoroughly develop the above proposal and to build a methodological framework with which to support SMEs in the definition of software indicators according to their maturity and limitations.
- To validate the methodological framework in an SME.

1.3. Framework

This section presents the research group in which the author developed the thesis and the sponsored research projects.

1.3.1. Sponsored Research Projects

This research has principally been developed under the COMPETISOFT project financed by CYTED. This project is carried out by universities, government agencies and SMEs from many

Latin American countries. The objective of COMPETISOFT is to create a software process model focused on Latin American SMEs and to establish it as a standard for the quality certification in these countries. The software process model developed by COMPETISOFT is based on the Mexican MoProSoft model (Oktaba, 2005), which has already been established as a software process standard in Mexico.

The intention is to integrate MIS-PyME into COMPETISOFT as the software process references model. One of the companies participating in COMPETISOFT is Sistemas Técnicos de Loterías del Estado (STL), which is the company where the author of this thesis is employed. The research results and follow-up were verified and validated in this company.

Title:	**COMPETISOFT** (Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica)
Sponsored by:	**CYTED (Ciencia y Tecnología para el Desarrollo) -** 506PI0287
Grant (Euros):	100000.00 €
Participants:	Castilla-La Mancha University and 20 Universities of 12 different foreign countries. (http://www.cyted.org/Menu5/ProyectoConsul.asp?CodProyecto=291)
Duration:	January 2006 - December 2009
Main Researcher:	Dr. Mario Piattini

Table 1-1. COMPETISOFT Research Project

INGENIO (Aplicación de buenas prácticas de Ingeniería del Software para la Mejora de los Procesos de Negocio) is another research project under which the thesis was developed. This research project is framed in the business process improvement area. Its goals are to propose methodological approaches with which to support the effective, adaptable and flexible definition of business processes in order for them to be suitable for unsettled business environment and to propose technological mechanisms with which to automate the organizational activities and the communication between the Information Systems.

Title:	INGENIO (Aplicación de buenas prácticas de Ingeniería del Software para la Mejora de los Procesos de Negocio de la Pequeña y Mediana Industria del Software de Iberoamérica)
Sponsored by:	Junta de Comunidades de Castilla-La Mancha. Consejería de Educación y Ciencia. - PAC08-0154-9262
Grant (Euros):	130000.00 €
Participants:	Castilla-La Mancha University and Politécnica de Valencia University
Duration:	January/2008 - December/ 2010
Main Researcher:	Dr.. Félix Óscar García

Table 1-2. INGENIO Research Project

The last research project sponsor of this thesis is ESFINGE. The goals of this project are as follows:

1. To develop measures and indicators for different architectures and models in different abstraction levels including the threshold values and the tools for their automatic calculation.
2. To define an environment for the reengineering and system evolution in software factories based on the MDSD approach.
3. To define a testing environment based on meta-models.
4. To develop an environment for business process model improvement and evolution.
5. To validate lightweight practices for software factories.

Title:	**ESFINGE** (Evolución de Software Factories mediante INGeniería del software Empírica)
Sponsored by:	**Ministerio de Educación y Ciencia - TIN2006-15175-C05-05/**
Grant (Euros):	340800.00 €
Participants:	Castilla-La Mancha, Universidad Politécnica de Valencia, Universidad de Murcia, Politécnica de Cartagena Universities and European Software Institute.
Duration:	May 2006 - December 2008
Main Researcher:	Dr.. **Mario Piattini**

Table 1-3. ESFINGE Research Project

1.4. Document Structure

This document is structured in six chapters and seven appendixes. The contents of the remaining document are as follows.

- **Chapter 2. Research Method**. The research method which was used to achieve the goals defined for this thesis is presented.
- **Chapter 3. State of the Art.** This chapter describes the methodologies and approaches aimed to carry out measurement programs, and their suitability within the context of SMEs. An in-depth study is also performed for the models, which aims to assess software measurement. An understanding of the contribution of this thesis will be achieved after reading this chapter.
- **Chapter 4. MIS-PyME**. This chapter provides a full explanation of the basis of MIS-PyME, the methodology, the measurement maturity model and MIS-PyME contributions.
- **Chapter 5. Case Study and Experience**. The evolution of the research and a set of case studies are presented in this Chapter in order to qualitatively demonstrate whether MIS-PyME fulfils its goals of defining measurement programs which are suitable for SMEs and which are adapted to the measurement maturity of the company. The case studies were performed jointly and were simultaneous with the implementation of a measurement program in the software

department of a medium-sized company. The results, benefits and lessons learnt from the experience are also explained.

- **Chapter 6. Conclusions**: This chapter presents the main contributions of this work, the results obtained and future research.
- **Appendixes.**
    - **A: COMPETISOFT**
    - **B: MIS-PyME Measurement Goals Table**
    - **C: MIS-PyME Indicator Templates**
    - **D: Measurement Process**
    - **E: MIS-PyME 3M Assesment Process and Questionnaire**
    - **F: MIS-PyME 3M and ISO 15504 Conformity**
    - **G: MIS-PyME and other Methodologies**
- **Annexes:**
    - **Acronyms**
    - **References**

2. Research Method

This chapter presents the research methods which were used to achieve the goals defined in this thesis. The methods applied were Action-Research and Case Study.

2.1. Software Engineering Research Methods

Understanding a subject implies learning, in other words, observing and thinking; it involves knowledge classification, model-building (for application domains, problem-solving processes, etc.), experimentation, and life model evolution (Basili, 2000). This paradigm has been used in medicine, physics, and industry. These areas have some differences which lie in how models are analysed, built and how experiments are carried out. There are different types of research methodologies (Runeson and Höst, 2009):

- Surveys, which are the "collection of standardized information from a specific population, or sample, usually, but not necessarily by means of a questionnaire or interview" (Robson, 2002)
- Experiments, or controlled experiments, which are characterized by "measuring the effects of manipulating one variable on another variable" (Robson, 2002) and in which "subjects are randomly assigned to treatments" (Wallace et al., 2002). Quasi-experiments are similar to controlled experiments, except that the subjects are not randomly assigned to treatments.
- Action-Research, whose purpose to "influence or change some aspect of whatever is the focus of the research" (Robson, 2002).
- Case Study: investigating contemporary phenomena in their context (Robson, 2002), to which Benbasat et al.(1987) added "information gathering from few entities (people, groups, organizations), and the lack of experimental control".

Different research methodologies serve different purposes. Robson's classification is as follows (Robson, 2002):

- Exploratory – finding out what is happening, seeking new insights and generating ideas and hypotheses for new research.
- Descriptive – portraying a situation or phenomenon
- Explanatory – seeking an explanation or a situation or a problem, mostly but not necessary in the form of a causal relationship.
- Improving – trying to improve a certain aspect of the studied phenomenon.

The IEEE Computer Society defines software engineering as (IEEE, 1990):
"(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
(2) The study of approaches as in (1)."

Most of the empirical research carried out in this area is of an experimental and quantitative nature, based on statistics techniques: (Moher and Schneider, 1981; Basili and Weiss, 1984). Wohlin et al. (2000) published the first methodology handbook, and this was promoted by Tichy (1998). However, in recent years qualitative research methods and particularly Action-Research methods (Avison et al., 1999) have deserved the attention and acceptance of the research community (Seaman, 1999). The analytical research paradigm is not sufficient to investigate complex real life issues, involving humans and their interactions with technology (Runeson and Höst, 2009).The first proposal was published by Wood-Harper (1985).

The area of software engineering involves development, operation, and maintenance of software and related artifacts (Jedlitschka and Pfahl, 2005). Research on software engineering is, to a large extent, aimed at investigating how this development, operation, and maintenance are conducted by software engineers and other stakeholders under different conditions. Software development is carried out by individuals, groups and organizations, and social and political questions are of importance for this development (Runeson and Höst, 2009). That is, software engineering is a multidisciplinary area and therefore methods used in social sciences, such as Action-Research and case studies, are normally conducted. Action-Research and a case study will be the research methods used for this thesis. These methods are explained in greater detail in the following sections.

2.2. Action-Research Method

Among the qualitative research models found in bibliography (most of which come from the area of social sciences), we have first focused on Action-Research since it is that which is most often used in Information Systems and Software Engineering. In fact, Action-Research does not refer to a specific research method, but rather to a set of methods of the same type which share the following properties (Baskerville, 1999):

1) Focus on action and change.
2) Focus on a problem.
3) An “organic” process model which involves systematic and interactive phases.
4) Participants’ collaboration.

Since it is not a specific method, there are many definitions of Action-Research, the most important of which are:

- According to McTaggart (1991), it is the manner in which to meet the required conditions, to learn from our own experiences and make them accessible to others.
- According to French and Bell (1996), it is the process of collecting research data by means of systematic mechanisms. The data collected refers to a current system related to an objective or system requirement; feeding the system with that data; undertaking actions by

means of alternative variables selected from the system, based on the data and the hypotheses; and evaluating the results of the actions by collecting additional data.

- According to Wadsworth (1998), it consists of the participation of all research members in studying the current problematic scenario in order to improve it or change it.

These definitions make it possible to deduce that Action-Research has two aims: to benefit the research "client" and to increase the research knowledge (Kock and Lau, 2001). Hence, Action-Research is a collaborative type of research which seeks to make theory and practice meet, to establish a link between research and practice by means of a cyclical process. Action- Research is focused on yielding new knowledge which is useful in practice and which is gained by introducing changes and researching into candidate solutions to different real scenarios which are relevant to a group in practice (Avison et al., 1999). This is achieved thanks to the intervention of a researcher in the real circumstances surrounding the group. The results of these experiences must be beneficial to both the researcher and the participants. A fundamental premise regarding this kind of research is the complexity of social processes (and the use of information technologies in this type of organizations), which can be better studied by making changes to those processes and observing the effects of the changes (Baskerville, 1999).

In the area of Information Systems, the client is generally an organization to which the researcher provides his/her services. These services may be software development or maintenance consulting, which allow researchers to access data which is relevant to the research and to receive financing (Kock and Lau, 2001).

However, those researchers that might use Action-Research in Software Engineering (SE AR) will always be serving two masters: The client and the Software Engineering scientific community. Their needs are usually quite different and sometimes opposed. Attempting to satisfy both of them is the main challenge that all SE AR researchers have to face. This, however, brings with it a number of new elements to the research which adds to its interest.

Wadsworth (1998) specifies the roles of Action-Research as follows:

- Researcher: The person or group of people who actively carry out the research process.
- Researched object: The problem to solve.
- Critical reference group: A group on which research is performed inasmuch as it has a problem that needs to be solved. Additionally, this group takes part in the research process but not as actively as the researcher. In the critical reference group there are people who know that they are participating in the research and others who do not.
- The stakeholder: Anyone that can benefit from the research but does not directly participate in it. Stakeholders may include organizations that are using a new method to solve IT problems or experts who apply those methods.

Since its origins, different ways of applying Action-Research methodology have been developed (Chein et al., 1948). French and Bell (1996) propose four variations which basically depend on the characteristics of the research project:

- Diagnosis: The researcher comes up against a difficult situation; s/he diagnoses it and gives recommendations to the critical reference group, but without afterwards controlling the effects.
- Collaborative: The critical reference group puts in place the recommendations made by the researcher, and informs him or her of the results and effects.
- Empirical: The critical reference group carries out broad and systematic research into the situations and effects. This characteristic makes this variant difficult to implement.
- Experimental: This consists of evaluating the different options that exist to achieve an objective. The main disadvantage is that the different options are difficult to measure since they will be generally applied either in different organizations (with different characteristics, which may cloud the research results) or in one organization but at different moments (therefore the work environment could have changed).

A research process which uses Action-Research consists of a set of activities making up a characteristic cycle. Padak and Padak (1994) identify four steps, which have to be followed in research when using this method.

- Planning: This identifies the main issues which will guide the research. They should be directly related to the object of study and be solvable. This activity consists of identifying new alternative research ways, lines to follow, or the reinforcement of something. The result is that other problems or issues to be solved are clearly identified. Some authors (Baskerville, 1997) distinguish between diagnosis (identifying initial problems) and planning (specifying actions to solve those problems).
- Action: Controlled, careful and deliberate variation of the practice. A simulation or test of the situation is carried out.
- Observation: Collecting information and data, documenting what happens. This information may come from different sources (bibliography, measures, results of the tests, interviews, etc.). It is also known as evaluation.
- Reflection: Sharing and analysing the results with the relevant people, putting new interesting questions up for discussion (Wadsworth, 1998). Carefully studying the area that is being researched, in order to yield new knowledge that could involve improvements, by making changes as part of the research process and studying what happens after those changes. This phase is also known as "learning specification". In some Action-Research variants, it is not really a single phase, but rather an on-going process.

Given these characteristics, the process defined as Action-Research could be said to be iterative inasmuch as it moves forward to solutions which are more refined by means of a cycle complexion. In each cycle new ideas are proposed which are tested in the next cycle, as is shown in Figure 2-1. This cycle identifies Action-Research as being a reflective learning process and research into solutions. Action-Research is therefore cyclical, which means that actions are evaluated and re-planned so as to follow a diagnosis and reflection.

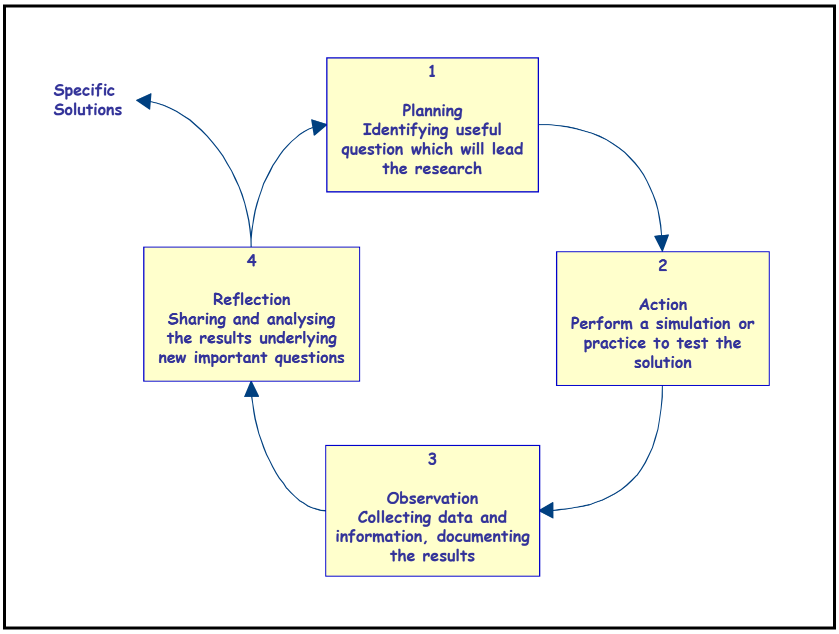


Figure 2-1: Action-Research Process

In recent years, Action-Research has been identified as being one of the qualitative research methods most frequently used in the area of Information Systems and Software Engineering research. However, the community of experts has detected some problems in its application, the causes of which are as follows:

1. Lack of SE AR methodology.
2. The consulting framework imposes an over-restrictive perspective since it implies contractual liabilities and organizational interests that could be detrimental to the research.
3. The lack of a defined research process model which indicates the steps to follow in SE AR.

All the issues indicated above may be understood to imply that this is not a rigorous research process. In order to solve these problems, the following alternatives are proposed:

- The research should be carried out by using a project management perspective;
- The inclusion of quality criteria which are specially defined;
- An analysis of the factors that lead to the formalization of a process; and
- The process should be organised by means of a project structure.

Estay and Pastor (2000a, 2000b) have combined some of these ideas to suggest using project management in order to improve the Information Systems (and SE) AR project formality. This means that a project structure which includes the main Information Systems AR elements should be used. In order to achieve the above, these authors indicate the need to adopt management practices which are suitable for Information Systems AR based on the PMBOK (*Guide to the Project Management Body of Knowledge*) (PMI, 2000), an internationally recognized management model.

According to Estay and Pastor, the concepts of Action-Research and Project are equivalent: they are both unique work experiences; their final results are also unique, and they share the notion of intervention, which means that both of them involve a voluntary change in reality. Action-Research intervention leads to changes in the work practice, but it is also a means to attain hands on experience data which are needed for the research process. The same authors have proposed a capability model based on the Capability Maturity Model (CMM) (SEI, 1995) which applies incremental management practices with the aim of guaranteeing rigour and quality improvement with regard to the use of Action-Research in Information Systems (Estay and Pastor, 2001).

In this Information Systems (and SE) qualitative research, it can be considered that there are two spheres (scientific and academic) which interact with each other in spite of their moving along different paths. IS/SE AR has a dual aim and is based on two Action-Research cycle types for two types of projects:

- Cycles aimed at solving problems within the Information Systems (and SE) projects. These projects consist of the development of a computing solution (whether they are computing, software development, information systems implementation or maintenance projects, etc.). In this case, the researcher is in charge of solving a problem.
- Cycles focused on research. These projects are related efforts to seek a result. In this case, Action-Research offers a work methodology and a justification through which to get closer to a specific reality in order to test a theory or hypothesis.

The IS (and SE) AR project structure proposed by Estay and Pastor (2000b) defines two specific cycles:

- A cycle aimed at creating a solution which provides new useful knowledge for practitioners and helps them to improve their practices. Researchers connect to reality by means of an intervention. Research is used to build models and theories influenced by reality, and to gain knowledge. This research cycle is focused on solving problems.
- A cycle focused on managing research in order to yield new knowledge to the Information Systems (and SE) subject and to improve the researchers' practices. This research interest cycle is focused on the research interest.

In short, IS (SE) AR can be analysed in two complementary manners (Figure 2-2).

- Vertically, based on the type of project.
- Horizontally, based on the typical bi-cycle structure of an IS AR project.

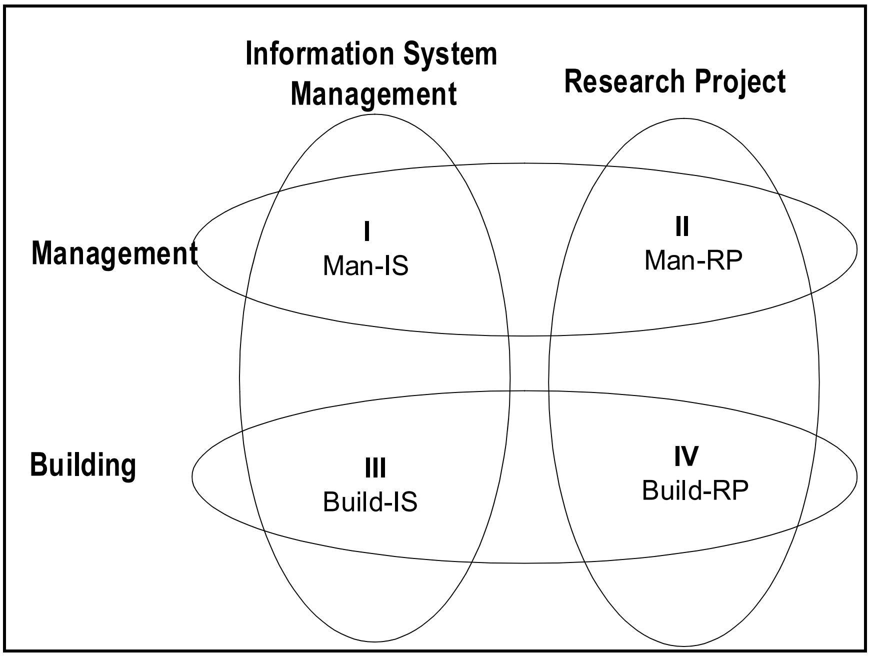


Figure 2-2: Two Dimensions for Information Systems (and SE) Action-Research.

Lau (1997) provides us with an outline of the use of IS/SE AR, including several examples published by different authors regarding the analysis, design and development of Information Systems, and particularly software implementation and related processes. Baskerville (1999) gives an introduction to the use of IS/SE AR, indicating ten IS (SE) Action-Research forms and four characteristics which determine the way in which Action-Research is used. These are as follows: Process Model (iterative, reflective, linear); structure (rigorous, fluid); typical involvement (collaborative, facilitative, expert); and primary goals (organizational development, system design, scientific knowledge, training). Baskerville and Wood-Harper (1996) list seven basic strategies to achieve IS/SE AR: using the "change paradigm", establishing an agreement or formal research contract, providing a theoretical framework, planning data collecting methods, maintaining collaboration and mutual learning between the researcher and the critical reference group, providing incentives for the performance of the typical cycle interactions and looking for the generalization of solutions.

2.3. Case Study

A case Study is closely related to Action-Research. More strictly, a case study is purely observational while Action-Research is focused on and involved in the process of change. In software process improvement (Dittrich et al., 2008; Iversen et al., 2004) and technology transfer studies (Gorschek et al., 2006), the research method should be characterized as Action-

Research. However, when studying the effects of a change, e.g. in pre-and post-event studies, the method used is Case Study (Runeson and Höst, 2009).

Case Study methodology can be used for all purposes: exploratory (which is the most common purpose), descriptive, explanatory (confirmation of hypothesis studies), improving (as Action-Research). It tends mostly to be based on qualitative data, as these provide a richer and deeper description. However, a combination of qualitative and quantitative data often provides a better understanding of the studied phenomenon (Seaman, 1999), i.e. what is sometimes called "mixed methods" (Robson, 2002).

In these cases, observation should be applied as needed and throughout the study, and not, therefore, at the end of it, or within a certain frequency or incidence. Case Studies are also conducted in real world settings, and have a high degree of realism, mostly at the expense of the level of control (Brereton et al., 2008). Case studies do not generate the same results in, e.g., causal relationships as controlled experiments do, but they provide deeper understanding of the phenomena under study (Runeson and Höst, 2009).

As they are different from analytical and controlled empirical studies, Case Studies have been criticized for being of less value, impossible to generalize from, biased by researchers etc. (Runeson and Höst, 2009). This criticism is quite comprehensive as it is a rather difficult method to apply rigorously. Zannier et al. (2006) reviewed 63 papers which had been randomly selected from 1227 papers published in 29 years of ICSE proceedings and concluded that Case Studies are usually performed incorrectly, they lack hypothesis and/or a real world case. However, some studies have been added to bibliography in order to support research in carrying out rigorous Case Studies. A book describing how to design Case Studies correctly was published by Yin (2002); Höst and Runeson (2009) have proposed practices and checklists for undertaking and reviewing Case Studies, and Brereton et al. (2008) have proposed a generic template in order to help Case Study researchers to construct a Case Study protocol, which is based on Yin's work.

The Case Study process steps are (Yin, 2002):

- Case Study design and planning
- Preparation of data collection, procedures and protocols for data collection are defined
- Collecting evidence
- Analysis of collected data and
- Reporting.

When the aim of the Case Study is to develop theories, as is the view of Eisenhardt (1989), the hypotheses are not performed during the design phase but at the end of the procedure: after analysing the data, the hypotheses are shaped and these hypothesis are then compared with the existing similar and conflicting literature. The following sections summarize the steps of a Case Study as indicated by Yin (2002).

2.3.1. Case Study Design and Planning

For Case Studies, five components of a research design are especially important:

1. Case Study questions: Case Studies try to answer the question "what can be learned?" when the goal is to identify hypotheses and propositions for further inquiries (as with any other exploratory research), but they also attempt to answer the questions "how" and "why", when the purpose is particularly explanatory. When the questions "who" "where" "how many" or "how much" are probable, other research methods such as surveys are more suitable.
2. Its propositions, if any: the proposition directs attention towards what should be studied. Without propositions the researcher may wish to cover everything, which is impossible.
3. Its unit(s) of analysis: This aims to define what the case is. In sociological case studies the case will probably be the individual. A unit of analysis should be derived from the questions in the case studies. Several units of analysis may exist if the case is composed of many units. For example if the organization of a hospital is the case under study, there might be other subunits that should be analysed such as clinical services, staff employed etc. Case Studies are not selected as logic samples as is done for surveys, these are specifically and carefully selected, and comply with certain conditions and characteristics.
4. The logic linking the data to the propositions.
5. The criteria for interpreting the findings.

Another important step for the Case Study design is the theory development, which should be carried out whether the ensuing Case Study's purpose is to develop or to test theory. Moreover, a complete research design already embodies a theory. The aim of the theory development is the understanding of what is being studied, why the study has been proposed, and what the researcher hopes to learn as a result of the study. Another important benefit of the theory development is that it is the main vehicle for the generalization of the results. Case Studies cannot be generalized as statistical generalization since the cases are not sample units, but are specifically chosen. The mode of generalization of Case Studies is "analytical generalization" in which a previously developed theory is used as a template with which to compare the empirical results of the Case Study. Empirical results are more potent if two or more cases are shown to support the same theory but do not support any equally plausible rival theory.

A variant of this methodology exists in which there are many cases under study. Multiple Case Studies are usually more robust. However, it is not always possible to perform them since the case may be very rare or may take up too many resources. Each of the cases selected for the multiple-Case Study may seek the replication of results or a contrasting replication; therefore cases are not selected as a logic sample. In addition, each of the multiple-cases may have many units of analyses. However, the analyses are not performed across the unit of analyses. On the other hand, each of these units of analyses is analysed in its case and context, and the results of the analyses for each case are later cross-analysed.

Good Case Study designs must ensure:

- Construct validity: establishing correct operational measures for the concepts being studied: This is performed during the data collection and composition phase and by means of:
  - Using multiple sources of evidence
  - Establishing a chain of evidence
  - Having key informants review the Case Study report
- Internal validity (for explanatory and causal studies only): Establishing a causal relationship whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships. Mechanisms through which to achieve this condition are performed during the data analysis phase and are as follows:
  - Pattern-matching
  - Explanation-building
  - Addressing rival explanations
  - Using logic models.
- External validity: establishing the domain to which a study's findings can be generalized. Mechanisms which help to fulfil these conditions are performed during the research design phase, and are as follows:
  - Using theory in single Case Studies
  - Using replication logic in Multiple Case Studies
- Reliability: demonstrating the operations of a study - such as the fact that the data collection procedures can be repeated with the same results. This is performed during the data collection and composition phase by means of:
  - Using Case Study protocol
  - Developing a Case Study database

2.3.2. Preparing Data Collection

The first step is to understand and gain the skills of a good Case Study researcher which are as follows:

- A good Case Study investigator should be able to ask good questions and interpret the answers.

- An investigator should be a good listener and not be trapped by his or her own ideologies or preconceptions.
- An investigator should be adaptable and flexible, so that newly encountered situations can be seen as opportunities not threats
- An investigator must have a firm grasp of the issues being studied, whether they are of theoretical or policy orientation, even in an exploratory mode. Such a grasp reduces the relevant events and information to be sought to manageable proportions.
- A person should be unbiased by preconceived notions, including those derived from theory. Thus, a person should be sensitive and responsive to contradictory evidence.

The next step would be Case Study training for the researchers who must have an in-depth knowledge of Case Study methodology in addition to an in-depth knowledge of the theory and the case to be studied.

The Case Study protocol is now defined. This contains the questionnaire for collecting data and the procedures and rules to be followed when using the protocol. The main parts of a protocol are as follows:

- An overview of the Case Study project (project objective, case study issues, and relevant readings of the topic being investigated).
- Field procedures (presentation of credentials, access to the Case Study "sites", general sources of information, procedural remainders)
- Case Study questions (the specific questions that the Case Study investigator must bear in mind when collecting data, "table shells" for specific arrays of data, and the potential sources of information for answering each question).
- A guide for the Case Study report (outline, format of the data, use and presentation of other documentation, and bibliographical information).

The following step will be to identify the case to study among all the possibilities; sometimes a mini-case study may be performed in order to select that which is most suitable. And finally, if necessary, to carry out a pilot Case Study whose aim is to improve the questions planned for the Case Study, the collection protocol or even the design.

2.3.3. Conducting Case Studies: Collecting the Evidence

Data Case Studies may originate from many sources of evidence. Six of these are important:

- Archival records: service records, e.g.: the number of clients served over a given period of time; organizational records, lists of names, survey data such as data previously collected about a site, personnel records such as calendars, etc.

- Interviews: These are one of the most important sources of information for a Case Study. They should be carried out with care, the researcher should follow the questions and inquiries planned in the collection protocol and ask the person interviewed the appropriate questions in order to obtain the information. In addition, the researcher should pose the questions in a friendly manner.
- Direct observation: This occurs when a visit to or a measure of the site takes place.
- Participant observation: this occurs when the researcher assumes a role within the Case Study. This may prove advantageous when attempting to obtain certain information which it would be impossible to obtain otherwise. However, its major problem is the potential for bias produced.
- Physical artefacts: technological devices, tools, instruments, etc.
- Documentation.

In addition, there are three principles for Case Studies data collection:

- The use of multiple sources of evidence: evidence from two or more sources, but converging on the same set of facts and findings. This is an especially important Case Study methodology for triangulation analyses, and makes the conclusions of the research more robust.
- A Case Study database: a formal assembly of evidence which is distinct from the final Case Study report should be established in order to store the Case Study notes, the Case Study documents such as the references, the tabular material such as the surveys, the narratives, etc.
- A chain of evidence: explicit links between the questions asked, the data collected and the conclusions drawn must be rigorously established.

2.3.4. Analysing Case Study Evidence

Data analysis consists of examining, categorizing, tabulating, testing or recombining both quantitative and qualitative data to address the initial proposition of the study. Some strategies through which to analyse data that use certain specific techniques exist. However, the researcher must first present the data and then the results of the analyses.

It is very important to define an analysing strategy before data collection preparation in order to ensure that the data will be analysable and to use the tools in a correct or useful manner. Without a strategy, it is difficult to obtain to a fair interpretation of data. There are three general strategies:

- Relying on theoretical Strategies: The first and preferred strategy is to follow the theoretical proposition that led to the Case Study. Proposition should have previously shaped the Case Study's objective, design and data collection plan.
- Thinking about a rival explanation: the aim is to define and test rival explanations. This strategy can be related to the first, in that the original theoretical propositions might have included rival hypothesis and the data collection may have also been guided by these rival propositions.
- Developing a case description: developing a descriptive framework with which to organize the Case Study and attempting to and identifying the appropriate causal links to be analysed, even quantitatively. This type of analysis strategies is more common when the original purpose of the Case Study has been descriptive.

Several analytic techniques can be used in any of these strategies and help to provide internal and external validity to the Case Study: pattern matching, explanation building, time-series analysis, logic models, cross-case synthesis.

2.3.5. Reporting Case Studies.

Reporting the Case Studies means bringing the results and findings to a closure. Regardless of the form of the report, similar steps underlie the Case Study composition: identifying the audience for the report, developing the compositional structure, and following certain procedures.

Finally the exemplary Case Study should fulfil the following conditions:

- The Case Study must be significant:
    - The individual case or cases are unusual and of general public interest
    - The underlying issues are rationally important, either in theoretical terms or in policy or practical terms;
    - Or both
- The Case Study must be complete: a Case Study is complete when the boundaries of the case, the distinction between the phenomenon being studied and its context are paid special attention. In addition, the investigator should convincingly demonstrate that s/he has spent an exhaustive effort in collecting the relevant evidence. The third condition is the absence of certain artifactual conditions. This signifies that if the study ended because resources were exhausted, the investigator ran out of time, or because she or he faced other non research constraints, the Case Study is not suitable to be an exemplary Case Study.

- A further issue that increases the quality of the Case Study is the use of alternative perspectives. The investigator may not have collected all the evidence and may have paid attention only to the evidence supporting a single point of view.
- The Case Study must also display sufficient evidence; this effectively presents the most relevant evidence, so that the reader can make an independent judgment with regard to the merits of the analysis; It should present not only the evidence that supports the investigator's conclusions, but also the challenging data. In addition, the investigator should show the reader that all of the single cases have been treated fairly and that the cross-case conclusions have not been biased by undue attention to one or several of the entire array of cases. In addition, the researcher should briefly show that s/he has paid attention to the validity of the evidence.
- The Case Study must be composed in an engaging, enticing and seductive manner in order to make the reader comfortable with and interested in its reading.

2.4. Research Strategy in this Thesis

In this thesis the methodology which guided the research was that of Action-Research. Action-Research methodology was chosen because the research comes from a need to define and implement a measurement program in the small software development and maintenance division of Sistemas Técnicos de Loterías del Estado (STL). The director of the development and maintenance department proposed a set of software process improvement goals and requested the development of a measurement program which would support the attainment of these goals.

From this initiative, the need for a company and a research goal appeared:

1. From the company point of view: to define and implement the required measurement program.
2. From the research group: to understand whether the popular methodologies for defining and implementing measurement programs (especially GQ(I)M) were suitable for SMEs and if they were not, to define a methodological framework for defining and implementing measurement programs focused on SMEs.

The participants in this research were (see Figure 2-3): the critical reference group, which was the development and maintenance division of STL; the researchers, who were the author and the supervisors of this thesis; and the stakeholders, who are the clients of STL who are expected to benefit from the process improvements achieved by the measurement programs implemented in STL. These measurement programs are expected to be defined using the measurement framework resulting from this research. In addition, other stakeholders will be small and medium software development companies which aim to implement software measurement programs, and particularly those which apply COMPETISOFT (Oktaba et al., 2007), the software process model into which the resulting methodological framework was integrated.

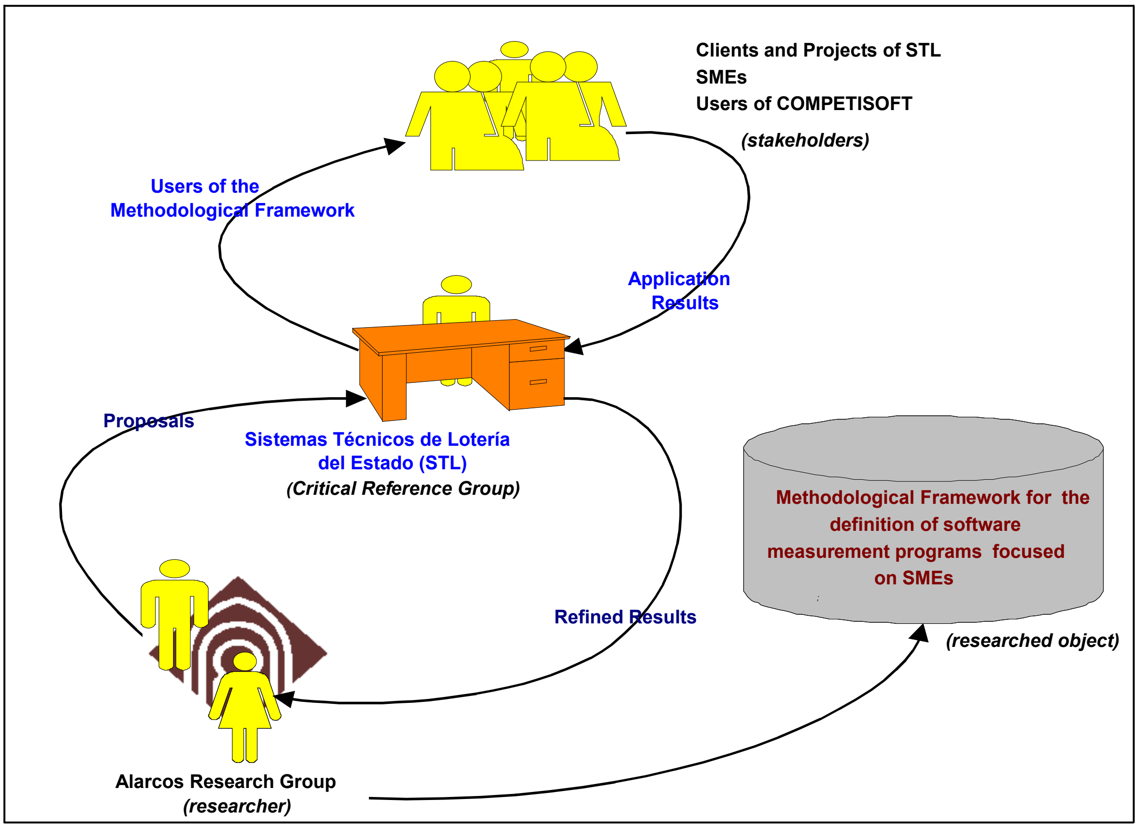


Figure 2-3. Application of Action-Research: Participants

The research group and the critical reference group collaborated in the research in the following way, according to AR cycle (see Figure 2-1): The research group and the participants in STL planned the cycle (Planning). The research group suggested actions and methods to be applied; the participants in the company applied these suggestions (Action) and communicated the results and lessons learned (Observation). The research group reflected upon the results and proposed new hypothesis (Reflection) which led to new methods and suggestions to be applied by the participants in STL (Next Cycle).

In addition, the Case Study research method was required in order to study specific issues which led to a reflection and a new hypothesis. This new hypothesis guided the next A-R cycle.

The spirit of the research was exploratory. Four research A-R cycles were necessary to carry out this research and three Case Studies were also performed. A more detailed description about how these research methods were applied can be found in Chapter 5.

3. State of the Art

This chapter analyses the methodologies and approaches which aim to define and implement software measurement programs and the models related to the determination of measurement capability maturity. These methodologies are compared according to certain requirements. This analysis is used to identify existing gaps, which provides the motivation for this thesis.

3.1. Classification of Contents.

The contents of this chapter are organized according to the software measurement classification presented in the Software Measurement KA (Key Area) proposed by Abran et al. (2004a) for SWEBOK (Software Engineering Body of Knowledge) (SWEBOK, 2008) in order to consider Software Measurement as a key area in the world of Software Engineering. In addition, we have proposed other sub-classifications, most of which were adapted from the Software Engineering Management KA and Process Measurement KA in the SWEBOK, in order to provide a clearer overview of the current state-of-the-art. These sub-classifications are as follows (see Figure 3-1):

- **Software measurement capability maturity models**. We have defined a new category in "Basic Concepts" which describes models with which to evaluate software measurement. The idea of this new category is similar to that which already exists in "Basic Concepts", called "Software Measurement Models". However, rather than describing models which support the measurement process plan, performance and evaluation, "Software measurement capability maturity models" highlight models that support the "Measurement Process Evaluation".
- **Perform Measurement Process**: we have sub-classified this section, established by Abran et al. (2004a), into the following: "Collecting Data", "Analysis" and "Feedback". "Collecting Data" means gathering the required data, verifying collected data and storing it. "Analysis" of data means transforming collected data to obtain typical indicators such as graphs, figures, etc. Data will be then analysed and initial conclusions will be drawn. These conclusions will be presented to stakeholders in the "feedback" sessions. They will be reviewed in order to verify whether they are meaningful and accurate, and whether they can result in reasonable action plans. Finally "Packing" means storing the measurement process plan, the collected data, the analysis, the conclusions and corrective actions in a feasible way in order to ease the reuse of the information for future projects and to allow the transference of information.
- Finally, we propose a further sub-classification. This has been included in the **Tools and Techniques** classification and is called "Experiences". The experience of implementing measurement programs in organizations is addressed in this category (methodologies used, challenges encountered and results).

Figure 3-1 shows the classification as suggested by Abran et al.(2004a), along with the extension to this classification that we have proposed by adding new categories which are highlighted with an oval.

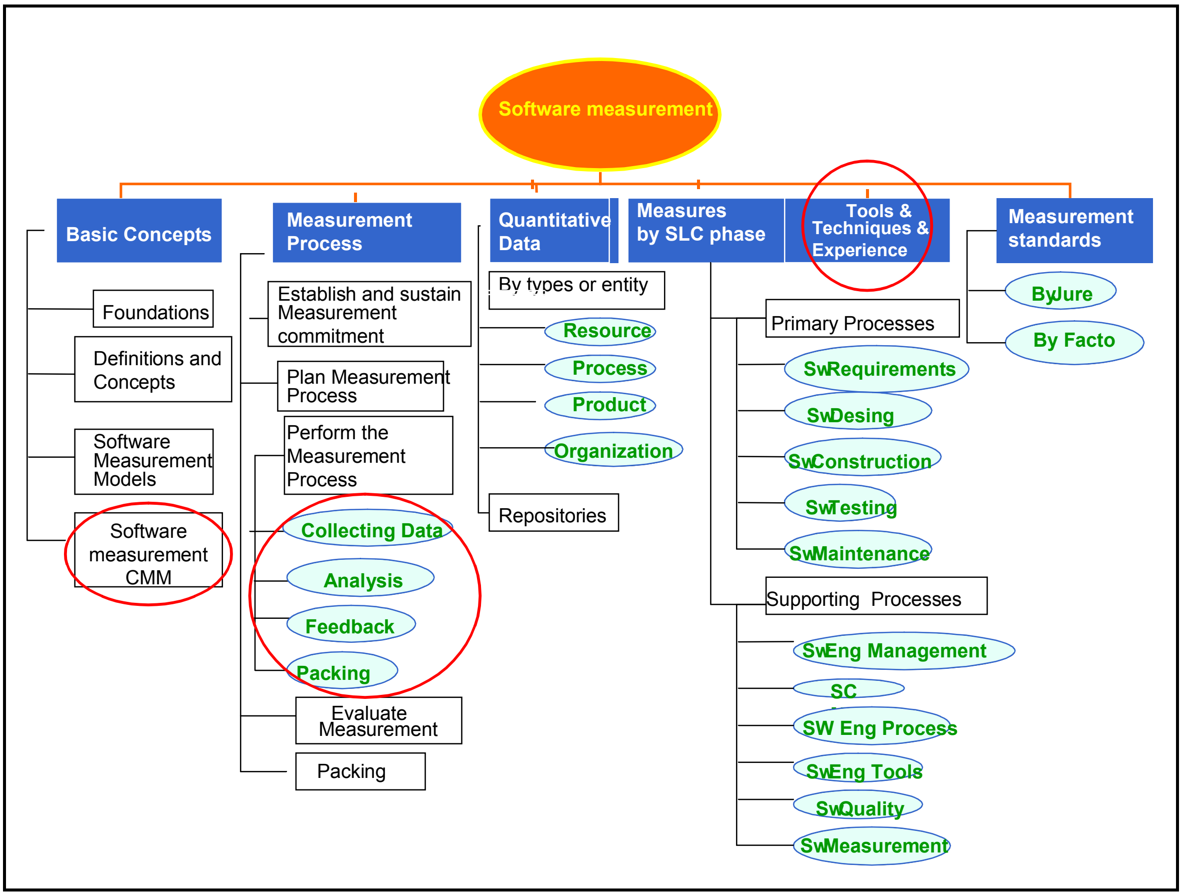


Figure 3-1. Software Measurement Classification Adapted from Abran et al. (2004a).

3.2. Basic Concepts - Software Measurement Models and Standards

This section introduces the most outstanding methodological frameworks and standards which support the definition, performance and improvement of the measurement process. We shall then go on to provide a comparison of these methodologies and to present the problems with regard to their adaptability to small and medium enterprises (SMEs).

3.2.1. Goal Question Metric (GQM)

The GQM method was first defined by Basili and Weiss (1984). The GQM approach was originally developed to evaluate defects in projects of the NASA' Software Engineering Laboratory (Bassman et al., 1995). The first evolution was the "Experience Factory", in which the concept of keeping experiences in order to reuse them was later introduced. In 1992 the

GQM process was first defined by Basili (1992), and this was extended by the definition of a structure for GQM plans (Rombach, 1991). Based on Basili et al. (1994a; 1994b) process and on the experience factory, the process was redefined by Gresse et al. (1995) during the CEMP project. Solingen and Berghout (1999) redefined the GQM process based on their experience with the CEMP project and by applying GQM at Schlumberger (Solingen and Berghout, 1999). Other improvements to the GQM method have been made by Differding (2001), Gresse (2002b) and Berander and Jönsson (2006). The GQM method establishes guidelines through which to define a measurement program: The context, objectives, and the measurement process plan. Guidelines on data collection, analysis, an interpretation of results and an identification of potential improvements are also provided.

GQM is a model which permits measurement goals to be refined into a set of quantifiable questions that are used to identify which data need to be collected to support the decision-making process. The required data provide guidance in building and selecting appropriate metrics. Software metrics are defined with their intended use in mind, along with the context in which interpretations of their values will be made (Daskalantonakis, 1992). Measured data allow us to answer the questions and to then analyse whether the goals have been attained. Thus, by using GQM, metrics, they are defined from a top-down perspective and analysed and interpreted following a bottom-up model.

The GQM method is composed of four phases: Planning, Definition, Data Collection and Interpretation (Figure 3-2) (Solingen and Berghout, 1999). The main characteristics of these phases are summarized in the following subsections.

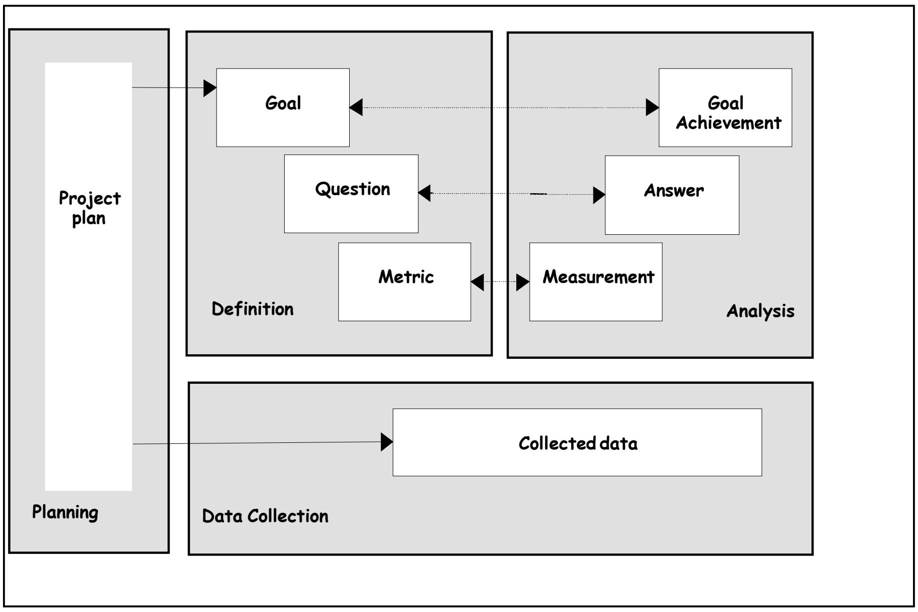


Figure 3-2. GQM Process (Solingen and Berghout, 1999)

3.2.1.1. Planning

The project plan for the measurement program is defined. All the information required to establish a commitment is defined here. The project plan is the main product delivered in this phase. The steps to perform this phase are the following: Defining the GQM team, selecting the improvement areas, selecting the project to be implemented and establishing a project team; creating the project plan, which includes an abstract and introduction of the measurement program, the schedule, the organizational structure of the project and the GQM team; a description of the management process, and the training plan for the project team members.

3.2.1.2. Definition

This phase includes the activities which are necessary to formally define the measurement program as a result of which the GQM measurement and analysis plan are obtained. The main steps in this phase are the following:

- **Establishing measurement goals**, which are formally defined by filling in the goal template provided which defines the purpose, quality focus, view point and context of the measurement goal:

- **Reviewing or defining the software process model**: This will support the measurement definition. This is carried out by the GQM team.

- **Defining the questions, hypotheses and metrics**: these include some typical interview questions on what the metrics that measure an object related to a certain goal should be or the factors that may influence this metric. After these interviews, questions and hypotheses are defined based on the knowledge of the experts in the organization. Questions are an operational refinement of the measurement goals, and for each question the expected answer is formulated as a hypothesis. These hypotheses are later compared with the measurement results. These questions and hypotheses are reviewed and metrics are then defined and checked.

- **Creating the plans**: the GQM plan, the measurement plan, and the analysis plan are created in this stage. The GQM plan serves as a guide for the interpretation of data and the development of the measurement and analysis plan. The measurement plan specifies for each metric how, by whom, and when the data are to be collected, and also includes data collection procedures and forms. The analysis plan describes how the measurement information should be processed in order for it to be easily interpreted by the project team.

3.2.1.3. Data Collection

The steps which are necessary to collect the data are classified into two main sub-groups: training and initial data collection, and construction of a Measurement Support System (MSS). The training and initial data collection aims to define, test and provide information about the collection procedures and template. Once this step has been achieved, the data collection phase can start. This phase consists of: collecting data, filling out the templates and delivering them to the GQM team, who will check data validation and package the templates for future use.

The second group aims to define a set of statistics tools, calculation spreadsheets, database systems and presentation sheets. The MSS system must support all the measurement activities related to data collection, data gathering, data processing and data packaging.

3.2.1.4. Data Interpretation

The results obtained from the metrics provide an answer to the questions defined and the achievement of the initial goal is then stated. At this stage, the feedback sessions should be prepared. The GQM members have to prepare all the material required: analysis sheets, presentations slides, and any additional material. The feedback sessions are performed; the project team should analyse the results, draw conclusions and decide what corrective actions are to be performed, and the GQM team will then write a report in which all their remarks, interpretations conclusions and improvement action areas will be indicated. This report must be delivered to the project team members and should include a report on cost-benefit performance.

3.2.2. Goal-Driven Software Measurement

In 1996, the SEI (Carnegie Mellon Software Engineering Institute) published the Goal-Driven Software Measurement guidebook (Park et al., 1996). This provides an extension to the planning phase of the GQM method, improving the way in which measures are derived from business goals and providing useful templates which help to define goals, indicators, measures, collected data, indicator’s representation, analysis sheets, etc. This extension to GQM is denominated as the Goal Question Indicator Metric GQ(I)M. The key instrument in this methodology is the indicator, which is used to link and encapsulate the information from the measurement goal to the measure. GQ(I)M provides a template which precisely defines information concerning the indicator’s (“who”, “what”, “where”, “when”, “why” and “how”).

GQ(I)M provides 10 steps to cover the definition phase of the measurement program. The first step deals directly with the identification of business goals and therefore, in contrast with GQM (Solingen and Berghout, 1999), clearly determines that the goal of the measurement program is to support a business goal. The next steps before the definition of the measurement goals provide guidelines and questions through which to derive questions, entities and attributes that will be specified in measurement goals. The next outstanding set of steps aims to define the indicators, which are the main element provided by this methodology. A summary of GQ(I)M steps is as follows:

- **Step 1 - Identifying business goals**: This step is achieved by defining the business ideas relevant to the organization. These ideas are then grouped based on their similarity. Each group is then given a name and a priority. The output of this step is a table in which the defined business goals are ordered according to their priority.
- **Step 2 - Identifying what you want to learn**. The idea is to identify what is required in order to understand, assess, predict, control or improve or motivate the elements linked to the achievement of the business goals. With that aim in mind, we should ask ourselves questions such as: “What activities do I have to manage or perform?”, and bear in mind considerations such as: “to do this I will need …” This implies a translation of the business objectives into operational statements. The questions related to the objectives are derived into entities (products or activities) and attributes (size, effort, quality) of the organization’s processes. It is highly important to at least identify the work products, activities and other entities that might be improved. The result of this step is a table in which entities and questions are matched.
- **Step 3 - Identifying sub-goals**: The questions identified in the previous point are grouped in accordance with the issues that they attempt to resolve. Based on these issues, sub-goals are formulated. The result is a list of sub-goals.
- **Step 4 - Identifying entities and attributes**: In this step, questions are used to refine the entities and attributes of the model defined in Step 2. The related entities are specified for each of the questions indicated in the previous step. Attributes are therefore listed for each of the entities which, if quantified, would help to answer the question.
- **Step 5 - Formalizing measurement goals**: In this step, business objectives are translated into measurement goals. Measurement objectives are defined, and the corresponding (entity), purpose, point of view, environment and measurement constraints are then defined.
- **Step 6 - Identifying quantifiable questions and indicators**: The questions and indicators related to each measurement goal are defined. Indicators are the result of the measurement activities and are used by directors and managers as a basis for their analysis and decisions. In (Goethert and Siviy, 2004) a template to define the indicators is described, and this is shown in Table 3-1.

Indicator Objective: The objective of the indicator
Questions: The questions related to the indicator that the user is trying to answer
Visual Display: A graphical view of the indicator
Perspective or viewpoint: The description of the audience for whom the indicator is intended.
Inputs: The list of the measures required to construct the indicator and its definitions.
Algorithms: The description of the algorithm used to construct the indicator on the basis of the measures.
Assumptions: The list of assumptions in the organization, its processes, life-cycle model, and so on, which are important conditions for collecting and using the indicator.
Data collection information: Information pertaining to how, when, how often, by whom, etc. the data elements required to construct the indicator are collected
Data reporting information: Information on who is responsible for reporting the data, to whom, and how often.
Data storage: Information on storage, retrieval, and security of the data.
Analysis and interpretation of results: Information on how to analyse and interpret as well as on how to not misinterpret the indicator.

Table 3-1. Goal-Driven Software Measurement Indicator Definition (Goethert and Siviy, 2004)

- **Step 7 - Identifying data elements**: These elements have already been identified in the indicator template. Review the elements that have to be collected in order to create the indicator and list them. The template will show a table in which the rows will be the data elements and the columns will be the indicators. For each data element, the indicators requiring that data element will be marked.

- **Step 8 - Defining measures**: In this step, the measures that have to be collected are identified in detail (see Figure 3-3). An operational definition of the measure will be provided in a checklist form. The Checklist will precisely identify which data are to be included /excluded from the measured values, along with how the data will be collected.

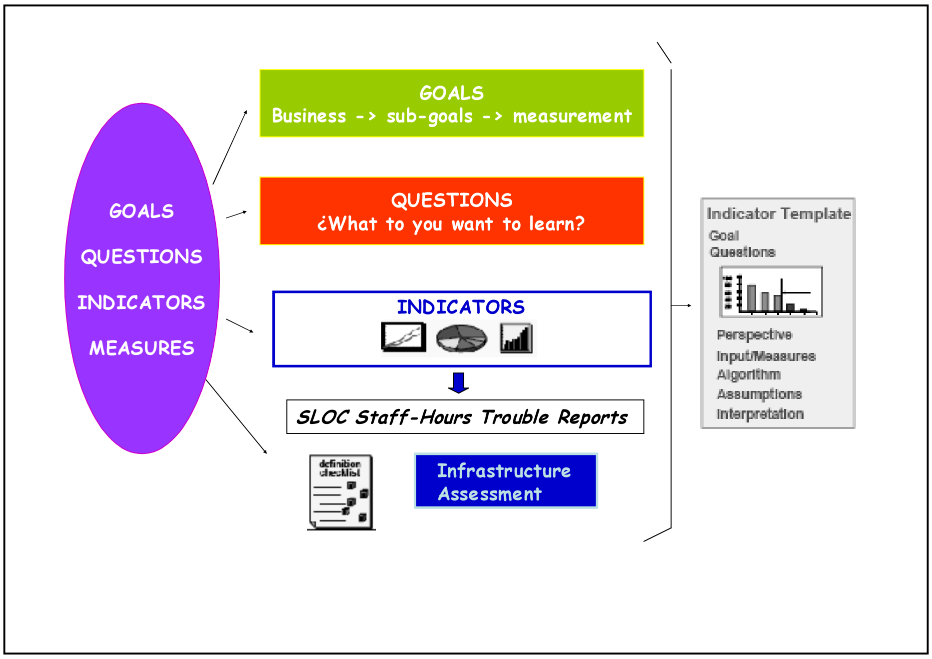


Figure 3-3. Step 8 of the Goal-Driven Software Measurement Methodology (Goethert and Siviy, 2004)

- **Step 9 – Identifying actions that must be implemented**: In this section, the ability of the organization to obtain that information will be analysed. It is necessary to: identify the source of the data; identify the data which is not available; and evaluate the effort that obtaining that data involves. The necessity for new tools, training or templates will be considered. In this section, special emphasis should be placed on measures with regard to the priority of the indicators. Finally, the following aspects should be indicated: the frequency with which and the point in the process at which this measure should be collected, the templates to record the results, the data collection and recording processes and the data analysis processes.

- **Step 10 – Preparing your measurement plan:** Once the analysis has been carried out, and the required measures and actions have been defined, an implementation plan should be drawn up to describe the specific actions that must be carried out in order to implement the measurement program in the organization.

3.2.3. GQM-lightweight method

GQM-lightweight (Gresse et al., 2003) method compresses the GQM (Gresse, 2002a; Park et al., 1996; Solingen and Berghout, 1999) approach in order to adapt this methodology to small and medium enterprises (SMEs). According to the GQM steps, GQM- lightweight adapts the GQM approach to SMEs in the following way:

- **Planning phase**.
    - As regards the GQM team, no separate measurement team will be established due to the small number of employees and informal structure of the organization. Only one person from the organization will be assigned on a part-time basis.
    - Less people should be involved in the pilot scheme and there should be less communication channels. The measurement program should be mainly supported by one or two people who are convinced of their added value.
    - Less effort should be made in the training activity. Kick-off sessions are important but they are not enough for establishing measurement programs in the organization. Motivation of the measurement program should be reconfirmed during the measurement process. Feedback sessions may help to achieve this aim.
- **Definition phase**: GQM lightweight does not propose any different practices with the exception of reusing quality and resource models (the result of a measurement program in a specific context) which will reduce the definition effort. In addition, it proposes a reduction as regards the review activity. The project team should only review data collection instruments and abstraction sheets. The GQM plan and the GQM analysis plan are only reviewed by the person responsible for the GQM program.
- **Collection phase**: GQM lightweight proposes the development of a suitable collection instrument, which is very well integrated into the development process in order to reduce effort. The measurement data can be gathered in a Database Management System (DBMS) or in SpreadSheets.
- **Interpretation phase**: GQM lightweight proposes that the intervals in which collected data is analysed and interpreted should not be too short in order to keep the effort low and not too long in order to provide feedback on time. In addition, since it is difficult to keep the collected data anonymous in SMEs, it is especially important not to use measures to evaluate people in these companies. Feedback sessions are very important but should be very well prepared in order to take up the minimum time.

Finally, GQM lightweight proposes a final phase aimed at facilitating the packaging of all the measurement program related information, the collected data, the analysis results, etc. to ease future context-reuse of this information.

3.2.4. Practical Software and Systems Measurement (PSM).

PSM (Practical Software and Systems Measurement) (DoD, 2000) is a framework which was created by the USA Department of Defence in 1994 and whose goal is to provide project and technical managers with the best Practices and guidelines in software measurement. It is based on proven measurement principles derived from actual experience in government and industry projects. Therefore, PSM should be considered as an effective management tool which does not only explain what should be done, but also how to do it, depending on the project and the organization.

The measurement process model is divided into four main activities: Measurement Tailoring (Measurement Planning); Applying Measurement (Measurement Performance); Measurement Evaluation; Compromise Establishment and Maintenance (Implementation Process) (Figure 3-4).

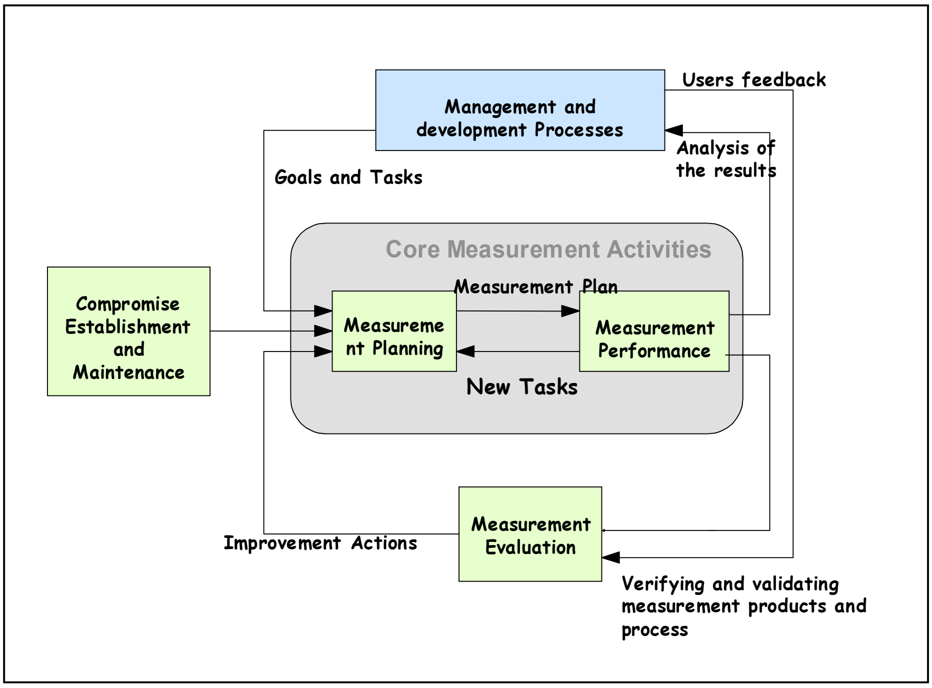


Figure3-4. PSM Methodology Context

The first two activities, Measurement Tailoring and Applying Measurement, form the core measurement process that directly serves the decision maker. The third activity, Compromise Establishment and Maintenance, includes the tasks that establish measurement processes within an organization. The fourth activity, Measurement Evaluation, identifies assessment and improvement tasks for measurement programs as a whole.

The PSM Framework provides a set of templates of measures related to typical project issues and categories in order to ease the identification of information needs and the selection of metrics. In addition, PSM provides guidelines for developing effective indicator graphs, guidelines on graph formats, and an in-depth explanation of typical indicators and analyses that can be performed with them.

3.2.5. ISO 15939: 2002- Software engineering - Software measurement process

ISO/IEC 15939 (2002) is based on PSM and identifies the activities and tasks which are necessary to successfully identify, define, select, apply and improve software measurement under a generic project or the measurement organization structure. It also provides the common measurement terminology for the Software industry. According to this standard, the main objective of the measurement process is to collect, analyse and provide relative data regarding the implemented products and processes in order to manage the processes and to objectively demonstrate the quality of a product. The software measurement process defined by ISO 159339 consists of four activities which are almost the same as PSM (Figure 3-4). The standard is defined as a set of activities which defines a set of tasks and each of these tasks specifies a set of norms (Table 3-2).

Activity	Task	Norm
Establish and sustain measure-ment. Commit-ment	Accept the requirements for measurement	The scope of measurement is identified
		Commitment of management and staff to measurement is established.
		Commitment is communicated to the organisational unit.
	Assign resources	Individuals are assigned responsibility for the measurement process within the organisational unit.
		The assigned individuals are provided with resources to plan the measurement process.
Plan the measureme nt process	Characterise organisational unit	Characteristics of the organisational unit that are relevant to selecting measures and interpreting the information products are explicitly described.
	Identify information needs	Information needs for measurement are identified
		The identified information needs are prioritised
		Information needs to be addressed are selected
		Selected information needs are documented and communicated
	Select Measures	Candidate measures that satisfy the selected information needs are identified.
		Measures are selected from the candidate measures.
		Selected measures are documented by their name, the unit of measurement, their formal definition, the method of data collection, and their link to the information needs
	Define data collection, analysis, and reporting procedures	Procedures for data collection, including storage and verification are defined.
		Procedures for data analysis and reporting of information products are defined.
		Configuration management procedures are defined.
	Define criteria for evaluating the	Criteria for evaluating information products are defined.

Activity	Task	Norm
	information products and the measurement process	Criteria for evaluating the measurement process are defined.
	Review, approve, and provide resources for measurement tasks	The results of measurement planning are reviewed and approved.
		Resources are made available for implementing the planned measurement tasks.
	Acquire and deploy supporting technologies	Available supporting technologies are evaluated and those which are appropriate are selected.
		The selected supporting technologies are acquired and deployed
Perform the measurement process	Integrate procedures	Data generation and collection are integrated into the relevant processes.
		The integrated data collection procedures are communicated to the data providers.
		Data analysis and reporting are integrated into the relevant processes.
	Collect data	Data are collected
		The collected data are stored, including any context information necessary to verify, understand, or evaluate the data.
		The collected data are verified.
	Analyse data and develop information products	The collected data are analysed.
		The data analysis results are interpreted.
		The information products are reviewed.
	Communicate results	The information products are documented
		The information products are communicated to the measurement users.
Evaluate measurement	Evaluate information products and the measurement process	The information products are evaluated against the specified evaluation criteria and conclusions on strengths and weaknesses of the information products are drawn.
		The measurement process is evaluated against the specified evaluation criteria and conclusions on strengths and weaknesses of the measurement process are drawn.
		Lessons learned from the evaluation are stored in the "Measurement Experience Base".
	Identify potential improvements	Potential improvements to the information products are identified.
		Potential improvements to the measurement process are identified.
		Potential improvements are communicated.

Table 3-2. ISO 15939 Activities

3.2.6. Comparison of the Methodologies

This section shows a comparison of the methodologies and standards described in the previous section by following the extended classification proposed in Section 3.1. At the end of this chapter, Table 3-14 shows a comparison as regards the structure of the methodological frameworks described in previous sections.

3.2.6.1. Plan the Measurement Process.

This phase includes the plan and definition of the measurement program. The differences between the studied methodologies are as follows:

- **Support Information**: PSM provides a set project issues and categories through which the information need, which is the origin of the measurement program, can be classified. With regard to these issues and categories, PSM provides a useful set of measures and indicator templates that can be almost directly implemented in the frame of a project. PSM is an excellent methodology for a company which is not mature in terms of software measurement and need to be guided to a greater extent than other more mature companies. GQ(I)M also provides a clear methodology, which includes templates, but the support of GQ(I)M is not comparable to that provided by PSM.
- **Complete methodology**: All the studied methodologies are complete, which means that they provide the necessary steps from the moment at which the measurement program is defined until the measurement program is implanted and on course. The exception is GQ(I)M, which only deals with the definition phase.
- **Organizational View**: ISO/IEC 15939 clearly focuses on establishing the measurement program which is not necessarily only for project support, but which may also support other organizational needs. GQ(I)M is focused on business and organizational needs and does not, therefore, focus only on software project support. However, GQM is more focused on defining measurement programs for a specific project, and not on supporting other organizational needs. The PSM methodology is focused on supporting projects and organizational needs. However, the PSM measure and indicator examples are clearly focused on projects.
- **Improvement and reuse of the measurement program**: PSM and ISO 15939 take into account the reuse and improvement of the measurement program. In fact, the process defined by these models is iterative: in the first iteration the measurement program is defined and implemented, and in the second iteration the same measurement program can be reused and improved in the same project or in a different project. However, the GQM, Goal-Driven Software Measurement does not take into account the reuse of the

measurement program. The GQM lightweight method does, however, specify the reuse of context-measurement models.

- **Integrating the measurement program**: Both ISO/IEC 15939 and PSM define a specific step for integrating the measurement program into the current development and project software processes, GQM and Goal-Driven Software Measurement does not pay any special attention to this.

3.2.6.2. Perform the Measurement Process

This phase deals with the collection of data, analysis of the measures and indicators, feedback of the results and packing the measurement program. The differences between the studied methodologies with regard to the activities in this phase are:

Data Collection

- GQM, specifies the "kick off" phase in which all the participants agree on the data collection activities, and are informed of the procedures, tools and templates. PSM and ISO/IEC 15939 do not clearly deal with these issues.
- On the other hand, PSM and ISO 15939 deal with how and at what frequency data should be collected and reported to the measurement analysis team. They also provide a checklist in the second step of the data collection phase, which helps to verify the data, i.e., to check whether the data matches the measurement specifications, their accuracy etc. None of these issues are dealt with in GQM.

Analysis

- GQ(I)M specifies the indicators by means of a detailed template describing the data required; how, who and when the indicator should be created; information regarding results analysis and interpretation, etc. It also gives some examples of types of indicators.

- PSM, bases the analysis phase on three types of analysis which are usually required: estimation, feasibility and performance. The indicators required are selected; for each of these. PSM defines a structured analysis model which supports the analysis and selection of indicators. Finally, PSM provides a wide variety of examples of usual indicators and analysis for each issue.

Feedback

- The Goal-Driven Software Measurement (Park et al., 1996) does not deal with this phase.
- GQM (Solingen and Berghout, 1999) recommends the performance of a cost-benefit analysis in order to demonstrate whether the measurement program is worthwhile, and it gives some guidelines on how to do this. In contrast, PSM and ISO/IEC 15939 do not.

3.2.6.3. Evaluate Measurement.

This phase deals with the evaluation of information products, measures, indicators, and analysis results, against specified evaluation criteria. It also deals with assessing the measurement process by quantitatively evaluating the performance of the measurement, conformance of the measurement process with the measurement plan, and by evaluating the process according to an external benchmark of process maturity.

PSM and ISO 15939 look at this issue and define evaluation criteria for measurement products. However, neither GQM nor GQ(I)M deals with this task.

3.2.6.4. Packing

This phase refers to the storing of the measurement program plan, collected data, analysis sheets, cost-benefit analysis and conclusions, and measurement action plans of the measurement in a feasible way in order to ease the reuse of the information in future projects and permitting information transference.

Both PSM and ISO/IEC 15939 deal with this issue. However, they are more focused on maintaining the improvements which result from the evaluation of the measurement process. The GQM lightweight method defines this step. GQM and GQ(I)M do not, however, deal with it.

3.2.7. Motivation of the Research of the Thesis

Companies usually have problems in successfully implementing measurement programs. Chapter 1 presents common causes of failure in implementing measurement programs and it has been highlighted that the tendency towards failure is particularly outstanding in the context of Small and Medium Enterprises (Gresse et al., 2003).

There are some special requirements which must be satisfied by any methodology which aims to be suited to SMEs' special restrictions. These requirements were defined after the first case study of this research, which consisted of applying GQ(I)M in order to define a measurement program in the Development division of STL (see Chapter 5). The requirements are derived from SME's restrictions and are the following:

Limited resources: SMEs do not usually have resources that can be dedicated to measurement program initiatives. Therefore, a methodology for defining and implementing measurement programs in the frame of SMEs should comply with the requirements as follows:

- **Few people involved in the process (FPP)**: Ensures that SMEs are able to follow the methodology since they cannot assign a great deal of resources to measurement initiatives.
- **Reuse measurement models (RUSE)**: Saves time and encourages companies to follow the same approach with regard to measurement in the organization, which makes organization-wide process improvement easier.
- **Few but effective and complete steps (FSTEP)**: Encourages people to start the measurement initiative since it should seem easy and effortless, and it should focus on the most important and basic measurement issues.

Poor software measurement knowledge: there is usually poor software engineering interest and knowledge, especially in the framework of SMEs. The methodology should provide support information in order to supply the lack of software measurement knowledge. The requirements are as follows:

- **Specific guidelines to support basic process improvement needs (GPIN)**: Assess the measurement goals which should help to achieve the process improvement goals and their development. This prevents users from spending too much time on defining measurement programs (effort saving) and promotes the definition of the measurement program by people who already work in the company and are not experienced in the field. Measurement usefulness can be better understood since its potential use is clearly shown when the measurement goal is derived from the software process needs.
- **Specific guidelines to integrate measurement into the software processes (GINT)**: Supports users in integrating the measurement program into the software processes, which ensures that users learn about the benefits derived from its use. This practice is essential for the continuity of the measurement program, for analysing the measurement results and making decisions and actions according to this.
- **Specific Guidelines to adapt measurement definition to the measurement maturity of the company (GMM)**: Advises the user to implement those measurement goals which suit the company's measurement maturity and prevents the users from defining measurement goals which cannot be successfully attained (effort saving). The need to contract measurement experts to develop measurement programs is thus avoided.
- **Specific Guidelines to understand the Benefits & Potential for management (GB&P)**: Ensures that users learn about measurement analysis, interpretation, decision making and their benefits. Prevents users from spending too much time on defining measurement

programs (effort saving), and facilitates the measurement programs' reliability and usefulness.

- **Measurement Examples (EXMP)**: Helps users to understand the measurement program definition result.

The aforementioned requirements were used to analyse whether the presented methodological frameworks: GQM, GQ(I)M, PSM, ISO/IEC 15939 and GQM lightweight method were suitable methodologies for SMEs. The following tables (3-3 to 3-7) show the results of this analysis and the conclusions obtained, which constitute the main motivation of this research. The tables are composed of three columns: the first column shows the requirements for a methodology aimed at defining measurement programs within the framework of SMEs; the second column indicates whether or not the methodology analysed complies with the requirement, and the values are "yes" if it complies with the corresponding requirement, "no" if it does not comply and "average" which means that the methodology partially complies with the requirement; the third column explains the requirement statement fulfilment.

As these tables show, none of the aforementioned methodological frameworks for defining and implementing measurement programs satisfy the requirements stated in this research. Therefore, based on these requirements, none of the methodologies are completely appropriate for SMEs. That which complies with most of the requirements is PSM, since it provides support modules and therefore satisfies the support requirements. However, its methodology should be adapted to SMEs with regard to the roles and number of people involved. The support information is project-oriented and there is a lack of specific measurement support for all levels of the processes established in a software development organization. Moreover, this methodology does not really guide the measurement analyst in defining the measurement program adapted to the measurement maturity of the organization.

As a conclusion of this analysis, the definition of a methodological framework for defining and implementing measurement programs which are suitable for SMEs is necessary, and this constitutes one of the main goals of this research.

GQM

REQ	Fulfilment	Explanation
FPP	No	GQM proposes the availability of an expert and an independent team to lead the measurement initiatives, which is not easy in SMEs with small budgets and small project teams. This is less likely still if the people who define the measurement program are from within the company.
RUSE	No	GQM does not take into account the reuse of measurement programs which have already been defined in the organization.
FSTEP	Average	GQM defines 11 steps, some of which could be better adapted to an SME. The planning phase could be simplified by only specifying the process improvement goals and the benefits, the project team, the measurement program validation and the calendar. The full measurement initial training could be replaced by a talk about the measurement program benefits, the goals, and the activities, people involved and calendar. The definition phase could be also simplified by defining the measurement plan at the same time as the measures are defined, and by specifying the indicators that help to answer the GQM questions, and define the analysis plan at that time. The possibility of reusing previous measurement programs makes the measurement definition more effective.
GINT	No	It does not provide specific information concerning the integration of measurement into the development, quality or management of software processes.
GPIN	No	It bases the measurement program on the process improvement goals. However, it does not provide the specific information about common measurement goals which help to achieve common process improvement goals.
GMM	No	It does not take the company's maturity into account when the measurement program is being defined
GB&P	No	It does not contain specific information about the potential of the analysis possibilities and results, and the usefulness of these analyses in process and business improvement
EXPL	Average	It provides examples but not specific examples for each common measurement need.

Table 3-3. Fulfilment of the Requirements for a Methodology aimed at Defining MP and Suitable for SMEs by GQM

GQ(I)M

REQ	Fulfilment	Explanation
FPP	No	It does not contain any information about the roles, the number of people and the profile of the people who should define and implement the measurement program
RUSE	No	It does not take into account the reuse of measurement models. It is designed to define measurement programs from scratch.
FSTEP	No	GQ(I)M defines the steps with which to define the measurement program. However, it does not provide the steps required to implement the measurement program.
GINT	No	It does not contain any information concerning the integration of measurement into the development, quality or management software processes
GPIN	No	It does not provide specific information about the common measurement goals which help to achieve common process improvement goals.
GMM	No	It does not take the company's maturity into account when the measurement program is being defined.
GB&P	Average	It incorporates clear examples to show how to use the methodology but it does not provide full guidelines to develop each common measurement goal and their management benefits
EXPL	Avarage	It provides clear examples but not complete information of common measurement goals, indicators, measures, etc.

Table 3-4. Fulfilment of the Requirements for a Methodology aimed at Defining MP and Suitable for SMEs by GQ(I)M

GQM Lightweight

REQ	Fulfilment	Explanation
FPP	Yes	This method takes into account the small number of employees in SMEs and states that no separate measurement team will be established to implement a measurement program. Only one person from the organization will be assigned on a part-time basis. In addition, less people should be involved in the pilot project and there should be less communication channels. The measurement program should be carried out by people that are convinced of its benefits.
RUSE	Yes	It includes two main activities in the process related to reuse. The first is carried out when measurement goals are defined and measurement goals are formalized, and this is called "Reuse quality and resources models". The second is performed at the end of the process and consists of packaging the context-specific measurement model in order for it to be reusable in the future.
FSTEP	Yes	It proposes 9 steps that effectively cover the entire cycle from the specification of the measurement program to its implementation. Less effort is dedicated to the training activity and the measurement program should be only reviewed by the person responsible for the GQM, with the exception of the data collection specification, which should also be reviewed by the project team. In addition, it states that less effort should be assigned to the training activities
GPINT ,GINT, GMM, GB&P, EXPL: GQM lightweight method adapts GQM and GQ(I)M to SMEs and does not include any support guidelines.		

Table 3-5. Fulfilment of the Requirements for a Methodology aimed at Defining MP and Suitable for SMEs by GQM Lightweight

PSM

REQ	Fulfilment	Explanation
FPP	Average	Parts 1.2.4. and 6.3 specify the roles defined in PSM with which to implement a measurement program. However, the number of people who play these roles are not defined. These specifications are defined in general and not for SMEs.
RUSE	Yes	Part 1.5 explains the usefulness of defining standard measurement processes for the organization which are applied to specific projects.
FSTEP	Average	The measurement process consists of four effective composite states. However, there are many parts which do not easily fit into SMEs: part 6- implement measurement process - should be simplified if it is to be applied in the SME context; parts 7.2 and 7.3- for evaluating the measurement process, etc. These activities should be adapted to the measurement maturity of the company, and they should refer to software process maturity models which are adapted to SMEs, not to those such as CMMI.
GINT	Yes	Part 2. 5.4 explains how measurement should be integrated into the technical and management processes. In a more practical manner, the measure templates define how the measure is integrated and in which activities of the technical and management processes it could be used.
GPIN	Average	Parts 2 and 3 show how measures are specified from project issues. Even if Specific-project issues are of great importance, they only cover the needs of a specific project, and do not take into account the organization-wide needs. Moreover, the model does not directly relate process improvement goals and measurement needs.
GMM	Avarage	PSM addresses information regarding the constraints on the type of analysis (e.g. part 5) or on the tools required (e.g. part 2), etc. but it does not relate these constraints to a measurement maturity model in which the limitations and evolution are clearly described (e.g. historical and analysed information is required to predict a goal in order to make an evaluation, and a high measurement maturity is necessary), and it lacks information with regard to the constraints of some of the indicators' purpose and their scope (e.g. before monitoring a project based on the work-packages, it is recommendable to gain experience by means of phase-by-phase monitoring).
GB&P	Yes	PSM provides full information for the development of the measurement goals derived from project needs. It provides full information with regard to the definition of each common issue, measurement category, and measures which develop the goal. The common analysis that can be performed, its interpretation and its benefits are also explained.
EXPL	Yes	Part 3 and 5 are specific chapters dedicated to showing and explaining common measure and indicator examples. The examples shown are related to the measurement specification templates proposed in this model.

Table 3-6. Fulfilment of the Requirements for a Methodology aimed at Defining MP and Suitable for SMEs by PSM

ISO/IEC 15939

REQ	Fulfilment	Explanation
FPP	Average	With regard to ISO/IEC 15939 (2002a), clause 5.1.2 indicates the roles required to define a measurement program. However, these are not adapted to SMEs
RUSE	Yes	The definition, implementation and improvement process take into account the reuse of standards or previously used measurement programs.
FSTEP	Yes	The measurement process consists of four effective composite states.
GINT	No	5.3.1 Indicates that measurement activities should be integrated into the other software processes. However, it does not guide how each common measure information need should be integrated
GPIN	No	It does not provide the common measurement goals which help to achieve common process improvement goals.
GMM	No	It does not take the company's maturity into account when the measurement program is being defined.
GB&P	No	It does not provide information about possible questions which may be posed to achieve each common measurement information need. It does not provide specific information for each common measurement indicator about the analyses' possibilities and the usefulness of these analyses
EXPL	No	It provides some examples of measurement programs and evaluation criteria, but not full common measurement program examples

Table 3-7. Fulfilment of the Requirements for a Methodology aimed at Defining MP and Suitable for SMEs by ISO/IEC 15939

3.3. Other Approaches

This section summarizes other approaches which complement the methodological frameworks shown above, considering whether these approaches are suitable for application in SMEs.

3.3.1. Description of the Approaches

The following subsections describe the main characteristics of the approaches by following the extended classification of Section 3.1.

3.3.1.1. Plan the Measurement Process

The approaches presented in this section are related to the definition of the measurement program goals, the indicators, the measures and the related procedures for collecting the required data.

Niessink and Vliet (1999)

Niessink and Vliet (1999) proposed a general model which highlights that the importance of the measurement program is the return value that it provides to organizations. Unlike the GQM (Basili and Rombach, 1988a) model, the objective of the measurement program is not a measurement goal but a high level business goal. Niessink and Vliet's model generally describes measurement programs from the high level business perspective (see Figure 3-5). The model uses four main steps to define how a measurement program is implemented to support companies with regard to a specific problem. Moreover, the success factors highlighted in the model are those which provide a return value to the organization and are external to the measurement program.

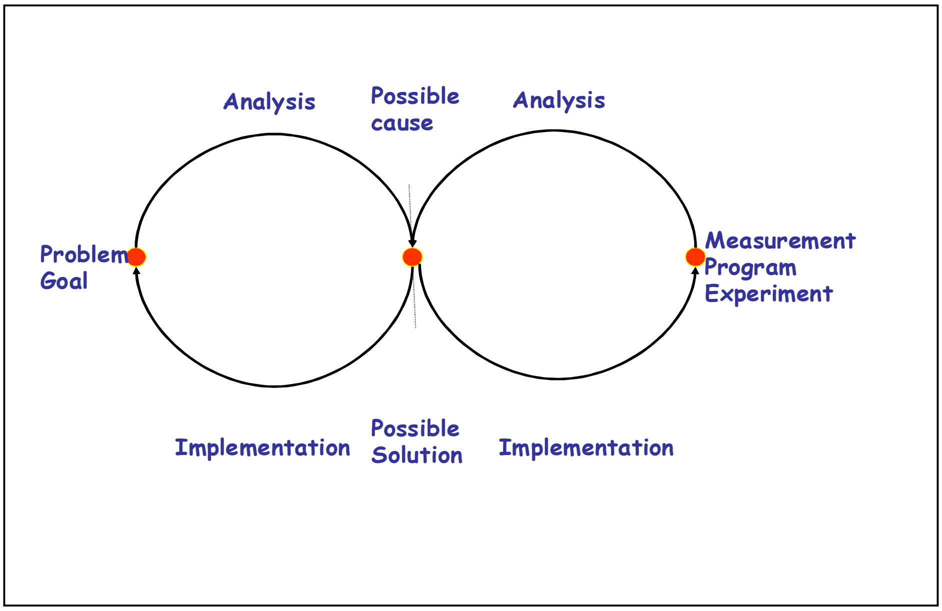


Figure 3-5. Niessink and Vliet's (1999) Approach

This idea of planning the measurement process from a high level business perspective has also been highlighted by the GQ(I)M model. The aim was to establish well defined links between the results of the metrics and the business context and development. However, taking the Niessink and Vliet's model as a reference, GQ(I)M is defined from the right-hand side viewpoint and Niessink and Vliet's model is defined from the left-hand side viewpoint.

Briand et al. (2002)

Briand et al. (2002) make a suggestion which aims to provide an accurate definition of predictive measurement programs, by defining an approach which is organized into six steps based on GQM (Solingen, 2002; Basili and Weiss, 1984; Basili and Rombach, 1988a), and by making it more formal. Predictive systems are a crucial application of measurement, but when it comes to defining measures there is a lack of theoretical basis, which makes it impossible to create an accurate prediction system and a system that accurately characterizes the object of study. Hence, most organizations are led to results that are unexpectedly negative, positive but difficult to interpret, or difficult to apply in subsequent projects.

Brijckers and Differding (1996)

Brijckers and Differding (1996) propose a technique which allows the measurement process to be defined easily and more accurately, and permits the addition of formality to the measurement process plan. It is based on the idea that modeling the entity to be measured and integrating it into the measurement process is advantageous to the measurement plan definition. This idea helps to precisely identify the object to be measured, which is especially important when GQM interviews are performed. It also helps identify who will collect each piece of data required and when it should be collected. It additionally supports the identification of data which could be reused in different goals. Finally, the data required to define a measurement program will be directly derived from the software process model. Additionally, a modeling language, the MVP-LLANGUAGE, is proposed to model resources, products, and processes in the framework of a software project, and guidelines are defined to support the definition of the measurement program plan.

Lavazza and Barresi (2005)

Lavazza and Barresi (2005) also support the former work idea, and take it further by implementing a tool which integrates the GQM plan and the model of the entity to be measured. They claim that the advantage of this idea, besides that which was indicated by Brijckers and Differding (1996), is the formality that it brings to the measurement process (if the entity to be measured is not otherwise modelled, it is then difficult to ensure that the collected data match the plan, as this knowledge lies with the people who perform the measurement plan). In addition, this would help to identify which data it is impossible to collect. GQM encourages organizations to use data that is difficult to collect. Therefore, the top-down approach definition suggested by GQM should be complemented by a bottom-up approach, consisting of assessing and exploiting available data. Additionally, the repeatability of the measurement process becomes more feasible. Finally, there are some extra advantages: the user can define a number of metrics and questions which were not taken into account previously since there are some elements in the model that are natural candidate metrics; additionally, by using this approach, the GQM plan will be packaged together with the process model being measured, which will make it more precise.

Baldassarre et al. (2003)

Baldassarre et al. (2003) define a framework known as the multiview framework which facilitates the comprehensibility, ease of use and repeatability of a measurement program.

This framework supports the identification of the measurement process complexity, particularly with regard to the interpretation of the GQM measurement plan. It highlights the idea that the measurement plan may contain some specific goals, but each of them should relate to a single view. Therefore the number of metrics will be limited, which will also reduce the overall complexity of the interpretation. The framework is useful to evaluate one success factor which is the comprehensibility of a measurement process. If the measurement process is difficult to understand, then only the person who has developed it will be able understand the process and interpret the results. Additionally it will be easier to use, which is another measurement program success factor. Both factors make the measurement process more repeatable.

Lindvall et al. (2005)

Lindvall et al. (2005) defined a pattern to represent the measurement plan definition which also complements GQM. Measurement patterns are built by creating a tree in which GQM hypotheses and questions from higher-level goals lead to lower level goals. Other related hypotheses and questions can also lead to more specific goals down to the lowest level, where goals are basic goals (characterization goals) and lead directly to metrics. The result is a tree in which the GQM definition plan is detailed.

The main advantage of the proposal is that, with measurement patterns (Figure 3-6), there is no need to start a measurement program from scratch and it is possible to identify a high-level pattern which can be applied in different contexts. In addition, measurement patterns make it possible to easily pack and transfer the measurement definition within and between organizations.

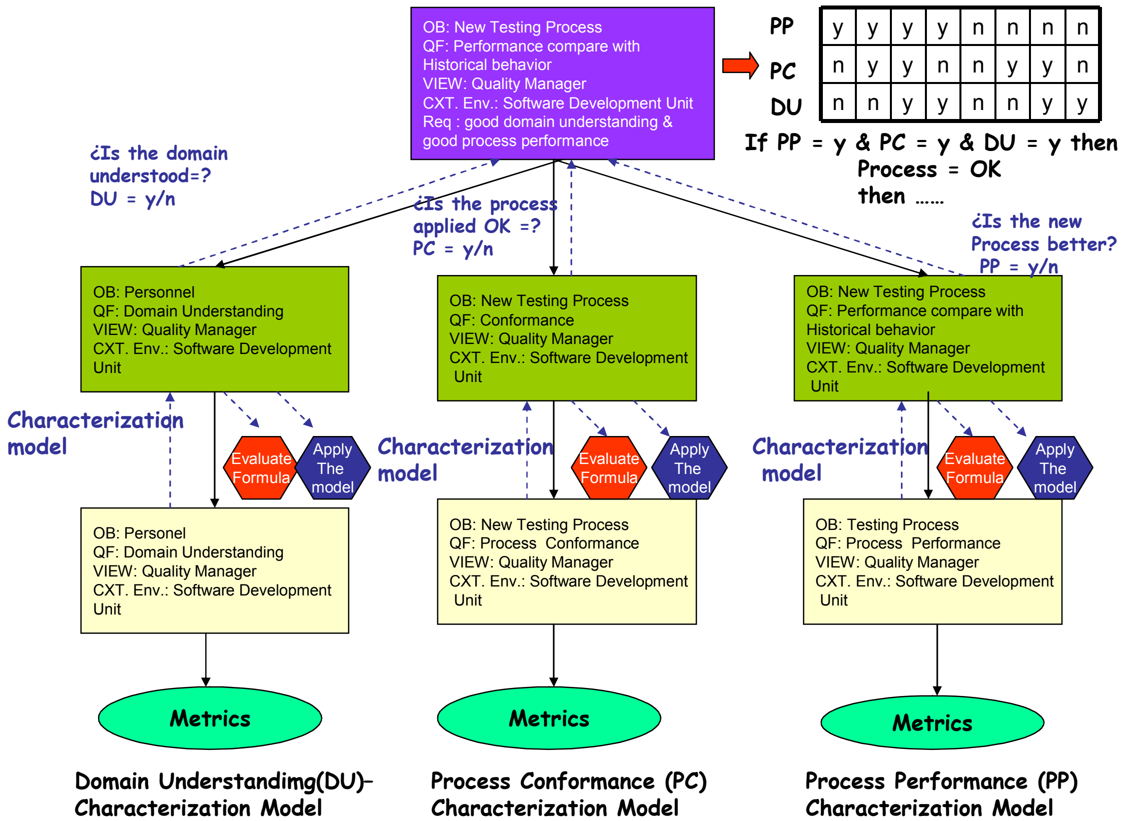


Figure 3-6. GQM Measurement Plan Pattern

Umarji and Emurian (2005)

In some situations it may be worth analysing whether the measurement program will be used in and correctly adapted to the organization. Regarding this, the *Social Impact* model defined by Umarji and Emurian (2005), predicts the success, and thus the usage of a measurement program by measuring for what purpose it is intended to be used. The Intention depends on the *Ease of Use*, *Usefulness*, *Control* and *Attitude* to use the measurement program.

Bégnoche et al. (2007)

Bégnoche et al. (2007) suggested integrating ISO/IEC 15939, Capability Maturity Model Integration (CMMI) and International Software Benchmarking Standards Group (ISBSG) to assist non measurement experience organizations to start defining their measurement programs. ISO 15939 covers all the activities and tasks necessary for a successful implementation of a measurement program. However, according to these authors this international standard is not sufficient in itself, as it does not catalogue software measures nor does it provide a recommended set of measures which can be applied to software projects. Rather, it provides guidance for “defining a suitable set of measures that address specific information needs” (ISO/IEC, 2002). CMMI (CMMI, 2006) defines goals and practices covering multiple maturity levels and multiple process areas. These goals and practices may be used to provide further guidance as to which elements of a software engineering process need measurement and to identify some of the information needs. However, neither ISO 15939 nor CMMI provides detailed data which can be of immediate use in organizations for benchmarking or guidance purposes. Such data are, however, available from the ISBSG (CMMI-Product-Team, 2006), which provides benchmarking standards based on ISO 15939, as along with a repository of over 4,000 projects, as of early 2007. The integration of these three elements is shown in Figure 3-7.

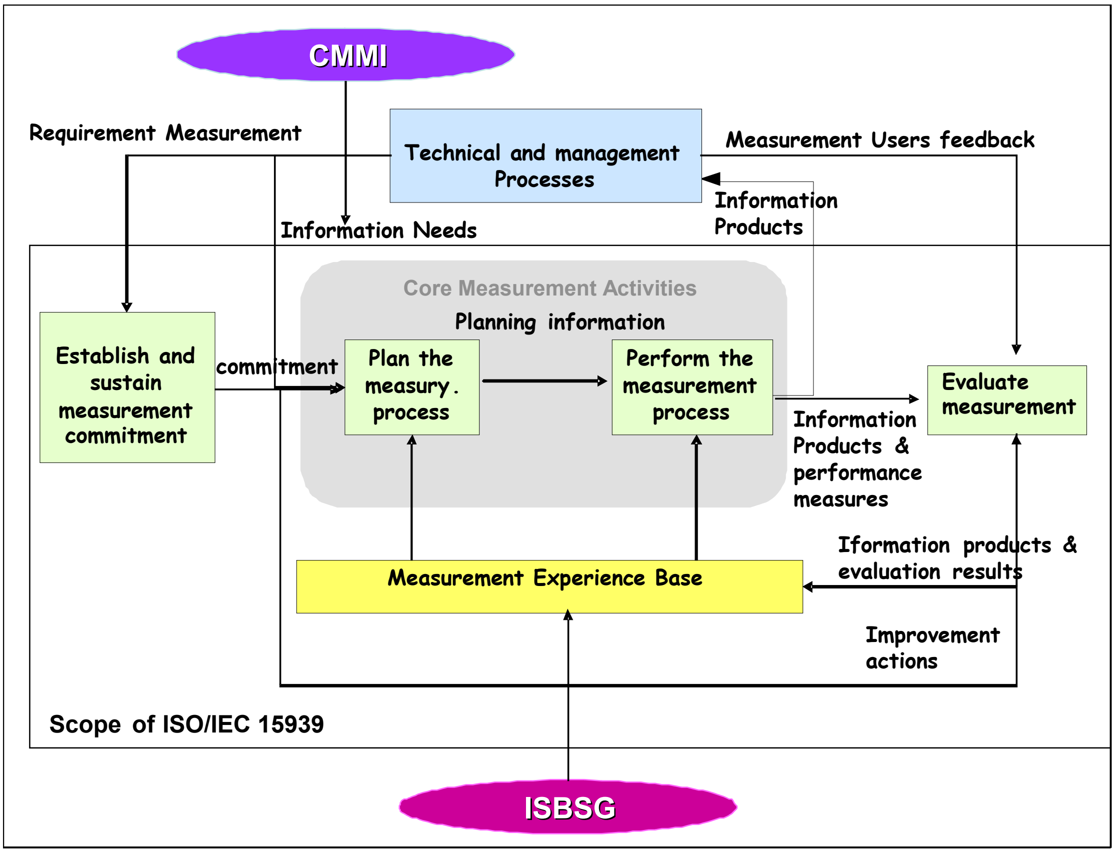


Figure 3-7. Integrating ISO/IEC 15939, CMMI and ISBSG (Bégnoche et al., 2007)

Dumke et al. (2006)

Balancing Multiple Perspectives (BMP) is a procedure designed to help project managers choose a set of project indicators from several concurrent viewpoints in order to prioritize the different viewpoint, improve the control by adding more perspectives (time, cost, quality, risk, etc.) and reduce cost by balancing the number of measurement controls across more than two perspectives. The BMP procedure consists of four steps:

1. Determine the dimensions of interest in the project: at least three dimensions – or four or five;
2. Determine the list of the most representative measures associated with each dimension;

3. For each of the measures selected, identify which other control variables might receive a negative impact (e.g. counterproductive impacts: for instance, higher quality will often mean a greater initial cost or longer project duration; the same applies to cost and risk);
4. Determine the best combination of indicators and the causal relations between them in order to build a measurement plan for the project.

Several studies have been performed in order to understand which viewpoints and measures are desired in software projects (Cuadrado-Gallego et al., 2007; Dumke et al., 2006) .

Briand et al. (1996b), Morasca (2008) and Weyuker (1988)

Axiomatic Approaches have also been proposed in order to rigorously and theoretically define internal attributes. Once these attributes are formally defined, they can be rigorously checked and the relations between the internal software attributes can be understood. Briand et al. defined an axiomatic approach, i.e., a means to discuss internal software attributes in a precise way, and evaluate the similarities and differences between them (Briand et al., 1996). They also proposed a set of axioms for a number of different attributes: size, length, complexity, cohesion and coupling. The axiom sets were defined based on a graph-based representation used to model the elements of software artefacts and the relationships among them. This axiom approach has recently been improved (Morasca, 2008), and more properties for the basic sets of axioms (size, complexity, coupling, cohesion) have been identified, and the similarities and differences between internal software attributes have been stated.

Weyuker (1988) also proposed nine properties in order to evaluate software complexity measures.

Heidrich and Münch (2008)

Specula is an approach which helps to interpret and visualize collected measurement data in a goal-oriented manner in order to effectively detect plan deviations. The methodology is based on the Improvement Paradigm (QIP) and makes use of the GQM approach to specify measurement goals. QIP is used to implement a project control feedback cycle and make use of experience and knowledge gathered in order to reuse and customize control components. GQM is used to drive the selection process of finding the right control components according to defined goals. In addition, it proposes architecture and a tool with which to build the dashboard.

In (Ciolkowski et al., 2008) the usefulness of this approach is stated. Previous project control tools have been provided: (Hendrick et al., 1992), WebME (Tesoriero and Zelkowitz, 1997) Amadeus (Selby et al., 1991), Ginger2 (Torii et al., 1999) Pentaho and Jaspersoft. However, all of these lack a fixed built-in set of control indicators and corresponding visualizations, a role-oriented approach to data interpretation and visualization, a methodological framework, etc. There are also a considerable amount of tools for supporting one specific area of the project control such as CAST's Goveranance Dashboards, Optimyth's

Checking QA tool for technical quality, Maven plug-ins for collecting white-box testing information from JUnit and Clover, or business testing (e.g. HP's Business Services Management, IBM Rational Quality Manager, etc.).

Basili et al. (2007)

In (Basili et al., 2007) an improvement of GQM, denominated as the GQM+Strategies approach, is presented. This aims to define measurement programs (based on measurement goals) which are consistent with business goals, strategies, and corresponding software goals. Strategies are formulated to deal with business goals, such as improving customer satisfaction. It is therefore determined that the software development part of the organization could best contribute by reducing the defect slippage to the customers by improving the software system testing process, which would be the software goal, and finally the measurement goal is defined: Measuring defect slippage where the traditional GQM method begins.

PSM, ISO 15939 and GQM do not explicitly link the measurement program to business goals and do not ensure that the totality of goals forms a rich set of relationships. This identification and the use of the relationships, both between different measurement goals and between measurement goals and higher-level business goals, is addressed in the extended GQM.

A recent trend in the IT governance domain and the IT service domain emphasizes the need for linking business goals and IT goals. However, the solutions proposed by models in these domains (particularly COBIT® 4.1 and ITIL release 3) offer only simple connections between predefined sets of goals and are rather focused on providing an IT infrastructure than on linking software development or software engineering with business strategies.

3.3.1.2. Perform the Measurement Process

The approaches presented in this section deal with the collection of data, analysis of the measures and indicators, feedback of the results and packing the measurement program.

Boffoli et al. (2008)

In (Boffoli et al., 2008) guidelines and a disciplined process for guiding practitioners in correctly using Software Process Control (SPC) during process monitoring are defined. This process is based on experiences in using Statistical Process Control in industrial projects (Baldassarre et al., 2004; 2005; Boffoli, 2006; Caivano, 2005). The guidelines consist of four patterns describing an SPC problem and a solution.

Solingen and Berghout (2001)

With regard to the feedback activity, Solingen and Berghout (2001) propose a model for the learning process between the software development team and the GQM team. This is based on learning theories and provides some learning enablers in order to stimulate the group and the learning sessions. The learning enablers deal, for example, with involving leadership, creating positive expectations about the measurement program, assisting the project team in data analysis measurement and improving the software process, creating an atmosphere of mutual trust, etc.

The second proposal refers to explicitly identifying whether the measurement program is actually worthwhile. Costs, and especially benefits, are explicitly defined when translated into financial benefits. It is sometimes quite easy to do this by measuring effort, efficiency benefits, and additional revenues. However, if other factors which are external to the development group have to be measured (e.g., if customer satisfaction has to be valued to understand a benefit in market terms), identifying the revenues derived from a measurement program becomes more difficult.

Birk et al. (1998)

Birk et al.(1998) suggest that the GQM method be used as a framework for addressing costs. The activities related to each stage of the GQM model are identified and their cost is calculated. With regard to benefits, they suggest addressing the following points: goal attainment, understanding of software product and process, improved communication, attention to software quality, and corporate identity.

3.3.1.3. Evaluate Measurement

The approaches presented in this Section deal with the evaluation of information products, measures, indicators, and analysis results, against specified evaluation criteria; and assessing the measurement process by quantitatively evaluating the performance of the measurement, conformance of the measurement process with the measurement plan, and by evaluating the process according to an external benchmark of process maturity

Mendonça and Basili (2000)

Mendonça and Basili (2000) presented a methodology with which to understand the data and the metrics and how they fulfil the needs of data users in a Measurement Framework. Their approach addresses the issues of data being collected for no good purpose and data being unused because its existence is unknown.

Berry and Vandenbroek (2001)

Berry and Vandenbroek (2001) present an analytical technique aimed at assessing and deriving improvements in a number of processes (e.g., CMMI Key Areas), which therefore includes the measurement processes. This is a bottom-up technique since the improvements made to the measurements and the relevant processes are derived from the individual’s satisfaction, which comes from the relationships between the performance of measures (performance of the measure according to a key process), and the importance assigned to measures; and between the performance of the key processes and the importance assigned to key processes. When performance is rated below importance, there is a performance gap that may be addressed by an improvement activity. According to Berry et al. (2004), this targeted measurement key process assessment identifies what has to be changed in a more accurate manner than does a generic model (for example, the Information Quality model based on AIQM (Lee et al.)). However, acquiring and deploying it requires more effort.

Eickelmann (2001)

Simulation techniques, such as those studied by Eickelmann (2001), could be used as a Visualization model in order to make a previous analysis of the benefits brought about through improvement, among other things. Current organization processes can be modelled, and simulation technologies could be used to determine the impact of inserting a new technology with regard to productivity, schedules, quality, reliability, and cost. Simulation technologies provide the capability to modify or add new variables, and to visualize the affected variables of the model, such as the productivity of people or computational devices, changes in the duration of activities, schedule impacts, and expected product quality attributes.

Gopal et al. (2002)

An analytical improvement technique could observe whether the measurement process matches the known success factors for a measurement program. Gopal et al. (2002) describe and prove some of the known measurement program success factors. They state that the success factors are those which use metrics for decision-making, and which are related to measurement and make an impact on the organization's performance. Technical factors will make a direct impact on the use of measures for decision-making. Some factors make the collection, analysis, and dissemination of metrics uncomplicated, thereby providing the project personnel with incentives to use them in decision-making. Organizational factors will make a direct impact on the improvements in organizational performance. These factors also ensure that the organizational culture accepts and will be built around the relevance of metrics.

Measurement success factors have, however, been identified by many authors. Some of these authors and the success factors identified are as follows, Table 3-8:

Success Factor	Authors
Do not attempt to define the best measurement program but that which is most capable, and improve it	(Basili and Rombach, 1988a; Bassman et al., 1995; Clark, 2002; Daskalantonakis, 1992; C. Dekkers and McQuaid, 2002; T. Dekkers, 2005; Goodman, 2004; Grady and Caswell, 1987; Hall and Fenton, 1997; Jeffery and Berry, 1993; Jones, 2003; Kautz, 1999; Kenett and Bake, 1999; Niessink and Vliet, 2001; Park et al., 1996; Pfleeger, 1993; Rifkin, 2001; Russac, 2002; Solingen and Berghout, 1999; Wiegers, 1997; 1999)
Motivate managers and gain commitments	(DoD, 2000; Grady and Caswell, 1987; Hall and Fenton, 1997; Jeffery and Berry, 1993; Jones, 2003; Musa et al., 1987; Niessink and Vliet, 2001; Russac, 2002; Wieczorek, 1997; Clark, 2002; Kautz, 1999; Kenett and Bake, 1999; Kitchenham, 1996; Minkiewicz, 2000; Wiegers, 1997)
Clearly define the goal of the measurement program	(DoD, 2000; Grady and Caswell, 1987; Hall and Fenton, 1997; Musa et al., 1987; Rubin, 1991; Niessink and Vliet, 2001; Park et al., 1996; Sommerville et al., 1999; Bassman et al., 1995; C. A. Dekkers and McQuaid, 2002; T. Dekkers, 2005; Goodman, 2004; Grady and Caswell, 1987; Jeffery and Berry, 1993; Jones, 2003; Kenett and Bake, 1999; Kitchenham, 1996; Musa et al., 1987; Rubin, 1987; Russac, 2002; Wieczorek, 1997; Wiegers, 1997)
Train people involved in the measurement program	(T. Dekkers, 2005; DoD, 2000; Hall and Fenton, 1997; Kautz, 1999; Niessink and Vliet, 2001; Russac, 2002; K. E. Wiegers, 1997)
Involve stakeholders in the measurement program definition and implementation	(Daskalantonakis, 1992; C. A. Dekkers and A., 2002; Goldenson et al., 1999; Gopal et al., 2002; Hall and Fenton, 1997; Jeffery and Berry, 1993; Kautz, 1999; Kenett and Bake, 1999; Niessink and Vliet, 2001; Solingen and Berghout, 1999)
Build a participatory management style involving all levels of audiences	(Jeffery and Berry, 1993)
Ensure that the level of technical difficulty of the software development process and products is within the capability of staff members	(Jeffery and Berry, 1993)
Ensure a supportive industrial climate of trust, respect, and esteem among all levels of staff with a predisposition to improvement	(C.Dekkers and McQuaid, 2002; Iversen and Mathiassen, 2003; Jeffery and Berry, 1993; Russac, 2002)
Design measurement programs in a precise manner	(DoD, 2000; Park et al., 1996; Solingen and Berghout, 1999; Wiegers, 1999; Hall and Fenton, 1997)
Provide simple, complete, and consistently documented operational definitions and process descriptions for the artefacts to be measured in a certain way	(Daskalantonakis, 1992; T. Dekkers, 2005; DeMarco, 1982; Goldenson et al., 1999; Goodman, 2004; Gopal et al., 2002; Kitchenham, 1996; Wiegers, 1997)
Provide regular feedback to people involved in the programme	(Basili and Rombach, 1988a; T. Dekkers, 2005; Hall and Fenton, 1997; Jeffery and Berry, 1993; Jones, 2003; Kitchenham, 1996; Niessink and Vliet, 2001; Pfleeger, 1993; Solingen and Berghout, 1999; Wieczorek, 1997; Wiegers, 1999)
Automate and Store the software measurement data in an organizational database and/or repository to allow for normalization	(Bassman et al., 1995; Hall and Fenton, 1997; Niessink and Vliet, 2001; Solingen and Berghout, 1999;Daskalantonakis, 1992; T. Dekkers, 2005; Gopal et al., 2005; Grady and Caswell, 1987; Jeffery and Berry, 1993; Russac, 2002)
Integrate the measurement program with the rest of the organization's	(Daskalantonakis, 1992; DeMarco, 1982; DoD, 2000; Gopal et al., 2005; Jeffery and Berry, 1993; Russac, 2002; Selby,

Success Factor	Authors
software processes	2005)
Involvement of the project managers and engineers in the analysis	(Daskalantonakis, 1992; DoD, 2000; Solingen and Berghout, 1999)
Do not use data to evaluate people	(Jeffery and Berry, 1993; Pfleeger, 1993; Solingen et al., 1997; Wiegers, 1997)
Measurement should be transparent to developers	(Clark, 2002; T. Dekkers, 2005; Goldenson et al., 1999; Gopal et al., 2002; Gopal et al., 2005; Hall and Fenton, 1997; Jeffery and Berry, 1993; Pfleeger, 1993)
Use existing metrics materials	(Basili and Rombach, 1988a; DoD, 2000; Goodman, 2004; Hall and Fenton, 1997; K.E. Wiegers, 1999)
Ensure reliability and integrity data	(Hall and Fenton, 1997; C. Dekkers and Bradley, 1999; T. Dekkers, 2005; Jeffery and Berry, 1993; Pfleeger, 1997; Russac, 2002; Umarji and Emurian, 2005)
Usefulness of metrics data	(Hall and Fenton, 1997; Jeffery and Berry, 1993; Niessink and Vliet, 1999; Pfleeger, 1993; Rubin, 1987; Russac, 2002; Wiegers, 1997)
Involve metric enthusiastic people in the measurement program definition and implementation	(Daskalantonakis, 1992; Hall and Fenton, 1997; Jeffery and Berry, 1993; Clark, 2002; Goodman, 2004; Gresse et al., 2003; Solingen and Berghout, 1999)
Use of external metrics gurus	(Hall and Fenton, 1997; Daskalantonakis, 1992; Jeffery and Berry, 1993; Solingen and Berghout, 1999)
Start with projects in trouble to demonstrate the capabilities of software measurement as a valuable tool.	(Pfleeger, 1993;Wieczorek, 1997)
Assign clear and distinct implementation and data collection responsibility and send out a signal of their importance	(Grady and Caswell, 1987; Jeffery and Berry, 1993; Kitchenham, 1996)
Assign three part-time people to the team.	(DeMarco, 1982)
Ensure that the rest of their time is spent on something entirely different from the projects, the team will be measuring.	(DeMarco, 1982)
Have the group report to someone outside the project(s) being measured	(DeMarco, 1982)
Make the software measurement data collection sufficiently easy and unobtrusive by obtaining tools for automatic data collection and analysis	(Basili and Rombach, 1988a; Daskalantonakis, 1992; Grady and Caswell, 1987; Hall and Fenton, 1997; Jeffery and Berry, 1993; Jones, 2003; Kitchenham, 1996; Pfleeger, 1993; Solingen et al., 1997)
Publicize success stories of software measurement and encourage exchange of ideas	(T. Dekkers, 2005; Grady and Caswell, 1987; Jeffery and Berry, 1993; Pfleeger, 1993)

Table 3-8. Measurement Program Development Success Factors

Pfleeger (1999)

Pfleeger (1999) states that it is necessary to link the establishment of a measurement program to the maturity level of an organization. “Metrics are welcome only when they are clearly needed and easy to collect and understand.”

For example, a measurement immature organization should not attempt to implement a predictive model. Moreover, measurement cannot exceed the software process: if the development process does not define the types of tests, it is not possible to evaluate the efficiency of some tests in comparison with others. Therefore, when the measurement program defined in a company is not adapted to the measurement maturity of the company, the measurement program implementation is destined to fail. This is an especially interesting success factor since the methodology for defining and implementing measurement programs presented in this thesis (MIS-PyME) integrates this success factor into its methodological framework.

3.3.2. Other approaches suitable for SMEs

Some of the approaches presented in the previous section provide useful ideas for the creation of a measurement framework aimed at defining and implementing measurement programs for SMEs. However, none of these approaches can be integrated into its basis method and fulfil the requirements for SMEs stated in Section 3.2.7. In other words, although most of the approaches provide an improvement to GQM, the main problem of GQM as regards its suitability for SMEs is that it does not provide support modules (in contrast to PSM) to assist SMEs in the definition and implementation of software measurement programs. The above approaches do not provide these guidelines and support either. Another problem of GQM is that its methodology is not adapted to the size of an SME. Likewise, none of the above approaches deals with this issue. The exception is the work of Lindvall et al. (2005) which provides an improvement to GQM which improves the reuse of the measurement programs defined. This approach would allow GQM to fulfil the requirement of “reuse measurement models”, but is not sufficient to fulfil the remaining requirements.

The approaches of Niessink and van Vliet (1999) and Basili et al. (2007), which claim for the understanding of the measurement program as a means to achieve a business goal, can be adopted, but are not directly related to SMEs

Other approaches, such as that proposed by Bégnoche et al. (2007), use models which are not recommended for SMEs, such as CMMI, and other approaches propose practices which may be too costly for an SME (Eickelmann, 2001;Berry and Vandenbroek, 2001; Mendonça and Basili, 2000; Kitchenham, 1995; Schneidewind, 1992; Heidrich and Münch, 2008; Briand et al., 1996; Morasca, 2008; Weyuker, 1988; Dumke et al., 2006; Baldassarre et al., 2003; Lavazza and Barresi, 2005; Brijckers and Differding, 1996).

Other approaches related to predictable models and software process control are out of the scope of a small and medium setting since the maturity required for this kind of measurement programs is too high, and requires a high level of measurement knowledge and, on occasions, too much effort. These are the work of: Briand et al. (2002), Briand et a. (1996), Morasca (2008), Ramil and Lehman (2001) and Boffoli et al (2008).

None of the aforementioned approaches can, therefore, be applied to any of the studied methodologies (PSM, GQM, ISO/IEC 15939, GQ(I)M and GQM lightweight) to make these methodologies suitable for SMEs.

3.4. Measurement Maturity Capability Models

This section provides a summary of the main measurement maturity methods and models found in bibliography. This section is included in this state–of-the-art since the methodological framework aimed at in this thesis is intended to guide the measurement analyst in defining measurement programs which are adapted to the measurement maturity of the company. An analysis of the existing models and their coverage to support the definition of measurement programs is therefore presented. .

3.4.1. Daskalantonakis et al. (1990)

Daskalantonakis et al. (1990) proposes a method for assessing an organization's software measurement technology, which is consistent with the SEI Software process assessment methodology (SEI, 1995). However, it differs from the SEI because it focuses on measurement technology assessment (rather than development process maturity assessment). It is also more concrete and measurable because it tracks the evolution of measurement themes for each measurement technology.

This method is based on a number of assumptions which determine the focus of the Measurement Technology Assessment. Ten themes are derived from these assumptions, according to which the company is characterized and evaluated.

The assumptions and derived themes are as follows:

- A well defined quality-focused software development process will very likely result in a quality software project and product. Therefore, the first theme is:

  Theme 1: Formalization of the development process.

- Measurement is facilitated by, and facilitates, a well defined software development process and product. It requires a well-defined process which is integrated into the development process. This measurement process includes automation of the data collection, evaluation and feedback of the deficiencies, and the improvement of an organization's projects, products and processes. Tools such as configuration management, along with problem tracking and analysis, permit emphasis on key measures for tracking the process. Database is critical for capturing and analysing knowledge. The derived themes are as follows:

Theme2: Formalization of the measurement process

Theme3: Scope of measurement within the organization

Theme4: Implementation support for formally capturing and analysing knowledge.

- Measurement follows an evolution pattern. This begins with the project, then the product and finally the process measurement. The derived themes are as follows:

Theme 5: Measurement evolution within the organization

Theme 6: Measurement support for management control of software projects

- Project, process and product improvement is achieved by using collected data as information that identifies problem areas and by implementing mechanisms for problem prevention based upon informed analysis of the product and process. Improvement can be short-term, based on current-projects and the factors which lead them to succeed or fail. The derived themes are as follows:

Theme 7: project improvement using measurement technology

Theme 8: product improvement using measurement technology

Theme 9: Process improvement using measurement technology

Theme10: Predictability of project, product and process characteristics.

Five evolutionary stages were defined for each theme, which a software development organization may follow in order to reach the highest level of maturity for a particular theme. These five evolutionary stages correspond to the process maturity stages defined by SEI (initial, repeatable, defined, managed, and optimized). Level I of the software measurement technology corresponds to the i-th stage of the ten themes that are used to characterize and evaluate the measurement technology maturity.

In addition, the proposal provides a method for assessing the current measurement technology level of an organization, which is consistent with the SEI process assessment with regard to themes and questions. This method can be used by the organization to identify what is necessary to improve its measurement technology level.

3.4.2. Niessink and Vliet (1998)

Niessink and Vliet (1998) define a capability maturity model for measurement as being that which can be used to assess the measurement capability of software organizations and to identify means of improving their measurement capability. The model measures the measurement capability on a five ordinal scale which matches Daskalantonakis' maturity stages. However, Niessink and Van Vliet define a set of pre-established key process areas which are different for each level and which must be in place if an organization is to reside at that level. The C-MM is as follows:

- Level 1 – this is initially the level at which all organizations that have no key process areas implemented reside.
- Level 2- the repeatable level- organizations have a basic measurement process in place, which means that they are able to collect measures during projects. Measures are probably not comparable across projects since each project potentially has its own measurement goals and defines its own measures.
- Level 3-defined level- this problem is solved, because the organization standardizes its measurement process and determines a basic set of measures that each project has to collect. Also, an organization-wide measurement database is created which contains all historic project data.
- Level 4 - the managed level - the organization will be able to assess the costs of different measures. Technology is being used to make the measurement process more efficient.
- Level 5 - the optimizing level - the organization ensures that measurement processes are not only efficient, but also effective. Measures are regularly judged on their merits and measurement processes are adjusted when necessary to reflect changes in the measurement environment.

3.4.3. Measurement CMMI

The CMMI (CMMI, 2006) is a model to assess the capability of a software organization. One of the most important contributions of the CMMI, when compared to the previous models, is that it has increased the number of suggested measures in the Engineering PAs: Requirement Management, Requirement Development, Technical Solution, Product Integration, Verification, and Validation, and furthermore, CMMI states what is necessary to establish and sustain a feasible measurement and analysis process. A specific Process Area was therefore added to the model to guide the measurement process, which is known as the “Measurement and Analysis Process Area” (M&A PA). Its purpose is to develop and sustain a measurement capability that is used to satisfy management information needs.

The Measurement and Analysis Process Area supports all process areas by providing measurement practices, which give organizations guidance on how to align their measurement needs and objectives with results that can be used to make decisions and to take appropriate corrective actions.

Measurement and analysis practices are organized in the M&A PA under two specific goals which consist of:

1. Aligning measurement activities with identified information needs and objectives.
2. Providing data analyses and results that address those needs and objectives.

A successful process measurement should fulfil these objectives.

Finally, CMMI mentions that it models the natural evolution of measurement that should occur as organizations make an effort to improve their processes throughout all levels (see Figure 3-8). Measurements start to be taken into account at level 2 of the CMMI. The measure types defined for each level are the following (Johnson and Kulps, 2004):

- Level 2: Primarily status measures – planned versus actual size, effort, cost, schedule; this also includes the number of changes, number of non-conformances in product and process.
- Level 3: This adds a focus on process improvement and product development. Measures for process improvement and quality measurement, including defect density and productivity, include the number of defects, amount of rework, number of problem reports, and defect density for product development. Other metrics are also included (e.g., number of process improvement proposals submitted, accepted and implemented).

Bégnoche et al. (2007) presents the software measurement information needs identified throughout the CMMI model (CMMI, 2006) from the level 2 to 3 (see Table 3-9).

- Level 4: This adds a focus on quantitative management with the creation and use of process performance baselines PPBs, process performance models PPMs, quality and process-performance objectives, and event-level measurements. The Quantitatively Managed level is reached when detailed measures of quality and process performance are collected and statistically analysed.
    - A Process Performance Baseline statistically documents the historical results of a process. It is used as a benchmark against which to judge project performance. Some organizations have been successful with a small number of PPBs, such as Review PPB, an Effort Distribution PPB, and Productivity PPB.
    - A Process Performance Model describes the relationships among the attributes of a process and its work products. PPMs are based on PPBs and are calibrated to each project. PPMs are used to estimate or predict a critical project value that cannot be measured until later in the project's life (e.g., number of delivered defects or total effort).
    - Quality and process-performance objectives are requirements or goals set by the organization, and its projects with their product quality and the performance of critical processes in mind.

- Level 5: This adds a focus on continuous improvement based on a quantitative understanding of the common causes of variation inherent to the processes. The Optimising level is reached when "processes are continually improved based on a quantitative understanding of the common causes of variation inherent in processes"(CMMI, 2006). At Maturity Level 5, information needs come only from "quality assurance", since the aim is solely improvement.

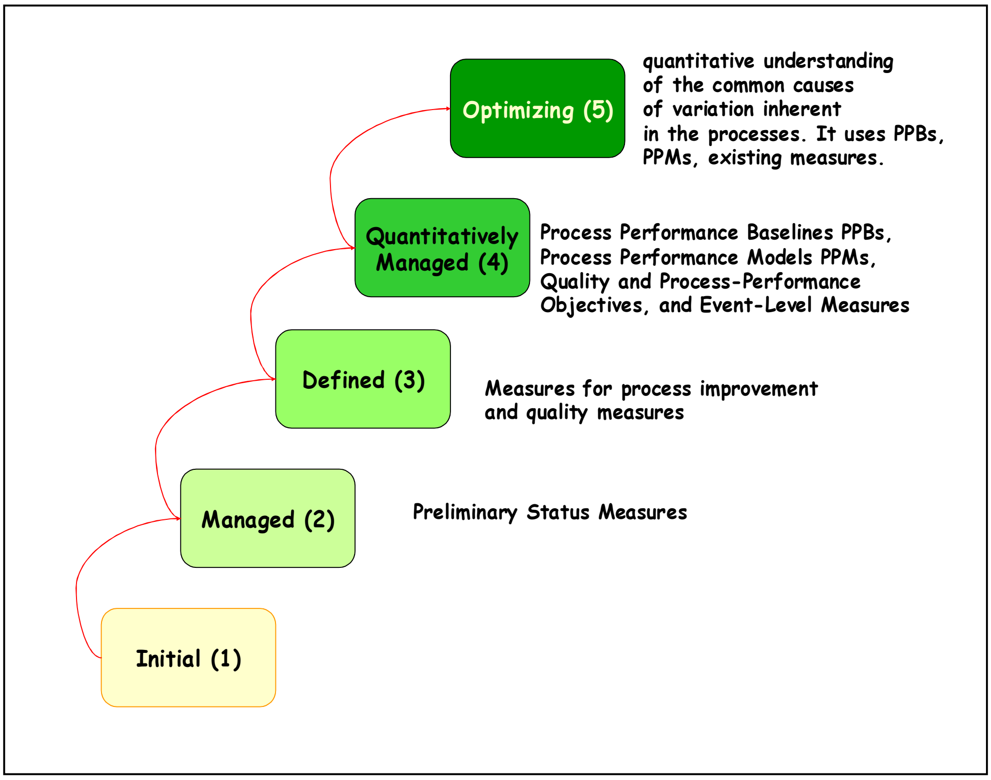


Figure 3-8. Measurement Progress through Maturity Levels

Level 2	
Requirement	Need to know the degree of compliance of the requirements with established criteria
	Need to evaluate the impact of requirements for commitment
	Need to know the consistency of other work products vis-à-vis the requirements
Analysis	Need to evaluate the risk associated with a project
	Configuration management Need to evaluate the impact of change requests Need to know the integrity of the baselines
Project management	Need to collect data about project effort, project cost, work product attributes and task attributes
	Need to estimate effort and cost using models and/or historical data
	Need to track the actual project performance.
	Need to know the effectiveness of corrective actions when taken on identified issues.
Quality Assurance	Need to evaluate the process as performed against the applicable process descriptions.
	Need to evaluate the work products against applicable descriptions
Level 3	
Requirement	Need to know the functional size of the requirements
	Need to know the completeness, feasibility and verifiability of the requirements.
	Need to track technical performance requirements during development effort
Analysis	Need to evaluate the risk associated with the requirements.
	Need to evaluate, categorise and prioritise identified risks using established criteria.
	Need to trigger a risk mitigation plan when an unacceptable level or threshold is reached
	Need to compare alternative solutions using established criteria in order to select the best solution.
Design and implementation	Need to know the degree of compliance of the design with established criteria.
	Need to know the consistency of the design vis-à-vis the requirements
	Need to evaluate the completeness and coverage of all product component interfaces
	Need to know the degree of compliance of the code with the design.
Verification and validation	Collect data from peer reviews on the code.
	Collect results from unit testing.
	Collect data from peer reviews on the documentation.
	Need to evaluate assembled product components following product integration.
	Need to confirm correct operation at the operational site.
	Need to identify corrective actions by analysing verification and validation data.
Project management	Need to estimate the project's planning parameters using the measurement repository.
	Need to manage the project using a set of specific measures.
Quality assurance	Need to appraise processes, methods and tools periodically to identify strengths and weaknesses and to develop recommendations.
	Collect data from peer reviews on the common set of measures and procedures for storing and retrieving measures.
Training	Collect data about training activities.
	Collect data about test results.

Table 3-9. Software Measurement Information Needs CMMI level 2-3, (Bégnoche et al. 2007)

3.4.4. Comer and Chard (1993)

Comer and Chard (1993) describe a software measurement process model that can be used as a reference model for the assessment of software measurement process maturity. They state that the software measurement process can be reduced to a sequence of key processes that can be assessed individually. The key processes of the measurement model are as follows:

1. Process definition (specification of entities and attributes, identification of goals, derivation of software measures);
2. Collection (operational definition of software measures, data gathering, data storage, verification, examination for patterns);
3. Analysis;
4. Exploitation.

Unlike the other maturity models described above, the measurement maturity model of Comer and Chard does not define different levels of maturity.

3.4.5. Motivation of the Research of the Thesis

After the first Action Research cycle, and as a result of reflecting on the case study performed in this cycle (see Chapter 5), a set requirements that should fulfil the measurement maturity model was stated. The aim was to integrate a measurement maturity model into the methodological framework for defining and implementing measurement programs in order to define the measurement programs according to the measurement maturity of the company. The requirements that should be fulfilled are as follows:

- 3M-REQ1: The measurement maturity model must explicitly indicate the criteria which decides whether a process is in one level or another.
- 3M-REQ2: The measurement maturity model must explicitly indicate the information needs that the measurement program is able to satisfy for each maturity level.
- 3M-REQ3: The measurement maturity model must explicitly indicate the support tools and other resources of the measurement process that must be implemented for each maturity level.
- 3M-REQ4: The measurement maturity model must explicitly provide a process with which to assess to assess the measurement maturity level of the organization.
- 3M-REQ5: the measurement maturity model used to define measurement programs adapted to the measurement maturity of the company must be compliant with the international standard ISO/IEC 15504.

a. 3M-REQ 5.1.The process reference model must be compliant with the requirement stated in the standard for the process reference model.

b. 3M-REQ 5.2. The measurement maturity model must be compliant with the measurement framework of ISO/IEC 15504.

c. 3M-REQ 5.3. The measurement maturity model must represent a mapping of the process reference model and the measurement framework of ISO/IEC 15504.

d. 3M-REQ 5.4. The process assessment of the measurement maturity model must be compliant with the process described in ISO/IEC 15504.

In order to understand whether the measurement maturity models described above were suitable for integration into the methodological framework under development, it was necessary to analyse whether these models completely satisfied the stated requirements. The results are shown in the tables below.

Table 3-10 shows how the measurement maturity model of Daskalantonakis et al. (1990) complies with the requirements previously stated.

Requirement	Compliance	Description
3M-REQ1	Yes	The themes: formalization of the measurement process and measurement support fulfil this requirement.
3M-REQ2	Yes	The themes: Scope of measurement, measurement evolution, product improvement, project improvement, process improvement and predictability fulfil this requirement.
3M-REQ3	Yes	The theme implementation support fulfils this requirement
3M-REQ4	Yes	It provides a method for assessing the current measurement technology level of an organization, consistent with the SEI process assessment with regard to themes and questions
3M-REQ5	No	The maturity levels and the process assessment are based on SEI process assessment which is one of the models on which ISO 15504 was based. However, the measurement maturity model is not compatible with the measurement framework of ISO/IEC 15504 in terms of the attributes that the process should fulfil for each maturity level (REQ 5.2, REQ 5.3).

Table 3-10. Daskalantonakis et al.'s Measurement Maturity Model Requirements Fulfilment

Table 3-11 shows how the M-CMM (Niessink and Vliet, 1998) complies with the requirements previously stated.

Requirement	Compliance	Description
3M-REQ1	Yes	This model defines a set of key process areas that the measurement process must fulfil in order to reside in each measurement maturity.
3M-REQ2	No	The model does not define the information needs that the measurement program is able to satisfy for the maturity level.
3M-REQ3	Yes	There are key process areas in this respect.
3M-REQ4	No	It does not provide an assessment process
3M-REQ5	No	The measurement maturity model is not compatible with the measurement framework of ISO/IEC 15504 in terms of the attributes that the process should fulfil for each maturity level, maturity levels, etc. (REQ 5.2, REQ 5.3). Since it does not provide a measurement assessment process, REQ 5.4 cannot be fulfilled either.

Table 3-11. M-CMM Requirements Fulfilment

Table 3-12 shows how the measurement maturity model of Comer and Chard (1993) complies with the requirements previously stated.

Requirement	Compliance	Description
3M-REQ1	Yes	This model defines a set of key processes that the measurement process should implement effectively and improve.
3M-REQ2	No	The model does not define the information needs that the measurement program is able to satisfy for the maturity level.
3M-REQ3	No	The model does not define the resources required to perform and improve the measurement process.
3M-REQ4	Yes	It proposes an assessment methodology based on the transformation of answers to a questionnaire into a grading System. The methodology checks the implementation, progress and effectiveness of the activities of each key process.
3M-REQ5	No	The measurement maturity model is not compatible with the measurement framework of ISO/IEC 15504 in terms of the attributes that the process should fulfil for each maturity level, maturity levels, etc. (REQ 5.2, REQ 5.3). The measurement assessment process does not comply with the measurement assessment process of ISO/IEC 15504 (REQ 5.4)

Table 3-12. Comer and Chard (1993) Measurement Maturity Model Requirements Fulfilment

Table 3-13 shows how CMMI (2006) complies with the requirements previously stated.

Requirement	Compliance	Description
3M-REQ1	Yes	This model defines a set of goals that the measurement process has to fulfil for each capability level.
3M-REQ2	Avarage	The model specifies the measurement needs for each maturity level. However, these are spread over all the process areas of the CMMI, and it is not,therefore, clear and easy to find this information.
3M-REQ3	Avarage	The model give guidelines to the resources required in each maturity level but these cover all of the process areas. It does not, therefore, explicitly determine the measurement support resources that should be met for each measurement maturity level.
3M-REQ4	Yes	It specifies an assessment process standard CMMI appraisal method for process improvement (SCAMPI)
3M-REQ5	Yes	CMMI was defined together with ISO/IEC 15504 and therefore complies with all of the requirements

Table 3-13. CMMI Requirements Fulfilment

To sum up, none of the measurement maturity models found in bibliography complies with the requirement established.

The most complete measurement maturity model is CMMI. This model clearly defines the measurement process improvement, but the identification of measurement needs and measurement resources for each maturity level is assigned to the rest of the software processes: Project Management Process, Quality Assurance Process, etc. In order to provide a clear and efficient measurement support with regard to the measurement maturity information, all the related measurement information must be grouped together and organized in a manner which facilitates its efficient use.

A measurement maturity model which complies with the above requirements was therefore developed in this research. This measurement maturity model will support the measurement analyst in defining measurement programs adapted to the measurement maturity of the company.

3.5. Experiences

This section describes the experiences found in bibliography with regard to the implementation of measurement programs in companies. We shall focus on the following: in which company the measurement program was implemented, its objective, the methodology used, the results, and any other significant information about how the measurement program was carried out. Note that in this section GQM is used not only to define measurement programs for the organization and project management and control, but also to support other software-related goals.

Daskalantonakis (1992)

One of the measurement program experiences most referenced in bibliography is that of Daskalantonakis (1992) which describes a software measurement implementation in Motorola. This was a 3-year project and one of the company initiatives regarding software quality.

Metric areas were defined on the basis of the Quality Policy for Software Development. Areas such as Process Improvement, Process Control etc. were defined. The overall philosophy of the company-wide software metrics initiative was that measurement was not the goal, the goal was improvement through measurement, analysis, and feedback. The GQM methodology was used to define metrics.

Some experiences gained from this project were the following:

- Metric data collected should be interpreted in its context.
- Engineers and managers involved in the project are the best people to analyse the collected data, because they have expertise in the project domain and can interpret what the data indicates.
- If only one metric is used, projects can manage to optimize the value of that metric (indicating positive results). However, they will not have a global view of the situation and the project may have significant problems that are not obvious by examining one metric only.
- The people at Motorola have started thinking more seriously about software processes and quality. The data has helped projects to understand the extent of the problems they were facing and motivate them to improve. For example, the focus within a Motorola Division on improving software quality (and tracking results through metrics) has achieved a 50% reduction in defect density of the released software within 3.5 years. As a result of improving software quality, total Customer Satisfaction has increased

MacDonell and Fletcher (1998)

MacDonell and Fletcher (1998) studied the metrics that should be defined in an organization that develops multimedia systems. The aim was to define a multimedia framework to determine the measures that are dependent on the development effort. Hence, the dependent attribute was the effort and the independent attributes were the measures of the framework. The objective was to demonstrate a significance correlation between both sets (dependent and independent attributes).

The GQM (Basili and Rombach, 1988a) approach was used in this study to determine system and component characteristics that are considered by industry to be influential and which affect multimedia systems development efforts.

The multimedia measurement framework was tested by means of a survey carried out in 22 multimedia organizations. The result of the measurement program indicated that there are differences between these multimedia organizations and those that develop traditional systems such as process control systems. Finally, a framework which addresses the issues of which development effort is dependent on multimedia systems was defined.

Geppert and Weiss (2003)

Geppert and Weiss (2003) developed a measurement program to select the most suitable domain of a product in order to start creating the product-line domain.

The measurement program was designed for the Avaya laboratories and the aim was to choose the 20 best module candidates from the 200 modules available in one of its major product lines.

The GQM (Basili and Rombach, 1988a; Basili and Weiss, 1984), model was used to perform the measurement program and to determine what measures would be used to assess candidate domains. The sub-goals of the selection were the corporate impact or value of the product line development, and the likelihood of success.

Finally, the measurement program was performed and the best modules were selected.

Lindvall et al. (2002)

Lindvall et al.(2002) defined a measurement program in order to demonstrate the advantages of software architecture restructuring. The value of an improved architecture is clear to technical staff, but it is often difficult to convince upper management that extra effort is necessary. A measurement program was therefore carried out.

The goal was to create a measurement program to evaluate the two versions of the Experience Management System (EMS) at the Fraunhofer Center in Maryland: EMS1 (before restructuring), EMS2 (after restructuring). The focus was on maintainability and the point of view was software development.

The GQM methodology was used to define measures in order to evaluate the maintenance of the two architectures.

The results of the measurement program were as expected: the new architecture was easier to maintain than the old one.

Layman and Rohde (1999)

Layman and Rohde (1999) present the lessons learned by the authors through their experiences in implementing PSM during the last three years in industry. The lessons learned were identified as follows:

- Lesson 1: There is often a disconnection between the measures currently collected and the issues "real projects" face. The organization may have a measurement process that should be followed in all projects in order to perform cross-project analysis, etc. but the project manager may find that it is not useful for a specific project. Therefore:
    - Recognize that the organization reporting requirements are simply a subset of the project measurement process and learn to make better use of what is required in the project.
    - Encourage the measurement group to understand and communicate the benefits and usefulness of the measurement results reported.
    - Differentiate the information needed to make organizational decisions from the information needed to provide senior management oversight into key projects.
    - Recognize that, without effective use of measurement at the project team level, the measurement program within an organization will be weak.
- Lesson 2: Making people need measurement is the best first step toward institutionalizing it.
- Lesson 3: The project culture will make an impact on the implementation of measurement:
    - Management that reacts negatively rather than constructively to less-than-stellar performance creates a culture that finds itself constantly in "crisis" mode.
    - Do not give customers any insight into the development process—they do not understand these things

  Both management and customers must be willing to listen and respond constructively to bad news.

Goethert and Hayes (2001)

In (Goethert and Hayes, 2001) the lessons learned from three experiences of software measurement program implementation using GQ(I)M are reported. The cases are the following:

- Case 1: The purpose of this measurement program was to establish a system of uniform measures across a global enterprise. Among the advantages that this organization was trying to achieve with their measurement program were:
    - the ability to answer questions about the enterprise (For example: are we getting better or getting worse; is an enterprise-wide improvement program having an effect?)
    - the ability to evaluate new technologies, methods, and practices by collecting identical measures to enable meaningful comparisons and trend analysis by creating a large pool of project data from which similar projects can be chosen for comparison purposes
    - the establishment of a visible ongoing enterprise focus for software engineering excellence

  The business goals were as follows: Increase productivity by a factor of 2 over 5 years , improve quality by a factor of 10 over 7 years , improve predictability to within 5% over 7 years, reduce development time by 40% over 7 years, reduce maintenance effort by 40% over 7 years
- Case 2: Assessing the impact of Software Process Improvement: The purpose of this measurement program was to assess the impact of investment in Software Process Improvement (SPI). As a consequence of the ongoing implementation of SPI activities, the schedule, cost, and quality of future software projects were expected to be significantly better than previous efforts.
- Case 3: The purpose of this measurement program was to support management in workload balance and effective project management in the context of an ongoing process improvement program. Establishing a common basis for comparing information across a widely distributed organization was a major concern. From the perspective of the organizational sub-unit, their top priority was to report compelling information that accurately reflects the work being done. From the perspective of the sponsor, the expressed priorities were to support enterprise-wide performance management and to use measurement to support the transfer of process improvement suggestions throughout the enterprise.

The indicators, measures and procedures for implementing these measurement programs were defined. A set of lessons learned were derived.

3.5.1. Motivation of the Research of the Thesis

As can be concluded from the bibliographical study, there are few experiences of the definition of measurement programs in SMEs. The measurement program experiences have all been reported in big companies. Since the origin of this research is derived from the needs of a small software development division of a medium company, this thesis implies an experience in an SME. The development and results of defining and implementing a measurement program in this context constitute one of the main goals of this research and the third motivation of this thesis.

	GQM(Solingen and Berghout, 1999)	GQ(I)M(Park et al., 1996)	GQM lightweigh method	ISO/IEC 15939	PSM
Planning,				Accept requirement for measurement: Identify the measurement scope, management commitment and communication commitment	Obtain Organizational Support
	Defining GQM team		Introduce measurement program	Assign resources	Define Responsibilities, and Provide Resources
	Select improvement areas	Identify business goals and identify what you want to learn		Characterize organization unit and Identify information needs	Identify and prioritize project issues
	Select application project and establish project team				
Definition:	Define measurement goals, conduct GQM interviews, review or produce software process models.	Identify sub-goals, identify entities and attributes formalize measurement goals	Define measurement goals, goals formalization.		
	Define questions and hypotheses and review them	Identify quantifiable questions and indicators	Define questions		
	Produce analysis plan	Identify data elements			
	Define metrics and review	Define measures	Define metrics	Select and specify measures	Selecting and specify project measures
	Produce GQM and measurement plan	Identify the actions needed to implement measures, prepare the plan.	Produce GQM plan, define data collection procedures, define data instruments	Define data collection, analysis, and reporting procedures	Integrate measures into the technical and management processes - Define data collection, define indicators, reporting procedures
				Define criteria for evaluating the information products and the measurement process	
	Review plans		Produce data collection plan	Review, approve, and provide resources for measurement tasks	
Integration/ Data collection phase	Hold Trial period: Define collecting procedures, create metrics base, etc. and test them.		create metrics base	Acquire and deploy supporting technologies	
				Integrate procedures	

	GQM(Solingen and Berghout, 1999)	GQ(I)M(Park et al., 1996)	GQM lightweigh method	ISO/IEC 15939	PSM
	Hold kick-off sessions The objective is that all participants agree on the data collection activities and are informed of the procedures, tools and templates				
	Collect and check data collection form, store measurement data in metrics base		Collect and validate data. Store data collected	Collect data: collect data, store it and verify it	Collect data: collect data, store it, verify it and normalize data
The Analysis and Interpretation phase.	Define analysis sheets and presentation slides		Data analysis	Analyse data, Interpret results, and develop and verify information products. Communicate results	Analyse indicators
	Prepare feed-back sessions, organize feed-back session, report measurement results.		Data interpretation/ feedback sessions		Report Results and Use Results for decision making
Evaluate measurement				Evaluate information products and the measurement process, Identify potential improvement and identify lessons learned (measurement experience base)	Evaluate measures and indicators, the measurement process and update the experience database.
Packaging			Packing results		

Table 3-14.Comparison of Methodologies

4. MIS-PyME

This Section describes the methodological framework developed in this thesis. Its aim is to define and implement measurement programs focused on SMEs and adapted to the measurement maturity of the company.

4.1. MIS-PyME Requirements

MIS-PyME is a methodological framework focused on defining measurement programs based on software indicators in small and medium enterprises or settings (divisions of companies). It is focused on software development and maintenance companies or settings with the typical characteristics of SMEs as regards measurement activities, namely:

- The people that are involved in the measurement program, including the measurement analyst, are from within the company and do not always have any great expertise in the field.
- Poor measurement maturity: poor measurement culture, knowledge and training; measures collected in the company are few and the measurement process is not established in the unit or company, or it does not exist.
- The personnel are reluctant to use measurement.
- A small or medium software development company or unit with limited resources and with less than approximately 50 people.

In order to define a methodology adapted to this context, MIS-PyME has been designed to satisfy certain requirements which were derived from the first Case Study results of this research (see Chapter 5). A set of decisions were made in order to ensure that this methodological framework satisfied the requirements. The requirements and the related decisions are as follows:

- **Few but effective and complete STEPs (FSTEP):** the methodology should be as simple as possible. Only the necessary steps should be explained in detail in order to support the whole cycle of the measurement program, from its definition until its implementation. The base methodology chosen was GQM (Solingen and Berghout, 1999) and GQ(I)M (Park et al., 1996). The GQM methodology was chosen since it is one of those most used in industry, it is a clear and complete methodology and it provides a description which begins with the definition and continues to the implementation of the measurement program. MIS-PyME is influenced by GQM since it describes how to derive a measurement program from goals; and it is influenced by GQ(I)M since it adds the indicators as a key element of the measurement program. The structure of the measurement program proposed in MIS-PyME is based on the indicators, and the GQ(I)M indicator template has been adopted and tailored to MIS-PyME. Moreover, certain aspects have been adapted in order to define a methodology which is easy to use in SMEs. Some of these aspects are: describing the roles that will be involved in the measurement program definition and implementation, identifying the people who should perform each of the activities and determining the number of people who should perform each role. These issues are focused on SMEs and satisfy the requirement denominated as **“Few People involved in the Process” (FPP)**. In addition, in order to access the MIS-PyME Framework, improve its understanding, and follow the methodology in a more simple and rapid manner, it was decided to specify the methodology using the Software Process Engineering Metamodel (SPEM) (OMG, 2009).

- **ReUSE measurement models (RUSE)**. The “Reuse and project-specific tailoring” principle, stated by Basili and Rombach (1988a; 1988b) indicates that measurement planning should reuse models and metrics which have been defined for the whole organization. However, these measurement models might be tailored to the project or specific characteristics of a product. MIS-PyME encourages companies to define measurement programs that will be valid and useful in almost all products or projects, and by doing so makes the establishment of measurement programs easier in the development unit, and also makes cross-project and product control possible.

  In relation to the above principle, we base the measurement program on the indicators, since they provide high level information and can be applied to most projects, products and processes depending on the nature of the indicator. Measures used in the indicator may be specific to certain projects or products, but when the measurement program is to be applied in a new project (product or process) these measures will be adapted to the new project.

- **Specific Guidelines to support basic Process Improvement Needs (GPIN)**: Defining measurement programs with the sole purpose of supporting the software process improvement initiatives was suggested by Daskalantonakis (1992) after his experiences at Motorola. MIS-PyME states that it is essential for the success of the measurement program to integrate the measurement process into the other processes of the organization. If the measurement process is still not well integrated in the organization’s culture, and it is not sufficiently mature, it is costly to establish measurement programs. Small and medium settings cannot achieve the successful implementation of measurement programs which come from just any business goal, or at least this achievement would be costly. Processes and measurement are, therefore, two areas which are highly connected and which need to improve together. Since measurement is not an aim but a means, MIS-PyME leaves the responsibility of seeking the goal of the measurement program to process improvement initiatives. Therefore, the starting point of MIS-PyME methodology is a process improvement goal. MIS-PyME does not deal with the relationship between the measurement program process improvement goal and the business goal, as is stated in (Basili et al., 2007) (See Chapter 3), since business and process management processes already manage this relationship.

  MIS-PyME provides the set of software measurement goals commonly required in software development and maintenance SMEs which are necessary to achieve common process improvement goals (P.I.G). These process improvement goals originate in the

COMPETISOFT process model (see Appendix A). The idea is similar to that proposed by Bégnoche et al. (2007) (See Chapter 3). However, the information needs do not come from CMMI but from a model adapted to SMEs, as is COMPETISOFT. Figure 4-1 shows the structure of MIS-PyME from the viewpoint of the initial input of MIS-PyME, and its relation with the COMPETISOFT model.

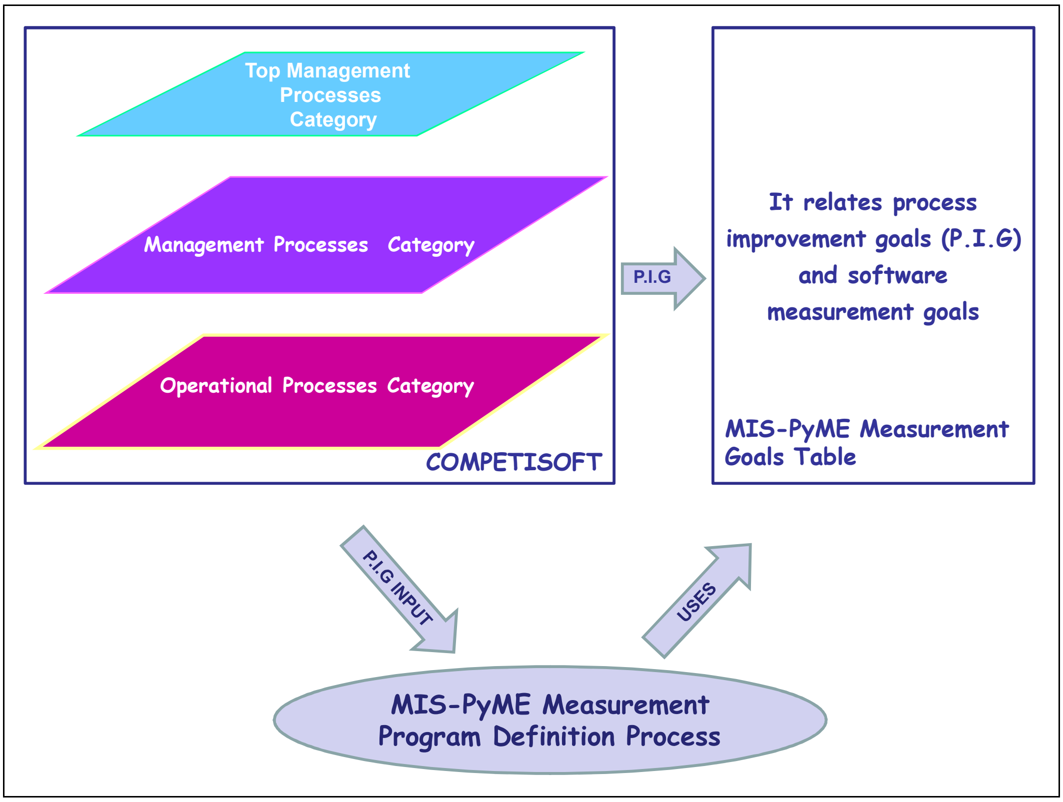


Figure 4-1. COMPETISOFT and MIS-PyME

MIS-PyME is, therefore, focused on defining the measurement programs for the organizational management of software development and maintenance SMEs. It is not as focused, as is PSM, on managing only projects, but on supporting all the levels of the organization. In addition, MIS-PyME provides an indicator template for each of the common measurement goals which helps the measurement analyst to define the measurement program based on the measurement goal identified (see Section 4.2.3).

- **Specific Guidelines to INTegrate measurement into the software processes (GINT)**: Integrating the measurement program into the organization's processes is a widely recognized success factor in bibliography (see Chapter 3). As mentioned in the previous section, the MIS-PyME design is based on the idea that the measurement program must be completely integrated into the organization's processes. Some of the improvement goals of these processes therefore become the input of the measurement program and the measurement program is thus developed to be directly integrated into the organization's processes since the resulting measurement program is a "tool" that these processes require. In addition, each of the MIS-PyME Indicator Templates provides information regarding how each indicator should be integrated into the company's processes. The reference model for this integration is COMPETISOFT.

- **Specific Guidelines to understand the Benefits & Potential for management and other guidelines (GB&P):** The MIS-PyME Indicator Templates provide information to assist the measurement analyst in understanding the usefulness of the results of this indicator for management decision making, monitoring of the project and process performance, etc.

- **Specific Guidelines to adapt measurement definition to the Measurement Maturity of the company (GMM):** MIS-PyME promotes the definition and implementation of measurement programs which are adapted to the measurement maturity of the setting. Companies may work in defining and implementing measurement programs that they are able to successfully implement and not to attempt to achieve the best measure when there are several obstacles that make a successful implementation impossible. In order to achieve this goal, MIS-PyME integrates a measurement maturity model into the methodology in order to assist the measurement analyst in a definition of measurement programs that are adapted to the company's measurement maturity.

Next sections describe the main components of MIS-PyME framework, namely the MIS-PyME Methodology, the MIS-PyME Measurement Maturity Model (MIS-PyME 3M) (see Figure 4-2) and the supporting prototype tool.

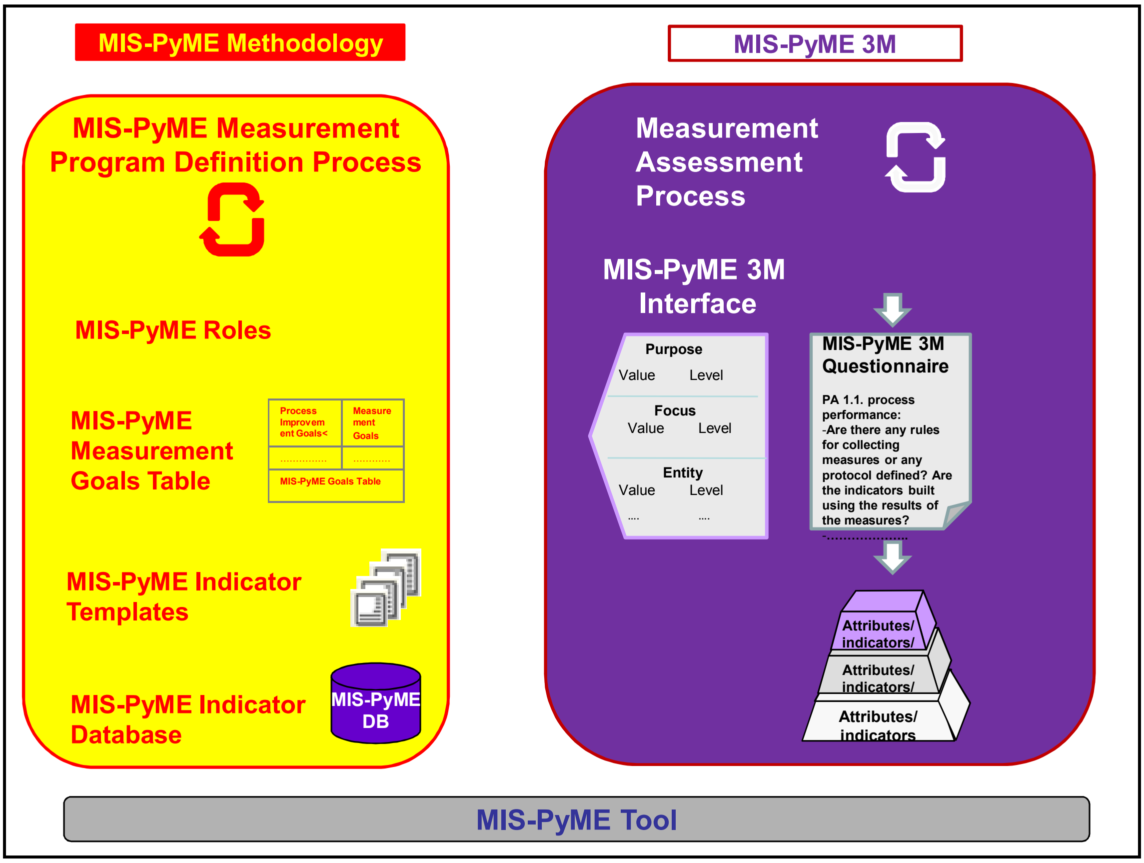


Figure 4-2. MIS-PyME Framework Components

4.2. MIS-PyME Methodology

This section provides a detailed explanation of the contents of the MIS-PyME framework methodology. Figure 4-3 shows an overview of MIS-PyME methodology. The following subsection describes each of its main components: MIS-PyME roles, MIS-PyME measurement program definition process and MIS-PyME support modules.

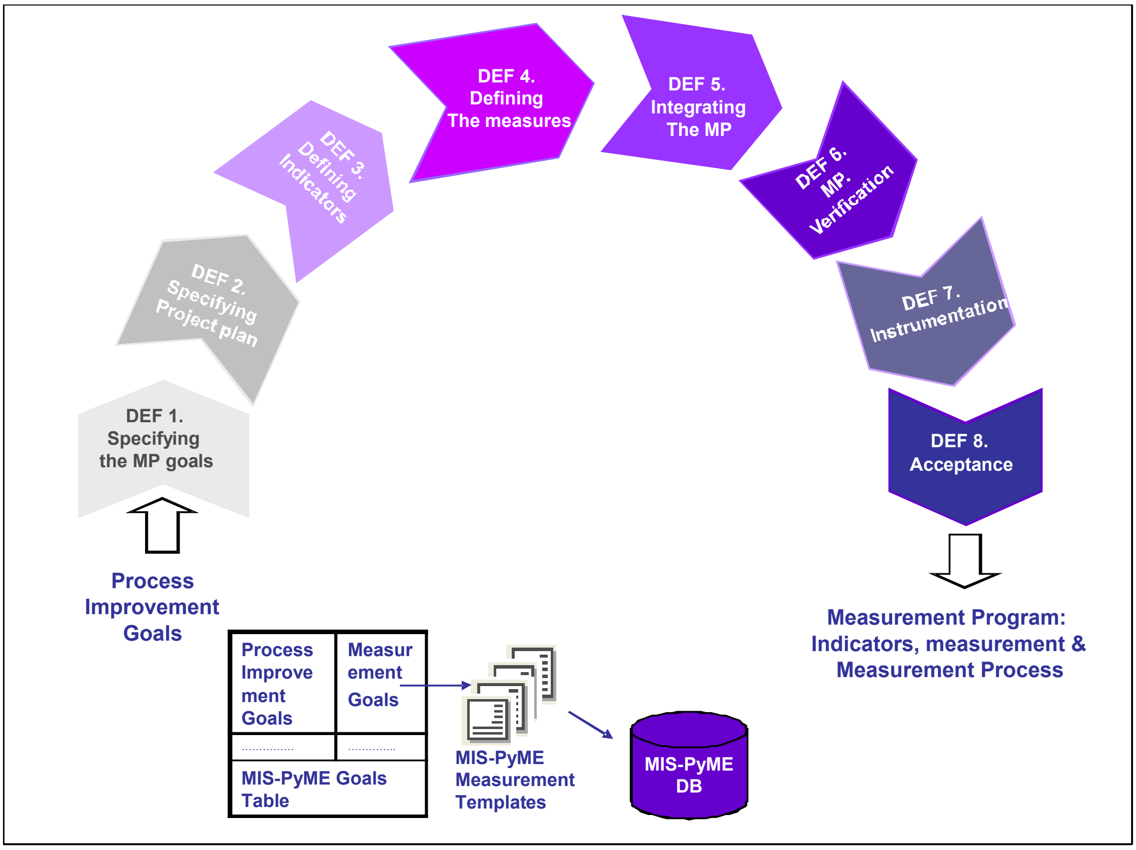


Figure 4-3: General View of MIS-PyME

4.2.1. MIS-PyME Roles

MIS-PyME methodology defines a set of roles involved in the definition and implementation of the measurement program. These are:

The **Measurement Analyst**, who should be familiar with the processes in the company. It is preferable if his/her usual work relates to management, configuration management, quality or security issues, rather than to design or development tasks. S/he may have some knowledge of software measurement but may not be highly experienced. S/he will be assigned to the measurement activities on a part-time basis.

The **Sponsor** is the person who sponsors the measurement program initiative. S/he is responsible for commitment to the measurement program development and implementation, and for giving information to the measurement analyst in order for him/her to be able to develop the measurement program according to the needs. S/he should have an in-depth knowledge of the working method, software processes and process improvement needs. S/he is responsible for defining the measurement program process improvement goals. S/he might be the person responsible for processes management.

The **Reviewer**: S/he is the person responsible for performing the verification and acceptance activities. This role should usually be played by project managers or experienced software developers.

4.2.2. MIS-PyME Measurement Program Definition Process

The MIS-PyME process aims to define and implement measurement programs. The process is formed of activities and the activities contain a set of tasks. For each task, the input and output products are described, and the role of the person responsible for carrying out that task is stated. The process is as follows:

DEF 1. Specifying the measurement program goals

The process improvement goals which are the input of the process are clarified, and an explanation of why these process improvement goals are required is presented. The measurement goals that will help to achieve those process improvement goals are also defined in this activity, and the plan for developing the measurement program is defined and communicated to the people who will be involved in this project (see Figure 4-4).

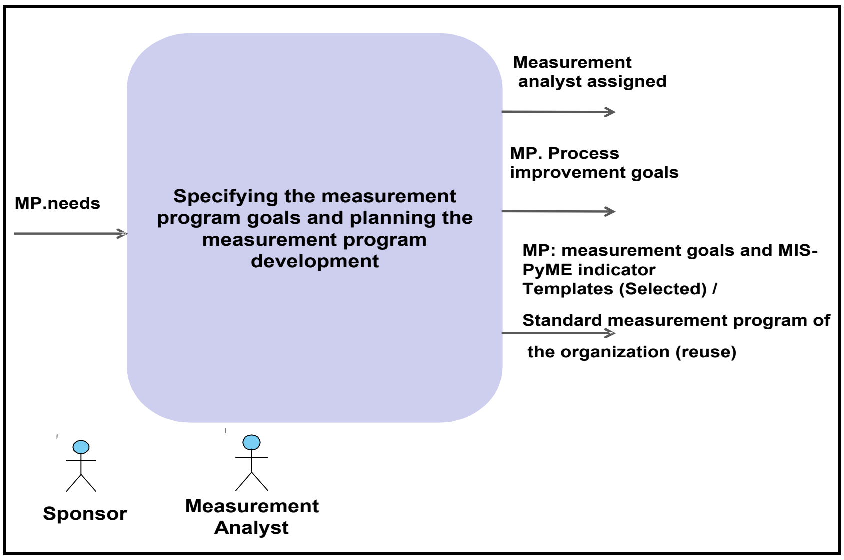


Figure 4-4. MIS-PyME Methodology DEF 1

DEF 1.1. Initiate the measurement program: The purpose of this task is to clearly state the process improvement goals that the measurement program aims to support and why these process improvement goals are required. In addition, the measurement analyst who will develop and implement the measurement program is appointed.

The measurement program's origin is the company's need for a measurement framework which supports a certain process improvement goal. Likewise, the process improvement goal is defined to support any business goal defined in the company. Business goals are determined by the board of directors and process improvements are usually identified by those responsible for the processes management. Therefore, the measurement program should have the commitment of the directors and managers of the company from the outset.

In this task, the sponsor has to determine the process improvement goals that require a measurement program. In addition, the sponsor should clearly determine the benefits of the stated process improvement goal and why this need arose. These process improvement goals are the input of the measurement program. It is recommendable to develop a measurement program which satisfies only one process improvement goal in order to increase the chances of success.

Some examples of process improvement goals may be: to improve process efficiency, to improve project monitoring practices, to improve the quality of the product, etc.

Finally, the sponsor has to appoint the measurement analyst who will be responsible for the development and implementation of the measurement program.

Main Role:	Sponsor
Input	MP.needs
Output	MP. Process improvement goals Measurement analyst assigned

Table 4-1. MIS-PyME Methodology – DEF 1.1

DEF 1.2. Formalizing measurement goals and checking whether a measurement program can be reused: The purpose of this task is to determine the measurement goals that support the process improvement goals and to identify whether a measurement program that can be reused has already been defined in the company.

The measurement analyst should first determine whether a measurement program that supports the required process improvement goal has already been defined in the company. If one exists, it is necessary to verify whether the measurement goals and indicators defined for the measurement program suit the needs. If the measurement program is suitable for reuse, the related measurement processes for the measurement program are identified and the measurement process is carried out. On the other hand, if there is no measurement program that can be reused, then the measurement analyst must define one.

The common measurement goals that will support the common process improvement goals are defined. The MIS-PyME measurement table supports the measurement analyst in completing this task. The steps which should be take place are as follows:

1. **Identifying measurement goals:** The measurement analyst and the sponsor should determine the processes affected by the process improvement goal. For example, if the process improvement goal is to improve project tracking for a specific critical project, the administration of a specific project process will be affected. If the process improvement is to improve the maintainability of the products developed, then the software development process is affected. If the aim is to implement the improvement in all of the organization's projects (e.g. improve project estimation), the management process and the specific project administration will be affected.

   The MIS-PyME Measurement Goals Table (see section 4.2.3) helps to perform this task since it determines common process improvement goals and the related measurement goals. This table does not, perhaps, explicitly specify the required process improvement goal but there are other related process improvement goals whose related measurement goals are relevant to the desired process improvement goal. If the process improvement goal appears in the table, the measurement analyst and the sponsor should analyse which of the measurement goals indicated in the table they wish to implement in order to achieve the process improvement goal. Once these measurement goals have been selected, the measurement analyst and the sponsor should look at the corresponding indicator template in order to clarify and modify (if necessary) the description of the measurement goal. The template's "description field" helps to complete this task.

2. **Suitability of the measurement goal in terms of maturity:** The indicator template is also checked to discover whether the recommended measurement maturity level for the indicator is suitable with regard to the company's measurement maturity. The measurement analyst is thus able to assess the measurement maturity of the company (see Appendix E). If the measurement maturity of the company is N, MIS-PyME recommends the unproblematic implementation of those measurement goals which belong to maturity levels N or lower. If the measurement goal belongs to the measurement maturity N+1, a thorough analysis should be performed in order to determine whether the measurement goal can be successfully implemented. Measurement goals which belong to a higher maturity level are not recommended.

   If the measurement goal that was intended to be implemented is too high as regards the measurement maturity, the measurement analyst and the sponsor will attempt to implement a more simple measurement goal and to evolve the measurement program over time.

If the MIS-PyME Measurement Goals Table is not helpful in determining the measurement goals that satisfy the process improvement goals, then the measurement analyst should put questions to the sponsor with regard to what information needs to be known in order to understand how to achieve the improvement goal. The measurement goal is derived from these answers. The measurement goals should be described as follows: Purpose + Focus + Entity to be measured + Usefulness.

- **Purpose**: Why the object will be analysed. MIS-PyME restricts the purposes as follows: characterizing, monitoring and evaluating. Predicting and controlling/changing are purposes which correspond to maturity levels 4 and 5, which an SME is not expected to attain.
    - Characterizing: aims to form an overall picture of the current state/performance of the software development processes and products
    - Monitoring: aims to follow the trends/evolution of the performance/state of processes and products.
    - Evaluating: aims to compare and assess the quality of products and the efficiency/effectiveness of processes.
- **Focus of the indicator**: This states the particular attribute of the object under study that will be characterized, monitored or evaluated. Examples of quality focuses are cost, reliability, correctness, defect removal, changes, user friendliness, maintainability, etc.
- **Entity to be measured**: Target of the measurement activity: process (PROC), project (PRJ), product (PROD) or resource (RES). If cross analysis of these entities are required, then the label "ORG" should be included at the end of the name of each entity.
    - PRJ: The entity to be measured is the project. Therefore the attributes required to track and improve the project are measured, such as the duration, the effort allocation per phase of the project, etc. (Daskalantonakis, 1992).
    - PROC: The entity to be measured is the process applied in the project, and are, therefore, the attributes meant to improve the software development and maintenance process. Examples of such attributes are the effectiveness of inspections, performance, etc. (Daskalantonakis, 1992).
    - PROD: The entity to be measured is the software product, so the attributes meant to improve the software product quality are measured. Some examples of these attributes are the complexity of the design, the size of the source code, and the usability of the documentation produced.(Daskalantonakis, 1992).
    - RES: The entity to be measured is the resource and therefore the attributes which are meant to improve human and non human resources are measured. Some examples are "cost" and "conformity".
- **Usefulness**: "In order to …." The final purpose of the measurement goal is described.

Main Role:	Measurement Analyst
Additional Roles:	Sponsor
Input	Standard measurement programs of the organization (if they exist)
Output	MP: measurement goals and MIS-PyME indicator Templates (selected) or Standard measurement program of the organization (reuse)
Guidelines	MIS-PyME measurement goals Table

Table 4-2. MIS-PyME Methodology – DEF 1.2

DEF 2. Specifying project plan

A small project plan should be defined and a brief introductory session to the people involved in the project should be given (Figure 4-5).

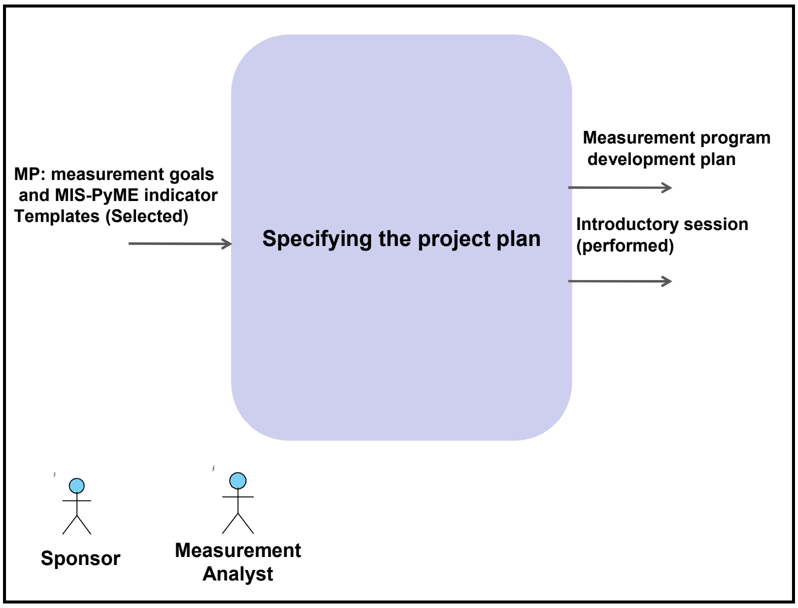


Figure 4-5. MIS-PyME Methodology DEF 2

DEF 2.1. Specifying the project plan: The purpose of this task is to define a small project plan in order to formalize the measurement program and to establish commitment in the company.

The project plan should include:

- the description of the measurement program goals,
- the benefits of the measurement program,
- the people involved and their roles,
- the calendar of the main activities, especially the acceptance phase, and
- the specification of the acceptance phase.

If the measurement program is defined for a specific project, the measurement program definition will be focused on that project and this will therefore be the pilot project and the implementation of the measurement program should be in accordance with the project schedule. On the other hand, if the measurement program is specified for other proposes, a trial analysis of the measurement program should be planned.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: measurement goals and MIS-PyME indicator Templates (selected)
Output	Measurement program development plan

Table 4-3. MIS-PyME Methodology – DEF 2.1

DEF 2.2. Specifying the project Introductory session A brief introductory session should be given to the people involved in the review and the acceptance of the measurement program. If the measurement program is focused on a project, this session should be given to the project manager, and it is not therefore necessary for all the people involved in the project to be present.

The introductory session should deal with the following issues:

- the benefits of the measurement program,
- the description of the measurement program goals,
- the calendar for the main activities, particularly the review and acceptance of the measurement program, and
- the plan for the acceptance of the measurement program.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	Measurement program development plan
Output	Introductory Session (performed)

Table 4-4. MIS-PyME Methodology – DEF 2.2

DEF 3. Defining indicators

The indicators required to implement measurement goals are defined in this activity (Figure 4-6).

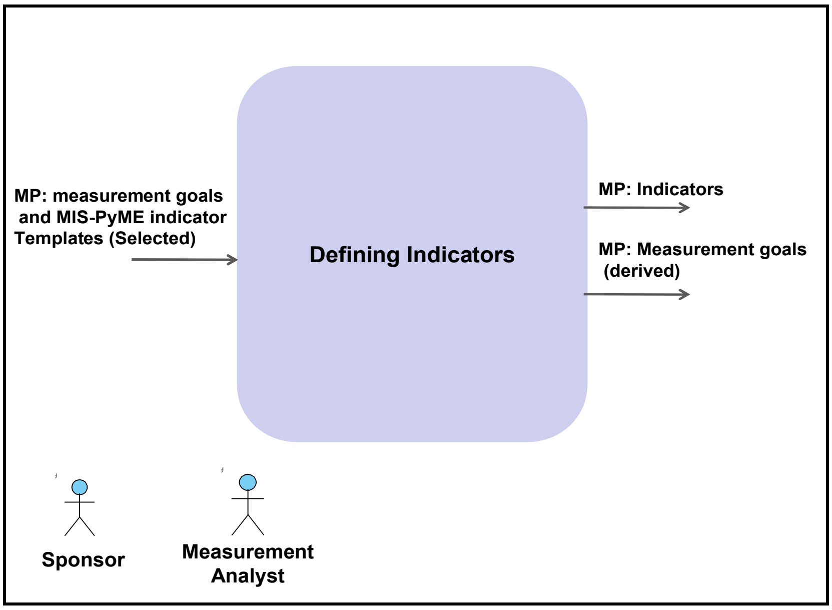


Figure 4-6. MIS-PyME Methodology DEF 3

DEF 3.1. Specifying the Indicators. The purpose is to define an indicator for each of the measurement goals defined in DEF 1.

The indicator is specified by filling out an indicator template provided by MIS-PyME. If the corresponding measurement goal of the indicator is a common measurement goal included in the MIS-PyME Measurement Goals Table, there will be an associated indicator template which includes useful guidelines for specifying the indicator.

The indicator is required to develop the measurement goal. The information that has to be specified for the indicator is as follows (Goethert and Siviy, 2004):

- **Name of the indicator**: This can be formed of the following fields: IND-ENTITY-ABBREVIATION (see Section 4.2.3)
- **Description**: The description of the measurement goal as described in DEF 1.2.
- **Point of view**: The description of the audience for whom the indicator is intended. This means the people who will be benefit from the analysis made by the indicator.
- **Context**: The environment in which the measurement will be performed, analysed and interpreted. This also determines how the results can be generalized.
- **Questions**: This is one of the most important pieces of information for the indicator. The questions that the indicator intends to solve and understand are specified. These questions will clearly determine the purpose of the analysis and will guide the analysis performed.
- **Inputs**: The input measures or derived indicators required to build the indicator.
- **Algorithm**: The description of the algorithm required to construct the indicator is specified at this point.
- **Improvement of the Indicator**: It is better to start small and evolve over the time and therefore, if the indicator implemented does not provide all its features and possibilities, this section indicates and describes the indicator's evolution and final intention.
- **Measurement Activity Information**: in this section the information as regards how this indicator will be integrated into the company's process is determined:
    - Person responsible for analysis and feedback: Which group or persons are responsible for the analysis?
    - Destination of the analyses and interpretation of results, audience: Which groups or roles are interested in this information?
    - Analysis and feedback frequency: How often can the indicator be analysed?
- **Indicator results report location**: Where will the full indicator report be located?
- **Integration**: How is the indicator integrated into the software process?

- **Resources required**: This section indicates the tool used to build the indicator and perform the analysis if necessary.
- **Analysis**: Typical outcomes and related interpretations are also proposed as an example in order to understand the usefulness and potential of the indicator.
- **Output Display**: The specification concerning how the indicator will be graphically shown is indicated, and an example is also provided.

The MIS-PyME measurement indicator templates provide full guidelines as regards how to define the common indicators required in a company. The measurement analyst and the sponsor should adapt the specifications indicated in the MIS-PyME indicator template to its roles, processes, questions required, etc.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: measurement goals and MIS-PyME indicator Templates (selected)
Output	MP: Indicators
Guidelines	MIS-PyME Indicator Templatess MIS-PyME 3M

Table 4-5. MIS-PyME Methodology – DEF 3.1

DEF 3.2. Searching in MIS-PyME database. The purpose is to search the MIS-PyME database for any indicator which matches the measurement goal being defined in order to obtain ideas and refine and review the indicator defined.

If the measurement goal being defined is a common one and is included in the MIS-PyME Measurement Goals Table, its matching indicator template may have examples of indicator specifications already implemented in the company.

The measurement analysts can check whether any of the examples in the database are related and, if a suitable one is found, they can obtain ideas for defining or reviewing the indicator.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MIS-PyME Indicator Templates (selected)
Output	MP: Indicators
Guidelines	MIS-PyME Indicator Templates MIS-PyME Indicator Database

Table 4-6. MIS-PyME Methodology – DEF 3.2

DEF 3.3. Identifying derived goals. The purpose is to identify measurement-derived goals. These goals are those derived from the questions presented or the inputs recommended in the MIS-PYME indicator template table.

When putting questions to the indicator or when the input measures or indicators required for building the indicator are specified, it might be recognized that there are other measurement goals and related indicators that may also be specified. These are called measurement-derived goals, which may also have their corresponding measurement goal in the table for MIS-PYME measurement goals and their corresponding MIS-PYME indicator templates. Tasks DEF 1.2, 3.1 and 3.2 may then be repeated until all measurement-derived goals and their relevant indicators have been defined.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: Indicators
Output	MP: Measurement goals (derived)
Guidelines	MIS-PyME Indicator Templates

Table 4-7. MIS-PyME Methodology – DEF 3.3

DEF 4. Defining your measures and specifying the measure results collection procedure

The measures required to build the indicators and the procedure explaining how to obtain the result of the measure are specified in this activity (Figure 4-7).

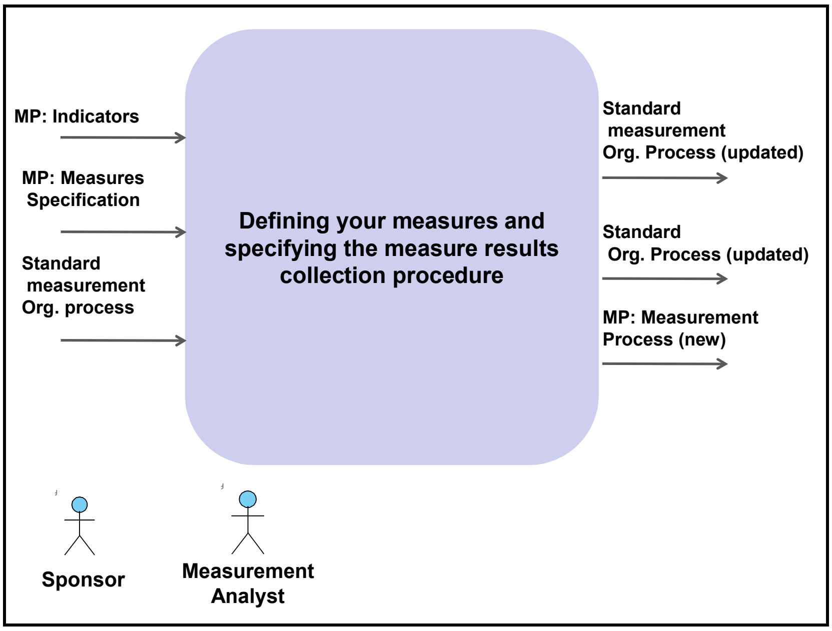


Figure 4-7. MIS-PyME Methodology DEF 4

DEF 4.1. Defining your measures. The purpose is to specify the measures required to build the indicators. The measures that have to be obtained are described in detail. The measure should be specified by a checklist which clearly specifies which data should be collected in order to obtain the result of the measure and which data are not taken into account. It is therefore possible to specify which data are to be included or excluded. In addition, the unit of measure should be specifically specified.

As an example, if the measure specifies that the faults which have arisen during the acceptance tests must be measured, then there should be a clear specification of what is considered to be a fault (e.g. any malfunction with regard the specifications of the software product detected during the execution of the product) and what kind of faults are taken into account for the measure: faults caused by operator errors are not considered, faults caused by errors in software documentation are considered, faults caused by hardware defects are not considered, faults which appear during the field test phase are not considered, etc.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: Indicators
Output	MP: Measures Specification
Guidelines	MIS-PyME Indicator Templates

Table 4-8. MIS-PyME Methodology – DEF 4.1

DEF 4.2. Specifying the collection procedures through which to obtain the measure results. The procedures required for collecting the data for each measure is specified.

The organization's ability to obtain the measures is analysed, and the way in which they could be collected is established. As an example, if the measure specification is to measure the faults detected during the acceptance tests, a specific query should be launched to the defect tracking tool. In this task, the specific query or other steps which are necessary to obtain the data are specified.

Once the data collection has been determined, the operations that should be performed in order to obtain the measure results should also be specified (e.g. percentage calculation, a division between the data collected, adjust or weight any data, etc.). This specification should be included with the measure specification described above.

If it is not possible to collect the desired data, or it is quite complicated since software and management tools cannot be easily adapted to obtain that specific data, the indicator specification may be modified based on this information. If it is possible to collect the data using specific tools and the measurement goal merits the expense of obtaining or implementing the tool (e.g. static analysis of the software quality), then an evaluation of the tool should also be performed during this step.

In addition, the procedure to collect the measure should, perhaps, be defined in several ways since these measures can be found in various contexts (different program languages, different procedures involved, different systems, etc).

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: Measures Specification
Output	MP: Measures Specification (including collection procedures)
Guidelines	MIS-PyME Indicator Templates

Table 4-9. MIS-PyME Methodology – DEF 4.2

DEF 5. Integrating the measurement program.

The aim of this activity (Figure 4-8) is to integrate the measurement activities into previous measurement processes and into the organization's other software processes (e.g. specific project administration process).

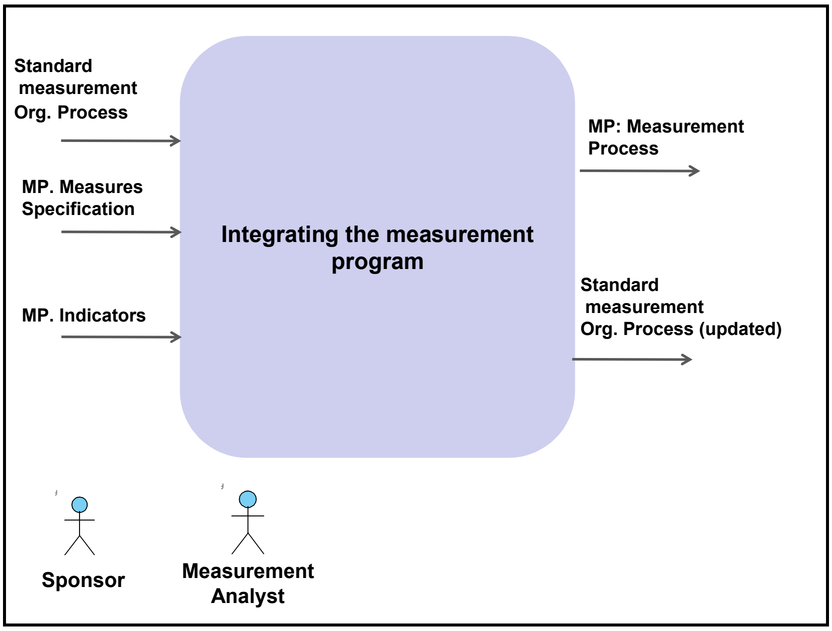


Figure 4-8. MIS-PyME Methodology DEF 5

DEF 5.1. Integrating the measurement program into the standard measurement processes. The purpose of this task is to integrate the measurement program into a measurement process.

The indicators and measures defined for the measurement program must be integrated into a specific measurement process. The measurement process specifies the measurement activities and tasks that have to be performed in order to carry out the measurement program goals. MIS-PyME provides a generic process which will serve as a "template" (see Appendix D).

The measurement process should be adapted to the organizational process in which it is to be applied. For example, a measurement process for the end-project activity may be defined in order to determine the indicators that have to be analysed in this project management process activity. Therefore, the roles of the "person responsible for measurement" and the "measurement data collector" may be specified to the process in which they are to be used, e.g. the role would be the project manager for the measurement process performed at the closing project activity.

The indicators and measures specified for the measurement program will be defined as "inputs" of the measurement process. The measurement process may have already been defined in the company, and if not, one should be created.

Not all the measures and indicators of the same measurement program have to be related to one measurement process. As an example, if the measurement goal of the measurement program is to improve project planning, the indicators and measured defined for the measurement program can be related to a measurement process for a specific project and others can be related to the Portfolio Project Management Process.

Main Role:	Measurement Analyst
Input	MP: Measures Specification MP.Indicators Standard measurement processes of the organization
Output	Standard measurement processes of the organization (updated) and/or MP: Measurement process (new)
Guidelines	MIS-PyME Indicator Templates

Table 4-10. MIS-PyME Methodology – DEF 5.1

DEF 5.2. Integrating the measurement program into the organization's other processes. The purpose is to integrate the indicator results, and thus the analysis, interpretation and improvement suggestions or decisions, into the organization's process reports.

The indicator results, and thus the indicator base data (and the graphic display), the indicator analysis and interpretation, and the improvement suggestions and the decisions made, are included in the organizations' process reports. As an example, the indicator results of the measurement process for the close-project activity can be included in the close-project report, the indicator results of the specific project measurement process can be included in the follow-up report, etc.

If there is a document template for each of the organization's process reports, these should be updated in order to include the specific indicator information.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: Measures Specification MP.Indicators Standard processes of the organization
Output	Standard processes of the organization (updated)
Guidelines	MIS-PyME Indicator Templates

Table 4-11. MIS-PyME Methodology – DEF 5.2

DEF 6. Measurement program verification

The measurement program is verified by a certain key interested party. An introductory session is held in order to present the measurement program and its integration into the other software processes, including the changes in the existing measurement process (Figure 4-9).

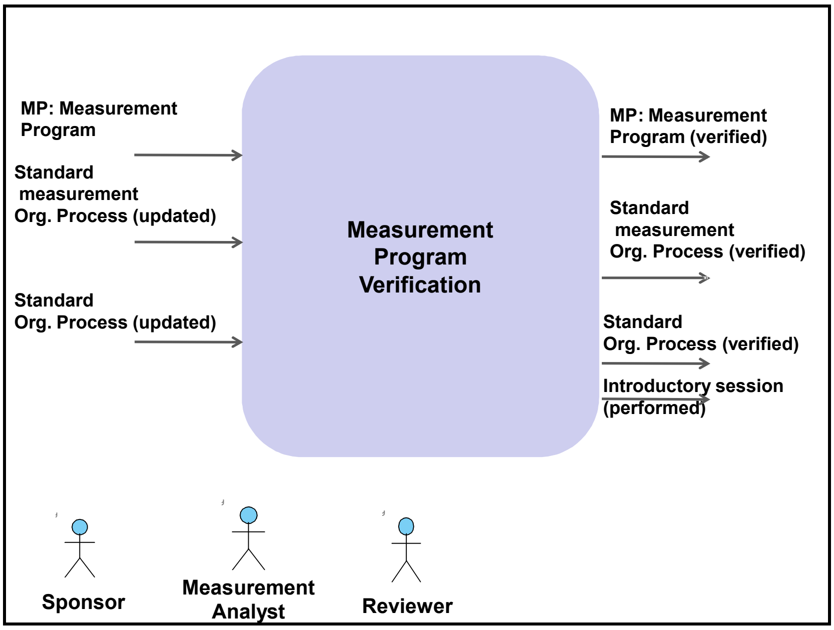


Figure 4-9. MIS-PyME Methodology DEF 6

DEF 6.1. Introductory session on measurement program verification

A meeting is held by the sponsor, the measurement analyst and the reviewers. The reviewers usually play the role of project managers and are responsible for the development process or the portfolio project management process. The number of reviewers depends on the importance of the measurement program, but one or two reviewers are recommended.

During the meeting, the measurement analyst and the sponsor should explain the following:

- Measurement goals of the measurement programs.
- Indicators defined for the measurement program in order to achieve the goals.
- The measures used to build the indicators.
- The measurement process created for this measurement program or the modified measurement process. How these measurement processes are to be integrated with the other company processes, i.e. when the measurement process will be performed depending on the company's other processes.
- The templates of the company's processes which have been modified to include measurement information.

Main Role:	Measurement Analyst
Additional Roles	Sponsor, reviewers
Input	MP: Measurement Program Standard processes of the organization (updated) Standard measurement processes of the organization (updated)
Output	Introductory Session (performed)

Table 4-12. MIS-PyME Methodology – DEF 6.1

DEF 6.2. Reviewing the measurement program. The reviewers check the measurement program, which is modified accordingly.

The reviewers check the measurement program. Those issues which should be checked are as follows:

- The indicators are suitable for achieving the measurement goals. The questions stated in the indicator guide the attainment of the information required to obtain the measurement goal.
- The measures specified for building the indicator are relevant and sufficient to answer the questions.
- The data collection specifications for obtaining the result of the measures can be performed, and are complete and easy to understand.
- The measurement process is suitably integrated into the organization’s other processes.
- The organization process report templates are correctly updated in order to include the measurement information.

After the reviewers have checked these points, a meeting is held (if necessary) by the measurement analyst and the reviewer in order to understand the errors detected in the measurement program definition and to understand the suggestions made.

The measurement program is updated based on the results of the reviews performed by the reviewers

Main Role:	Reviewers
Additional Roles	Sponsor, Measurement Analyst
Input	MP: Measurement Program Standard processes of the organization (updated) Standard measurement processes of the organization (updated)
Output	MP: Measurement Program (verified) Standard processes of the organization (verified) Standard measurement processes of the organization (verified)

Table 4-13. MIS-PyME Methodology – DEF 6.2

DEF 7. Instrumentation

Tools which support measurement process are developed or tailored (Figure 4-10).

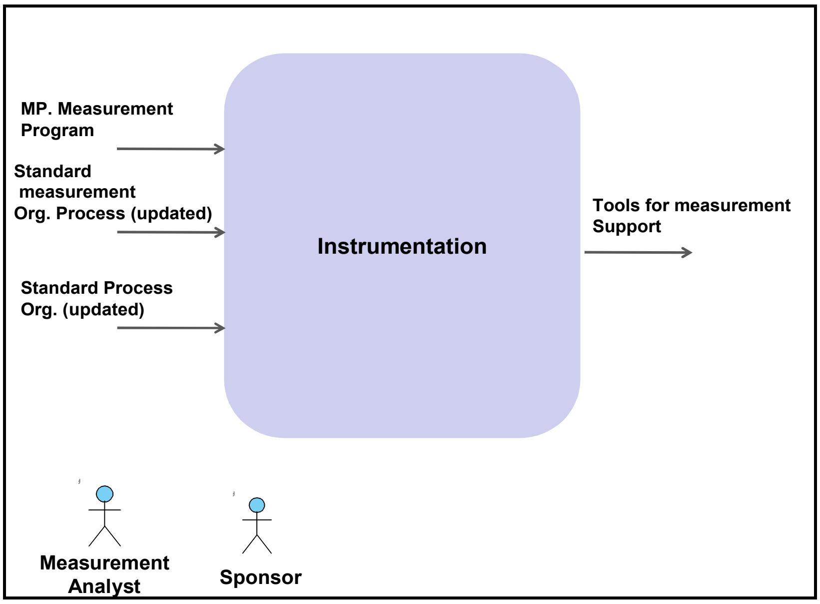


Figure 4-10. MIS-PyME Methodology DEF 7

DEF 7.1. Instrumentation. The purpose is to develop or adapt the required tools in order to automate the measurement program performance as far as possible.

Databases for gathering the data collected which are required to obtain the results of the measures are usually needed. In addition, the results of the measures are usually kept in databases. A database should therefore be constructed in this task in order to support the measurement activities. Excel Spread Sheets are useful when the company is starting with measurement.

Other tools may also be adapted in order to create the automatic queries which are required to obtain the data, or to represent any aspect of the product, process or project that was not considered.

MIS-PyME recommends that before obtaining powerful tools it is better to understand the process and to control the essential activities. Furthermore, the benefits that the tool provides may not make up for the cost of evaluating the tool and training people. Other more powerful tools can be acquired once the company is sufficiently mature.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: Measurement Program Standard processes of the organization (updated) Standard measurement processes of the organization (updated)
Output	Tools for measurement support

Table 4-14. MIS-PyME Methodology – DEF 7.1

DEF 8. Acceptance

The measurement program is used in a trial analysis, or in a real project (selected as a pilot project) if the measurement program was focused on a specific project measurement need. The measurement program is thus used as if it were already implemented. Those people involved in the trial make suggestions regarding the usefulness and the correctness of the measurement program and the tools. After the required modifications, the measurement program is accepted and a brief training session for all those who are directly involved or who may be involved in the future takes place (Figure 4-11).

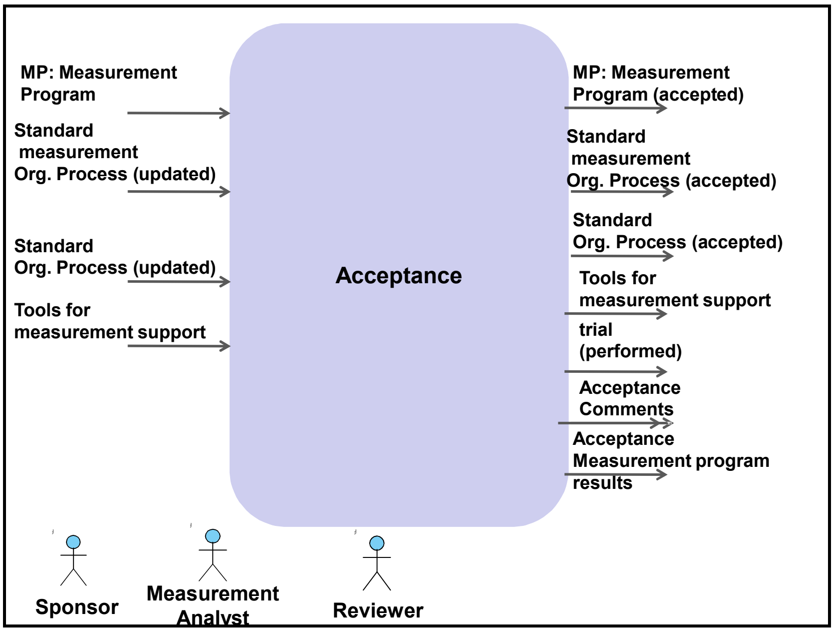


Figure 4-11. MIS-PyME Methodology DEF 8

DEF 8.1. Trial of the measurement program. The purpose of this task is to test the measurement program in a real scenario

The measurement program is used in a real scenario. This range of possible scenarios may be extremely wide. For example, the aim of the measurement program might be to use the measurement program to track projects and the purpose might be to apply the measurement program to all the projects in the company. Therefore, a small project with no time constraints could be selected as a pilot project. The measurement program is used to track the project and, based on experience, the indicators, measures and templates are updated. However, if the measurement program deals with cross-project analysis, the first time that this cross-project analysis is performed will be in this trial.

The people involved in the trial should first be trained. The main points of this training lie in the following issues:

- The global aim of the measurement program (process improvement goals).

- The indicators and measures defined in order to achieve the goal.
- The measurement processes updated or created in order to carry out the measurement tasks, and how these measurement processes are integrated into the organization's other processes.
- The roles involved in the measurement process in relation to the roles in the organization's processes into which the measurement process is integrated.
- The importance of their role and how the information they produce is used by other people. The consequences of their not working well could be also explained.
- The activities each person has to perform and where to obtain the information.
- The use of the support measurement tools.
- The measurement trial work plan and how to report the improvement suggestions or defects for the measurement program.

Once this training has been performed the trial can begin. At this time, defects will be identified or suggestions for improvement will be proposed. In addition the results of the analysis will be checked in order to understand if the questions correctly guide the analysis, and if the results are useful for achieving the goals. The measurement program will be updated accordingly by the measurement analyst.

.

Main Role:	Reviewers
Additional Roles	Measurement analyst
Input	MP: Measurement Program Standard processes of the organization (updated) Standard measurement processes of the organization (updated) Tools for measurement support
Output	MP: Measurement Program (accepted) Standard processes of the organization (accepted) Standard measurement processes of the organization (accepted) Acceptance comments Acceptance measurement program results Trial (performed) Tools for measurement support (accepted)

Table 4-15. MIS-PyME Methodology – DEF 8.1

DEF 8.2. Closure of the acceptance task The purpose of this task is to formally present the measurement program as implemented and accepted, and to train the other people involved.

The last task in the definition and implementation of the measurement program is to explain the use of the measurement program to the people in the company. A kick-off meeting should take place. Many people may have already being trained owing to their participation as reviewers.

The main points of this short training should be:

- To explain the global aim of the measurement program (process improvement goals)
- To explain the measurement goals which have been defined and implemented in order to support those process improvement goals
- To explain the relations between the measurement goals, indicators and measures defined which summarize the measurement program definition.
- To explain the measurement processes defined, or into which measurement processes the indicators and measures defined have been integrated, and when the measurement processes have to be performed.
- To explain the organization's process report templates which have been modified in order to include new measurement information.
- To explain the roles involved in the measurement process in relation to the roles in the organization's processes into which the measurement process is integrated.
- To explain the importance of their role and how the information they produce is used by other people. The consequences of their not working well could be also explained.

Main Role:	Measurement Analyst
Additional Roles	Sponsor
Input	MP: Measurement Program Standard processes of the organization (updated) Standard measurement processes of the organization (updated)
Output	kick-off training (performed)

Table 4-16. MIS-PyME Methodology – DEF 8.2

4.2.3. MIS-PyME Support Modules

The support modules aim to assist the measurement analyst in developing the measurement program from a software process improvement goal. This provides common measurement goals which support the achievement of the process improvement goal. In addition, MIS-PyME provides the indicators related to each of these measurement goals and the examples of the indicators which assist the measurement analyst to determine the required analysis and the data to perform the analysis, how this analysis will be integrated into the organization's other processes and for what reason, etc. The use of support modules permits the measurement program to be developed in an easier, more reliable manner, and with less effort.

MIS-PyME Measurement Goals Table: The measurement goals of each measurement program defined in the company should be stated in the process management process since this aims to improve and manage all the processes in the company in order to align them to the business goals stated in the business process. This process identifies the required information that should be obtained in each process in order to manage and control the processes and be able to take action in order to meet business goals. The MIS-PyME framework proposes a set of measurement goals usually required to achieve process improvement goals. The process improvement goals specified in the table are the process improvement goals which are commonly required in a software development and management company. These process improvement goals are referred to in the COMPETISOFT software process model. This table is a useful guide for a company when identifying the measurement goals that will help to improve certain required and identified aspects of the process. The measurement goals specified in the table are focused on the aspects of the process, product or project that are more frequently required in a company and the table does not, therefore, cover all the aspects of these entities but only those which are most common. The table consists of 5 columns which are as follows:

- Column 1: This column shows the one of the processes of the management process level of COMPETISOFT, i.e.: resources management, knowledge management or goods, services and infrastructure management processes. The measurement goals defined in its associated rows will be related to the process indicated in this column. In other words, the measurement program whose process improvement goals are related to this column may be required to improve these processes.
- Column 2: This column shows the COMPETISOFT Level 3 process, i.e.: specific software administrator, software development and software maintenance processes to which the measurement goals are related. This signifies that the measurement program whose process improvement goals are related to this column may be required to improve these processes.
- Column 3: Process Improvement Goals. The process improvement goals commonly required in the related processes (those stated in columns 1 and 2) are indicated.
- Column 4: The measurement goals commonly required to support the process improvement goals indicated in column 3 are specified.
- Column 5: The indicator templates related to each of the measurement goals are indicated. The indicator template aims to develop the definition of the measurement program for the related measurement goal.

Details of the MIS-PyME Measurement Goals Table are shown in Appendix B.

MIS-PyME Indicator Templates: An indicator template is defined for each measurement goal. The indicator template will guide users in developing the measurement program based on the measurement goals.

The name of the indicator is formed of *IND-ENTITY-ABBREVIATION*:

- IND: This label is always written in order to identify it as an indicator.
- Entity: Determines the entity to be measured: project (PRJ), (PROC), (PROD) or resource (RES). If the indicator is required to perform a cross analysis (e.g. cross-project analysis) the word ORG is placed at the end (e.g. PRJORG).

- Abbreviation: The abbreviation of the aspect of the entity that is to be measured is indicated. For example "RELIABILITY", "EFFICIENCY", "DURATIONDEV" (deviation of the duration of the project vs. planned), etc.

The indicator template provides information regarding the measurement maturity required in order to implement the indicator. The name of the indicator is coloured in accordance with the measurement maturity required for this indicator (Level 1- yellow, Level 2-blue, Level 3-pink).

It also contains thorough information with which to specify each of the fields required for the indicator: point of view, context, questions, inputs, algorithms, improvement of the indicator, measurement activity information, resources required, analysis and output display.

The MIS-PyME Indicator Templates are shown in Appendix C.

MIS-PyME Indicator Database: MIS-PyME aims to provide an example for each of the indicator templates of how these indicators were tailored to and implemented in a company. In short, the Indicator Database provides a set of tailored indicator templates which have been used in companies. Each of these indicators provide an example of one of the analyses performed, including the interpretation results, decisions made based on the indicator results, etc.

4.3. MIS-PyME Measurement Maturity Model

The aim of the MIS-PyME Measurement Maturity Model (MIS-PyME-3M) is to help the company to understand its measurement maturity and identify improvements, and to guide the measurement analyst in understanding whether the capability of the measurement process is sufficient to implement a certain measurement goal. MIS-PyME-3M is compliant with COMPETISOFT.

The MIS-PyME-3M design was focused on fulfilling the requirements derived from the case study of this research, which consisted of applying GQ(I)M to define a measurement program in the Development division of STL (see Chapter 6). The requirements are as follows:

1. 3M-REQ1: The measurement maturity model must explicitly indicate the criteria which decides whether a process is in one level or another.
2. 3M-REQ2: The measurement maturity model must explicitly indicate the information needs that the measurement program is able to satisfy for each maturity level.
3. 3M-REQ3: The measurement maturity model must explicitly indicate the support tools and other resources of the measurement process that must be implemented for each maturity level.

4. 3M-REQ4: The measurement maturity model must explicitly provide a process with which to assess the measurement maturity level of the organization.
5. 3M-REQ5: the measurement maturity model used to define measurement programs adapted to the measurement maturity of the company must be compliant with the international standard ISO/IEC 15504.

The indicators are used as a basis for collecting the objective evidence that enables an assessor to assign ratings. MIS-PyME-3M comprises a set of indicators. These indicators are classified in three views:

- Aligned with 3M-REQ1, MIS-PyME-3M defines performance and capability indicators (P). These indicators aim to assess the extent to which the outcomes of the measurement process are achieved.
- Aligned with 3M-REQ2, MIS-PyME-3M defines Input indicators (I): These indicators depend on the measurement indicators of the measurement process. They assess the type of measurement performed as regards the purpose, the focus and the entity which are measured.
- Aligned with 3M-REQ3, MIS-PyME-3M defines Resources (R): These indicators are based on the tools and infrastructure that support the measurement process.

The degree of fulfilment of each indicator is evaluated by means of three types of evidence:

- Direct: these are the products which are the result of an activity.
- Indirect: these are documents which indicate that an activity has been carried out.
- Comments: these are opinions given by those people who are involved in the process that is being evaluated.

MIS-PyME also defines an assessment process which is a specialization of the process provided in COMPETISOFT – PvalCOMPETISOFT, and since the model complies with COMPETISOFT it therefore complies with ISO/IEC 15504. Appendix F shows how MIS-PyME 3M complies with ISO/IEC 15504.

In addition, the same indicators are classified in two categories: the process performance indicator which applies to level 1 and capability indicators which apply to all the capability levels of the model. Process performance indicators determine whether the purpose of the measurement process is fulfilled. Process capability indicators determine the establishment of the process, the measurement capability as regards the information needs that the measurement process is able to carry out (input indicators), and the support tools of measurement process.

MIS-PyME-3M therefore determines the following:

- The capability levels. These capability levels will be aligned with COMPETISOFT. However, it is not expected that SMEs will achieve a measurement maturity level of 4 or 5. Therefore, only 3 levels, from levels 1 to 3, will be defined: performed process (Level 1), managed process (Level 2), and established process (Level 3).
- The attributes and indicators for assessing the capability of the measurement process, measurement indicators (I) and support resource (R), which are based on the Daskalantonakis et al. model (1990), the measurement evolution described in CMMI (Team, 2002; Weber and Layman, 2002) and the Niessink and Van Vliet (1998) capability model.
- A questionnaire (see Appendix E) used by the assessment module in order to determine its capability level and guide the measurement analyst in defining measurement programs adapted to the company's measurement maturity.

4.3.1. MIS-PyME-3M - Levels, Attributes and Indicators

This section shows the levels defined in the MIS-PyME measurement maturity model, the assessment attributes; and the process performance, process inputs and the tool indicators that are used to assess the attributes.

Level 1: Performed Process

The implemented process achieves its process purpose. Basic measurement processes are in place to collect and analyse the measures and provide feedback to software engineers and management. At this level, the measurement process is usually carried out with experienced people.

The following attributes of the process demonstrate the achievements of this level:

- PA 1.1 Process performance attribute: The process performance attribute is a measure of the extent to which the process purpose is achieved. As a result of a full achievement of this attribute, the process achieves its defined outcomes:
    a) Data is collected and processed: The data required to generate the measure result is collected. The indicator is built based on the measure results
    b) Indicators are analysed: The indicator base data are analysed and interpreted, questions are answered, and decisions are therefore made in accordance with the indicator goal.
    c) Feedback of the measurement results takes place: Analyses and interpretation results are communicated, reviewed and updated if necessary

- PA 1.2 (I) Basic project and product focus information attribute: this attribute measures the extent to which the basic information needs are achieved. As a result of full achievement of this attribute:
    a) Tracking project schedule/plan (phase by phase): The measurement process tracks the project schedule phase-by-phase versus plans and takes reactive action in the case of problems.
    b) Tracking critical product reliability in production: The measurement process tracks the reliability of the organization's main products in production based on defects, and takes reactive action in the case of problems.
- PA 1.3 (R): Basic management tools implemented attribute. This attribute measures the extent to which the measurement process is supported by basic software management tools. As a result of the achievement of this attribute:
    a) Incidents tool (production): Database tools are established in the organization to store and track the incident of products in production.
    b) Basic project management tool: Project management tools related to schedule, effort and cost tracking are established in the organization.

Level 2: Managed process.

The previously described Performed process is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained. The following attributes of the process, together with the previously defined attributes, demonstrate the achievement of this level.

- PA 2.1 (P) Performance management attribute: The performance management attribute is a measure of the extent to which the performance of the process is managed. As a result of full achievement of this attribute:
    a) Objectives (process improvement goals) for the performance of the process are identified.
    b) Performance of the process is planned and informally monitored.
    c) Performance of the process is adjusted to meet plans.
    d) Responsibilities and authorities for performing the process are understood, assigned and communicated.
    e) Interfaces between the involved parties are managed to ensure both effective communication and a clear assignment of responsibilities.
- PA 2.2 (P) Workproduct management attribute: This attribute measures the extent to which the work products produced by the process are appropriately managed. As a result of full achievement of this attribute:

a) requirements for the work products of the process are defined: the definition of what work products should be obtained takes place.

b) requirements for documentation and control of the work products are defined: If necessary, the specification of where the measure results and collected data are stored and documented takes place. The same occurs with the analysis results, improvement suggestions and analyses that result from the measurement activity.

c) work products are appropriately identified, documented, and controlled: work products are appropriately collected, documented and communicated to those people that may be interested in them. Work products are effectively used and managed to achieve the goals, such as taking corrective actions in projects, etc.

d) work products are reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements: measure results, indicator analyses and interpretation, decisions and improvement suggestions, etc. are appropriately verified and adjusted if necessary.

- PA 2.3 (I) project and product focus management attribute: this attribute measures the extent to which the process purpose achieves the basic project and product management. As a result of full achievement of this attribute:

  a) Manage planning information: The measurement process is able to understand partial and total deviation from the project versus plans in terms of cost, effort and duration. These data are used when estimations of new projects are performed in order to plan them.

  b) Manage reliability of the products during development: The measurement process tracks the reliability of the main products being developed based on defects.

  c) Manage customer satisfaction: Customer satisfaction is measured

- PA 2.4 (R): management and development tools implementation attribute. This attribute measures the extent to which the measurement process is supported by software tools such as defect and incident tracking tools, project management tools etc. and other necessary resources, methods and information. As a result of the achievement of this attribute:

  a) Defect tracking tools: database tools to store and track defects are established in the organization

  b) Project management tools for tracking effort and cost are in place.

Level 3: Established process. The previously described Managed process is now implemented by using a defined process which is capable of achieving its process outcomes. The measurement process is well understood in the organization. All projects use a tailored version of the organization's standard measurement process, and the process is carried out with the necessary frequency.

The measurement processes are well integrated into the other software processes.

- PA 3.1(P) Process definition attribute. The process definition attribute is a measure of the extent to which a standard process is maintained to support the deployment of the defined process. As a result of full achievement of this attribute:
    a) A standard process, including appropriate tailoring guidelines, is defined which describes the fundamental elements that must be incorporated into a defined process. For example, it may identify the mandatory and optional indicators that should be analysed during the process.
    b) The sequence and interaction of the standard process with other processes is determined. The standard measurement process is completely integrated into the other software development, management and quality processes. The collected data are therefore integrated into people's normal work when they are carrying out the development, quality or management processes. The measurement reports may be integrated into the project review/monitoring results report, into the close of project reports, etc.
    c) Required competencies and roles for performing a process are identified as part of the standard process;
    d) The necessary infrastructure and work environment for performing a process are identified as part of the standard process. The standard process identifies from where and how the measure result data are collected, where the indicator and measure results are located, which tools are used to analyse certain indicators, etc.
    e) Suitable methods for monitoring the effectiveness and suitability of the process are determined. These methods are used to review whether measurement data are collected and analysed as specified, or whether the results are communicated as specified and to what extent the action plan and improvement suggestions are carried out and the measurement process is eventually useful (measurement process management practice - evaluating the measurement program).
- PA 3.2 (P) Process deployment attribute. The process deployment attribute is a measure of the extent to which the standard process is effectively deployed as a defined process to achieve its process outcomes. As a result of full achievement of this attribute:
    a) A defined process is deployed based upon an appropriately selected and/or tailored standard process.

b) required roles, responsibilities and authorities for performing the defined process are assigned and communicated

c) The personnel performing the defined process are competent with regard to appropriate education, training, and experience.

d) Appropriate data are collected and analysed as a basis through which to understand the behaviour of the process, to demonstrate the suitability and effectiveness of the process and to evaluate where continuous improvement of the process can be made.

- PA 3.3 (I) Advanced project tracking attribute: this attribute measures the extent to which the purpose of the process is to apply standard measurement programs related to projects in an advanced fashion. As a result of full achievement of this attribute:

    a) Cross-project analyses availability: Since there is a standard measurement process, cross-project analyses are available which can be used to identify improvements that can be implemented in the whole organization.

    b) Manage development progress: development progress is managed by using measurement results as #requirements or Use Cases performed vs. planned.

    c) Requirements stability is managed.

- PA 3.4 (I) Advanced product tracking attribute: this attribute measures the extent to which the product is tracked in an advanced fashion. As a result of full achievement of this attribute:

    a) Manage quality information: The company starts to understand other quality aspects such as cyclomatic complexity, repeated code, inheritance levels, modules/classes dependability etc. There is an understanding of the maintainability of the products.

    b) Cross-product analysis availability: since there is a standard measurement process, cross-product analyses are available which can be used to identify common causes of problems that these analyses may derive from improvement actions which can be implemented in the whole organization.

- PA 3.5 (I) Process tracking attribute: this attribute measures the extent to which the process purpose manages processes. As a result of full achievement of this attribute:

    a) Process efficiency and effectiveness: aspects of the process (maintenance and development process) are measured as regards efficiency, effectiveness. The effectiveness of the test phases can therefore be analysed and the productivity of the construction phase can also be discovered.

    b) Productivity normal ranges: normal ranges as regards productivity are known.

    c) Manage the process compliance: the process is also measured in terms of compliance implemented in the whole organization.

4 Fix defect time: The measurement process tracks the time it takes to fix a failure in production, and normal ranges in terms of the time needed to resolve failures in production are understood.

d) Downtime: The measurement process tracks the downtime in production caused by failures, and maintenance actions and normal ranges of downtime are understood.

- PA 3.6 (R) Resources deployment attribute

  a) Organization Measure Database: Collected measures are stored in an organization-wide database and made available.

  b) Life cycle configuration management tool: There is a life cycle configuration management tool.

  c) Training Program: People are provided with the skills and knowledge needed to perform their roles.

  d) Static quality analyser tools: tools required to perform static analyses of the software are used.

  e) Resource Management: resources received by the provider are managed in terms of reception and problem resolution timing, etc.

  f) Project estimation techniques: project estimation techniques are used to plan the effort and scheduling of projects. However, these results are not yet completely reliable.

  g) Procedures in the use of these tools are well understood and are standardized throughout the organization: people know how to introduce the information, what that information means, and most people do this according to the procedures.

4.3.2. MIS-PyME 3M Assessment Process

The measurement assessment model aims to assess the measurement process based on the levels and attributes defined in the MIS-PyME measurement maturity model (see Figure 4-12).

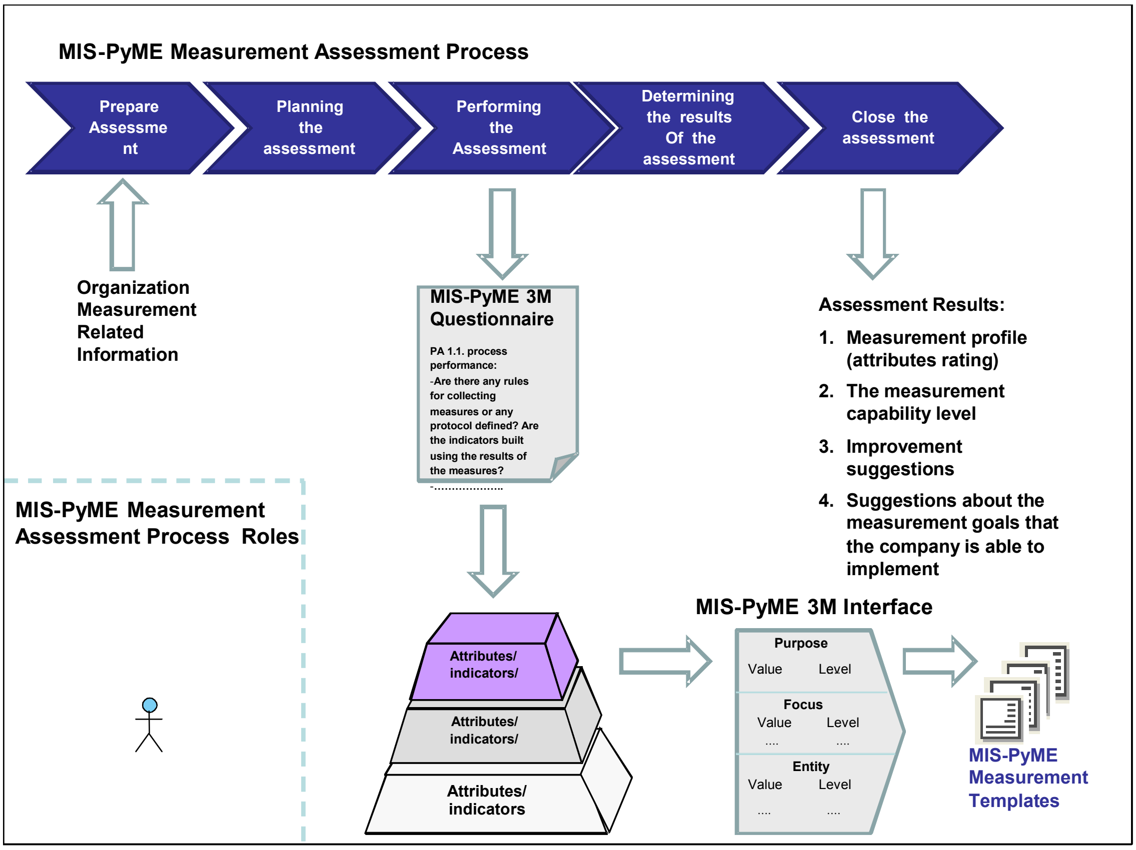


Figure 4-12. MIS-PyME 3M

The assessment may be used for self-assessment, when the organization wishes to know its measurement capability maturity level in order to detect improvement areas and to identify which measurement goals the organization is able to implement as regards its maturity.

With regard to the rating values, the extent to which the attribute of the measurement process is fulfilled is based on the following values set in COMPETISOFT: not achieved, partially achieved, largely achieved and full achieved:

- N: Not achieved: There is little or no evidence of achievement of the defined attribute in the assessed process.
- P Partially achieved: There is some evidence of an approach towards, and some achievement of, the defined attribute in the assessed measurement process. Some aspects of the achievement of the attribute may be unpredictable.
- L Largely achieved: There is evidence of a systematic approach towards, and significant achievement of, the defined attribute in the measurement process. Some weakness related to this attribute may exist.
- F Fully achieved: There is evidence of a complete and systematic approach towards, and full achievement of, the defined attribute in the measurement process. No significant weaknesses related to this attribute exist.

Therefore, the capability measurement maturity level will be that in which the attributes of that level are largely or fully achieved and the attributes of the lower levels are fully achieved (see Table 4-17).

Scale	Process Attributes	Rating
Level 1	Process Performance	Largely or fully
	Basic project and product focus information attribute	Largely or fully
	Basic management tools implemented attribute	Largely or fully
Level 2	Process performance	Fully
	Basic project and product focus information attribute	Fully
	Basic management tools implemented attribute	Fully
	Performance management	Largely or fully
	Work product management	Largely or fully
	Project and product focus management attribute	Largely or fully
	Management and development tools implementation attribute	Largely or fully
Level 3	Process performance	Fully
	Basic project and product focus information attribute	Fully
	Basic management tools implemented attribute	Fully
	Performance management	fully
	Work product management	fully
	Project and product focus management attribute	fully
	Management and development tools	fully
	implementation attribute.	fully
	Process definition attribute	Largely or fully
	Process deployment attribute	Largely or fully
	Advanced project tracking attribute	Largely or fully
	Advanced product tracking attribute	Largely or fully
	Resources deployment attribute	Largely or fully

Table 4-17. ISO/IEC 15504 Capability Level Ratings

MIS-PyME 3M Assesment Process assesses the capability maturity of the measurement process. This process is a specialization for the measurement of PvalCOMPETISOF. Appendix E shows this process in detail

During the execution of the assessment, the questionnaire shown in Appendix E is used to determine the extent to which the attributes are achieved. The questionnaire contains a set of questions for each attribute specified in MIS-PyME 3M. The answer to each question may be simply “yes” or “no”. If the answers to 0 to 15% of the questions are “yes” then the attribute is not achieved. If 16 to 50% of the questions are “yes” then the attribute is partially achieved. If

51 to 85% of the questions are "yes" then the attribute is largely achieved, and if the answers to 86-100% of the questions are "yes", then the attribute is fully achieved. If a question cannot be answered, since it is not applicable to the context, then this question is not taken into account.

The most important outputs of the process are the following:

- the set of profiles of the measurement process based on the attributes rating,
- the maturity capability level of the organization (N),
- the suggested measurement goals that the company is ready to implement according to its maturity.
- The set of issues to be improved, which will be the input of the measurement improvement process.

4.3.3. Integrating MIS-PyME 3M and MIS-PyME Methodology

The MIS-PyME measurement maturity model is mainly required during the indicator definition activity (DEF 3) of MIS-PyME metholodogy. When the measurement analyst defines the indicators, s/he will be supported by the corresponding MIS-PyME indicator template. This template includes (amongst other things) recommendations for measurement maturity in terms of indicator implementation. These recommendations come from the MIS-PyME measurement maturity model. Therefore, each of the indicator templates includes information concerning the required measurement level that the company needs in order to be able to successfully implement the indicator.

Some indicator fields depend on the maturity of the company as regards measurement. These fields are particularly some of those which determine the goal of the indicator. Purpose, Entity and Focus.

Purpose: The reason why this field determines the measurement maturity level required for implementing an indicator is as follows: characterization signifies the understanding of an aspect of the project, product or process. This purpose may be facilitated by collecting data, obtaining the results of the measures and indicators, and analysing the results. However, monitoring requires a fixed frequency for performing the same measurement activity, comparing results, storing the results, etc. It also requires a more formal activity and the measurement process must be more mature. Evaluation may require some knowledge of the normal ranges of the aspect to be evaluated in order to be able to perform a reliable evaluation. Normal ranges require the habit of performing the measurement activity in the same manner, and also necessitate that the results of the measurement activity are maintained for a time which may be quite long depending on the aspect to be evaluated. Therefore, if the company does not comply with these requirements it cannot implement such measurement goals. Prediction, which is not suggested for use in MIS-PyE, already requires a higher measurement maturity level since measurement data must be reliable and rigorously collected for the time required in order to be able to identify relationships between dependent and independent attributes. Control and change (neither suggested for use in MIS-PyME) require that these relationships should be quantitatively managed.

Focus of the indicator: This field of the measurement goal states the particular attribute of the object under study that will be characterized, evaluated, or monitored. Examples of quality focuses are cost, reliability, correctness, defect removal, maintainability, etc. The reason why this field determines the measurement maturity level required to implement an indicator is that there are certain concepts of the project, product or process that are easier to obtain than others. For example, there are some aspects that require measurement tools that may not be established in the company and there are certain terms that may also require a certain maturity level. As an example, the company must be able to understand what the cyclomatic complexity, or the dependability between modules mean in order to understand the quality of the product being developed. It may also require certain tools such as static quality analysers that may be neither understood by nor known in the company. Additionally, it may be impossible to measure certain aspects of the project or product since these aspects are not considered in the company's development or management process, i.e., it is not possible to track integration test effectiveness if test phases have not been differentiated in the correct manner. It is generally easier to implement measurement goals related to basic project tracking, and product information such as defects, and make step by step improvements in order to establish a measurement framework for advanced project management, advanced product quality management, etc.

Entity to be measured: This field determines the required measurement maturity level since the measurement goals whose "entity" field is labelled as "ORG" cannot be implemented until there is a standard measurement process which is correctly applied in different projects, products and processes in order to perform cross-project, product or processes analyses. In addition, measurement related to the process generally requires a more mature measurement level since it is usually more complex to measure process aspects.

Therefore, each of the common measurement goals provided in the MIS-PyME Measurement Goals Table and its related indicator templates indicate the measurement maturity level required to implement that indicator which is based on the measurement goal description. The measurement maturity level of each measurement goal and related indicator is identified by colours (see Table 4-18). These colours match the colours assigned by COMPETISOFT to each maturity level.

Maturity Level	Maturity Color
Level 1	Yellow
Level 2	Blue
Level 3	Green

Table 4-18: MIS-PyME 3M Colours

Tables 4-19, 4-20 and 4-21 form the interface between the MIS-PyME methodology and the MIS-PyME measurement maturity model. These tables highlight the possible values of each of the fields that depend on the company's measurement maturity and the suggested measurement maturity level at which the company should be in order to implement the indicator with that field value.

The measurement analyst should observe the suggested level of the indicator template or, if there is no template, the maturity level of the indicator will be the maximum levels of the fields (purpose, focus and entity). If s/he knows the measurement maturity level of the company it is already possible for him/her to decide whether the indicator is suitable for implementation. If s/he does not know the measurement maturity level or would like more information, s/he should perform at least an informal assessment, and decide whether the indicator is suitable for implementation.

If the company is at level N, MIS-PyME suggests that the indicator is suitable for implementation when the level of the indicator is N or N-1. If the indicator is at level N+1 it could be implemented if the company feels able to successfully implement the indicator, although special care should be taken with the resources required for its implementation.

Purposes	Suggested measurement maturity Level
Characterizing	Level 1
Monitoring	Level 2
Evaluating	Level 3(if the evaluation does not require a previous prediction of a goal, otherwise it would be level 4 and not recommended.
Predicting	Level 4 (not recommended)
Optimizing	Level 5 (not recommended)

Table 4-19. MIS-PyME. 3M Interface as regards Purpose.

Focuses	**Suggested measurement maturity Level**
Quality - reliability in production	Level 1
Quality - reliability during development	Level 2
Quality -maintainability	Level 3
Project Progress - Schedule	Level 1
Project Progress - effort & cost	Level 2
Advanced Resources Project Progress	Level 3
Product Stability - Requirements	Level 3
Process (compliance, effectiveness, efficiency)	Level 3
Client satisfaction	Level 2

Table 4-20. MIS-PyME 3M Interface as regards Focus.

Entities	Suggested measurement maturity Level
Project (PRJ)	Level 1
Product (PROD)	Level 1
Process (PROC)	Level 3
Cross-Project (PRJORG)	Level 3
Cross-Product (PRODORG)	Level 3
Cross-Process (PROCORG)	Level 3

Table 4-21. MIS-PyME 3My Interface as regards entity.

4.4. MIS-PyME Tool

MIS-PyME is a Web tool which eases the use of MIS-PyME methodology. Figure 4-13 shows its home page, and Figure 4-14 shows the functionality of the tool by means of a use case diagram. These main functionalities supported by the tool are:

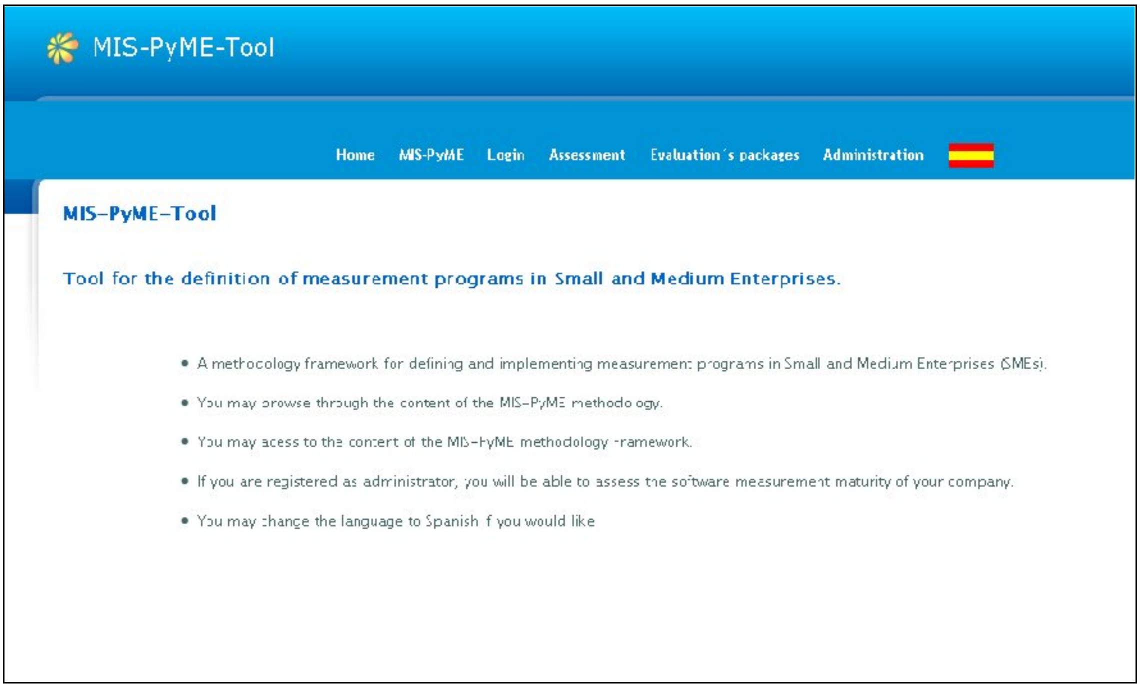


Figure 4-13. MIS-PyME Tool Home Page

WEB access to MIS-PyME Methodological Framework: As has previously been explained, MIS-PyME Framework is defined by SPEM and therefore MIS-PyME Tool provides a link to the Web view of MIS-PyME. Consequently, the user can easily access each of the activities and tasks defined in the process, along with the inputs, outputs, roles and guidelines used in each task. All the elements of the process are directly accessible by a link which is especially useful for accessing to the measurement indicator template related to the measurement goals included in the MIS-PyME Goals Table.

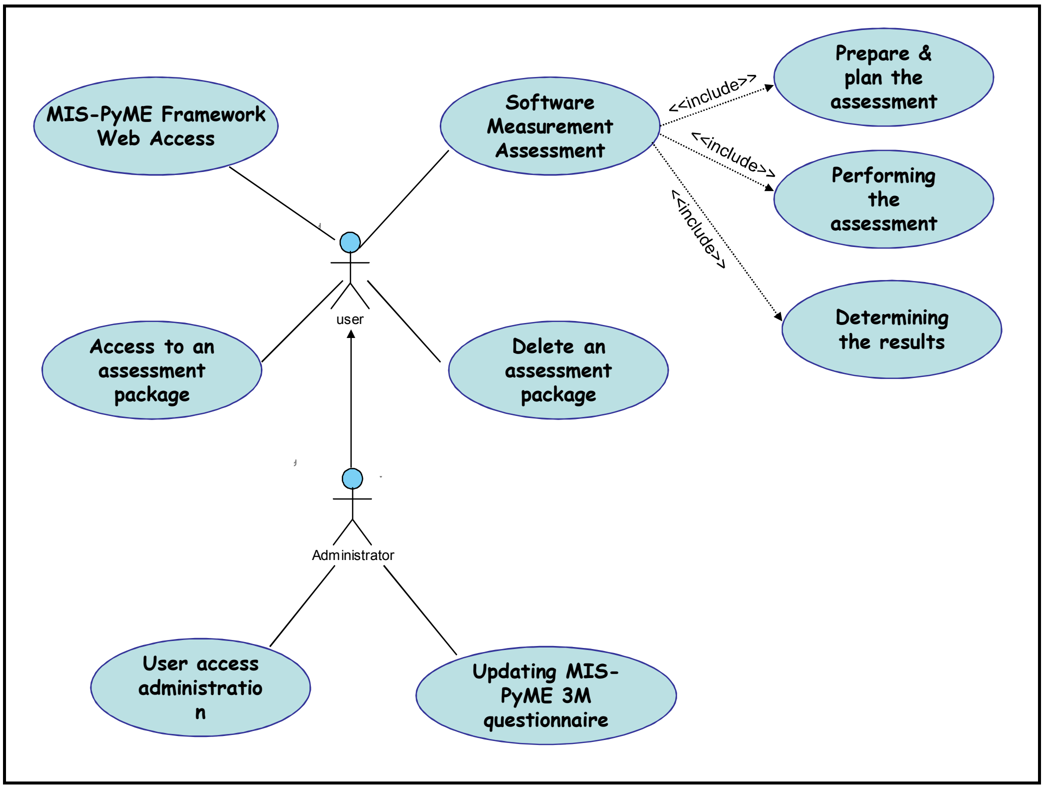


Figure 4-14. Use Cases of MIS-PyME Tool

The assessment process defined for MIS-PyME 3M is also specified in SPEM and its Web view is integrated into MIS-PyME Tool, as is the measurement process template provided by MIS-PyME methodology.

Figure 4-15 shows the Web page of MIS-PyME Framework.

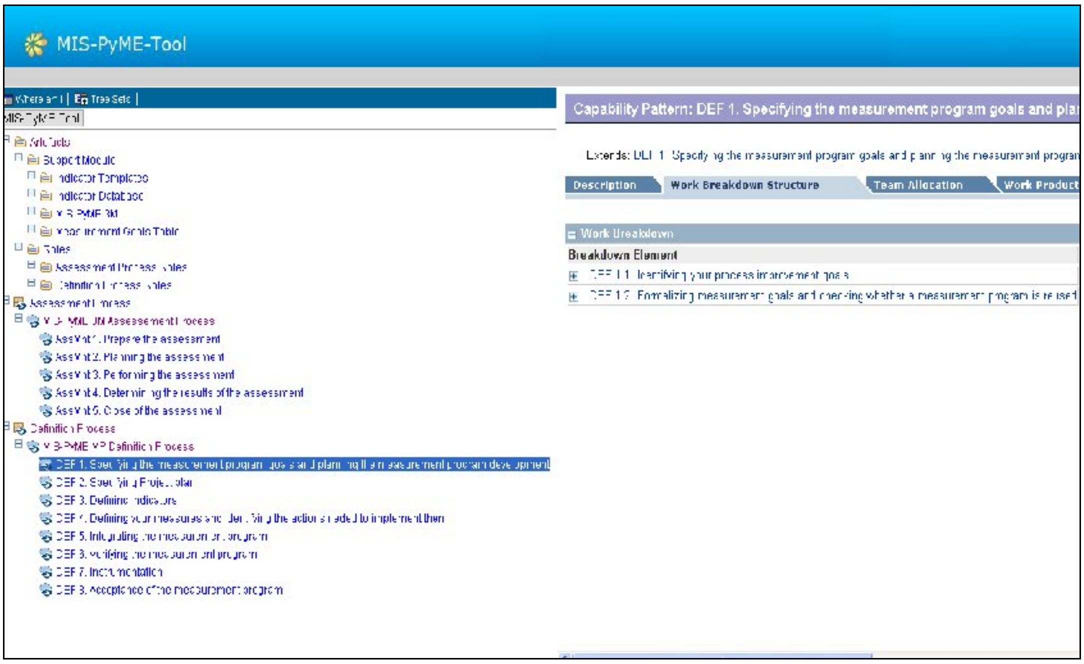


Figure 4-15. MIS-PyME Tool- WEB Access to the Methodology

Assessing the software measurement maturity of the company: The tool provides a mechanism with which to assess the measurement maturity of the company as regards MIS-PyME 3M. When the user chooses to start an assessment, an assessment package is opened in the database provided by the tool, and this guides the user in introducing the information required to state the assessment by following the MIS-PyME 3M assessment process. The tool therefore asks the user for the following information: Scope of the assessment, dates, any restrictions of the assessment, person responsible for and sponsor of the assessment, confidentiality agreement and assessment plan.

Once this assessment phase has been completed, the next step is to fill out the MIS-PyME 3M questionnaire to assess the measurement maturity of the company. The tool therefore shows the user each of the questions for each of the attributes of each maturity level. The assessor should answer yes or no. The user then asks the system to process the answers and determine the measurement capability maturity level of the company. The user is able to include other information such as the improvement measurement suggestions and the final report assessment.

All the information related to the assessment is kept in the database which includes control access.

Access to the assessment packages: The user is able to access an assessment to which s/he has access. If the assessment is not closed, the user can continue the assessment. In other words, the user is able to start an assessment, stop, and continue at another moment, and the system controls the state of the assessment and maintains the introduced data. However, if the assessment is closed the information cannot be modified and the system shows the main information for that assessment: General information (dates, person responsible, etc.), the measurement profile of the company and its measurement maturity level, the improvement suggestions, and the final assessment report.

Delete an assessment package: If the user has the necessary permissions, s/he will be able to delete certain assessments.

Update MIS-PyME 3M questionnaire: In this case, the MIS-PyME 3M questionnaire is improved and new questions are added or others are modified. MIS-PyME allows the administrator to change the questionnaire.

User access management: The system also provides user access control which is managed by the administrator.

4.5. MIS-PyME Contributions

As stated in the previous chapters, SMEs have certain special requirements which must be satisfied by any measurement program which aims to be suited to SMEs' special restrictions. These requirements were defined after the first case study of this research, which consisted of applying GQ(I)M to define a measurement program in the Development division of STL (see Chapter 6).

In Chapter 3: State of the Art, we analysed whether the methodologies for defining and implementing measurement programs- GQM, GQ(I)M, PSM, ISO/IEC 15939 and GQM lightweight method- fulfilled the requirements stated. The result of the analysis was that none of these methodologies fulfilled all the requirements, and a methodology which would comply with them and be suitable for SMEs was therefore required.

MIS-PyME was designed in order to fulfil these requirements. Having presented MIS-PyME we shall now state the reasons why MIS-PyME complies with all these requirements, as follows.

- **Few people involved in the process (FPEOPL):** In MIS-PyME methodology, the measurement program definition is carried out by two people: the measurement analyst and the sponsor of the measurement initiative. The remaining stake holders are involved in the review (one or two reviewers) and acceptance phases. In addition, MIS-PyME describes the roles that should perform each of the tasks for defining and implementing the measurement program, the number of people who play that role and its profile.
- **Reuse measurement models (RUSE):** Task DEF 1.2 of MIS-PyME methodology asks the user to reuse a measurement program which has already been defined and is implemented in the company if possible. In addition MIS-PyME guides the measurement analyst in

defining the measurement program in order to facilitate its reuse. Indicators, measures and measurement processes are all easily reusable elements.

- **Few but effective steps (FSTEP).** MIS-PYME methodology is formed of 8 activities, which cover the entire definition measurement process until its implementation. What makes this methodology more agile than its base methodologies (GQM and GQ(I)M) is the number of people involved in the definition of the measurement program, and that in-depth training is not given to the team but rather a series of short training sessions. This is focused on the re-use of measurement programs and makes use of the support modules that MIS-PyME provides.
- **Specific guidelines to support basic process improvement needs (GPIN).** The MIS-PyME Measurement Goals Table provides the link between process improvement needs and measurement goals which can support those needs. This table contains the most common process improvement goals required in any company that aims to develop and maintain software.
- **Specific guidelines to understand the benefits & potential for management and other guidelines (GB&P).** Each of these measurement goals has its corresponding indicator template which assists the measurement analyst to develop the measurement goal. The questions that will guide the analysis to achieve the measurement goal are therefore proposed. In addition, the measurement and/or indicators that may be required in order to build the indicator are proposed. Information about when and by whom the indicator will be analysed, and who is the interested in the indicator results are also presented. Additionally, a proposal of how the indicator data can be shown in a graphic manner is made. Moreover, the "analysis and interpretation" section of the indicator template provides guidelines regarding the type of analysis that can be performed on the indicator, along with its possible outcomes and interpretations, which also show the potential of the use of measurement for management proposes.
- **Specific guidelines to integrate measurement in the software processes (GINT):** MIS-PyME methodology dedicates an activity to the integration of measurement into the rest of the development process. Each Indicator Template contains a field called "integration" which guides the user in how integrate it into the measurement process. The measurement processes are defined in accordance with the process in which the measurement process will be used. The roles of the measurement process refer to the roles of the organization's processes, and the measure results and indicator analysis, interpretation and decisions are reported in the organization's process reports, etc.
- **Specific guidelines to adapt measurement definition to the measurement maturity of the company (GMM).** The MIS-PyME indicator template suggests to the measurement analyst the measurement maturity level that the company should attain in order to implement that indicator. The purpose, focus and entity are the indicator's fields that determine the measurement maturity level. The reference model by which the indicator

templates are classified by maturity levels is the MIS-PyME 3M. The measurement analyst may have to carry out an assessment in order to discover the company's measurement maturity level.

- **Indicator examples (EXMP)**. MIS-PyME Database: the indicator examples related to a measurement goal make it easier to define the measurement program and permit the user to improve and check its definition. These examples show how its corresponding indicator templates have been implemented in real companies.

Appendix G shows other characteristics of MIS-PyME with regard the other studied methodologies in bibliography.

In addition, the case study performed in this research has also derived a set of requirements for a measurement maturity model. MIS-PyME-3M was designed in order to fulfil these requirements which are not fulfilled by any other measurement maturity model, as was stated in Chapter 3. The reasons why MIS-PyME-3M fulfils these requirements are as follows:

- **3M-REQ1**: The measurement maturity model must explicitly indicate the criteria which states whether a process is at one level or another. In MIS-PyME 3M the indicators related to process performance and capability are clearly identified. These indicators are those labelled with "(P)" and determine to what extend the process performs the measurement process outcomes.
- **3M-REQ2:** The measurement maturity model must explicitly indicate the information needs that the measurement program is able to satisfy for each maturity level. In MIS-PyME-3M, the indicators labelled as "(I)" aim to evaluate the purpose of the measurement indicators. These indicators determine the capability as regards the measurement information that the measurement process is able to obtain in each measurement maturity level.
- **3M-REQ3**: The measurement maturity model must explicitly indicate the support tools and other resources of the measurement process that must be implemented for each maturity level. The indicators labelled as "(R)" determine the tools and infrastructure required to carry out the measurement process in each capability level.
- **3M-REQ4**: The measurement maturity model must explicitly provide a process with which to assess the measurement maturity level of the organization. MIS-PyME-3M provides an assessment process, which aims to assess the capability of the measurement process, which is explained in Section 4.3.2 and is shown in Appendix E.
- **3M-REQ5**: the measurement maturity model used to define measurement programs adapted to the measurement maturity of the company must be compliant with the international standard ISO/IEC 15504.
    a. 3M-REQ 5.1.The process reference model must be compliant with the requirement stated in the standard for the process reference model.

b. 3M-REQ 5.2. The measurement maturity model must be compliant with the measurement framework of ISO/IEC 15504.

c. 3M-REQ 5.3. The measurement maturity model must represent a mapping of the process reference model and the measurement framework of ISO/IEC 15504.

d. 3M-REQ 5.4. The process assessment of the measurement maturity model must be compliant with the process described in ISO/IEC 15504.

MIS-PyME is compliant with COMPETISOFT, and since COMPETISOFT is compliant with ISO/IEC 15504, MIS-PyME-3M is therefore also compliant with ISO/IEC 15504. Appendix F shows how the process reference model and the measurement assessment model (MIS-PyME-3M) comply with ISO/IEC 15504 by following the requirements stated in ISO/IEC 15504. In addition, the measurement assessment process also follows the assessment guidelines explained in ISO/IEC 15504, parts 2 and 3.

Therefore, MIS-PyME contributes to the bibliography by providing a methodological framework for defining and implementing a measurement programs which is suitable for SMEs. The methodology of this framework, in contrast with the popular methodologies found in bibliography, fulfils certain requirements which must be satisfied by any measurement program which aims to be suited to the special restrictions of SMEs. These requirements have been empirically identified. In addition MIS-PyME provides a measurement maturity model which, in contrast to the measurement maturity models found in bibliography, satisfies a set of requirements which have also been empirically identified. Consequently, and to the best of our knowledge, MIS-PyME-3M is the most complete measurement maturity model.

5. Action-Research Application: Cycles and Case Studies

The origin of the research in this thesis arose from the need to develop and implement a measurement program in the development division of STL. Suitable methodologies for defining and implementing measurement programs in this context were studied but, after concluding that none of these methodologies fulfilled the expected requirements, a methodology called MIS-PyME was defined and validated. This chapter shows how the research methods described in Chapter 2 were applied in the thesis, along with the results which were obtained.

5.1 Introduction

As is described in Chapter 2, the reference group and the company with which the research group collaborated to per sue the goal of this research is Sistemas Técnicos de Loterías del Estado (STL). This company was created by the Spanish government and provides the operations and IT development services for the national lottery. STL is formed of 246 people and its main business is to operate, develop and maintain the systems, networks and software for the Spanish lottery, the Spanish hippodrome and Services for Lotteries in Europe. STL manages 12275 terminals (IP, X-25 and satellite), and 2.096.300.000 transactions every year. The availability indicator is of 99.97% (234.750 minutes/year of the total of 234.814 minutes of service).

Software measurement initiatives have been encouraged by the quality control department of this company since 2003. However, there was no full agreement and acceptance with regard to the measurement program which was implemented at that time.

In June 2006, the director of the development and maintenance department requested the development of a measurement program which would support the achievement of the following process improvement goals:

- P.I.G 1: To improve project and process monitoring and control. The director was especially keen to improve the monitoring of the project's progress.
- P.I.G 2: To improve project planning.
- P.I.G 3: To improve integration testing effectiveness.
- P.I.G 4: To improve client satisfaction in terms of project conformance.

Consequently, there appeared both the need for a company (to define and implement the required measurement program) and a research goal: to understand whether GQ(I)M was a suitable methodology for defining measurement programs in SMEs and if that was not the case, to study whether there were other suitable methodologies. If there was no suitable methodological framework for defining and implementing measurement programs a new methodology would be defined and validated with this purpose. There was a special interest in GQ(I)M since this methodology is based on GQM, which is widely used in industry and clearly shows how to derive measurement goals from business or process improvement goals. It also proposes an indicator template which seemed very useful.

The chronology of the research is summarized in Table 5-1

Phase	Date	Description
Before the research	June, 2004	A measurement program was defined ad-hoc and implanted without success. The goals are as follows: - Evaluating the development process efficiency. - Evaluating the product in production - Evaluating the project deviation in terms of duration
Before the research	November, 2005	Some improvements were defined ad-hoc and without success.
A.R. Cycle 1	June-July, 2006	Case Study is performed. GQ(I)M is used to define the measurement program from scratch. However, it resulted in an unsuccessful measurement program. The measurement program definition was not accepted and a set of requirements for a methodology, aimed at defining and implementing measurement programs in the context of SMEs, was derived.
A.R. Cycle 2	August,2006	The second cycle of the research took place, and the popular methodologies in the bibliography for defining and implementing measurement programs were analysed in order to understand whether they complied with the requirements resulting from the previous cycle.
	September-December, 2006	MIS-PyME methodological Framework is defined
A.R. Cycle 3	December-January, 2007	The second Case Study was performed and MIS-PyME was used to define and implement a measurement program in the Development division of STL. The resulting measurement program was successfully implemented.
A.R. Cycle 4	February, 2007 - June, 2009	MIS-PyME was improved based on experience, particularly the Indicator templates and the measurement maturity model proposed by MIS-PyME. In addition, the development of MIS-PyME tool started (which finished in June, 2009)
	May, 2008	The Case Study 3 was performed. The measurement maturity of the Development Division of STL was assessed.
	April, 2009	The measurement maturity model of MIS-PyME and related templates were refined in order to better adapt them to an SME.

Table 5-1. Chronology of the Research

The people directly involved in the research were as follows:

- The author of this thesis, who was the measurement analyst in the department and whose usual work consisted of technical support, project coordination and testing. This person was partially assigned to the measurement activities. This person was also a member of the research group.

- The Top manager of the department, who was the sponsor of the measurement initiative.
- The reviewer who usually played the role of project manager.
- The supervisors of this thesis who were particularly involved in the reflection and planning of the research cycles.

5.2 Action-Research Cycles

The research was performed in four Action-Research Cycles, as is shown in Figure 5-1

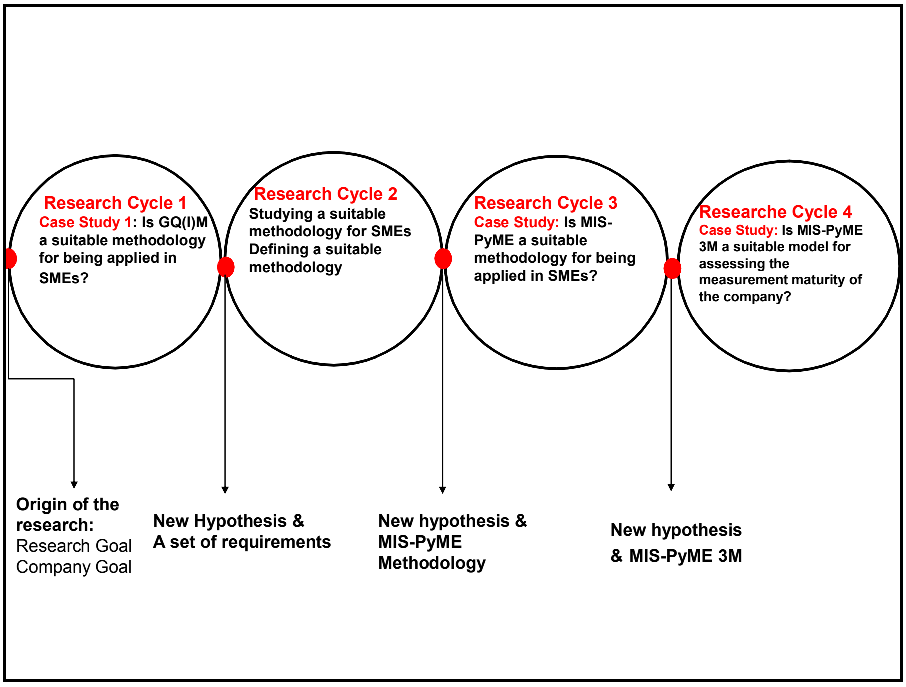


Figure 5-1. Action-Research Cycles

5.2.1 Action-Research Cycle 1

Cycle 1 aimed to define and implement the measurement program required by STL, using GQ(I)M. A Case Study was performed in order to understand whether GQ(I)M was a suitable methodology to define and implement measurement programs in SMEs.

The measurement program definition and implementation were unsuccessful. The measurement program defined by following GQ(I)M was not successfully carried out, mainly due to the fact that some indicators did not fit the measurement maturity of the company; another set of indicators could not be integrated into the remaining software processes; and the measurement program was not documented in an understandable fashion

Our opinion of this research cycle (after analysing the results of the case study) was that at least one methodology for defining and implementing measurement program would be more suitable if it provided guidelines and a lightweight methodology adapted to the restrictions of this kind of companies. These are: limited human resources, limited training and poor software measurement knowledge.

As regards the poor software measurement knowledge, a suitable methodology for defining measurement programs in SMEs should provide guidelines as follows: Specific guidelines to support basic process improvement needs (GPIN), specific guidelines to integrate measurement into the software processes (GINT), specific guidelines to adapt measurement definition to the measurement maturity of the company (GMM), specific guidelines to understand the benefits & potential for management and other guidelines (GGEN) and Measurement examples (EXMP).

Some studies also corroborate that SMEs demand more guidelines, examples, templates and checklists for applying methodologies and standards in addition to lightweight and easy-to-understand standards (Laporte et al., 2008).

As regards the need for lightweight methodologies whose roles and number of people that play each role are adapted to the size of these companies, the requirements stated are: few people involved in the process (FPP), reuse measurement models (RUSE), few but effective and complete Steps (FSTEP). A more detailed description of all these requirements was provided in Chapter 3.

5.2.2 Action-Research Cycle 2

The analysis performed in the previous cycle derived a hypothesis which is the input of this cycle: Do any methodological frameworks exist for defining and implementing measurement programs which fulfil the requirements stated?

The methodological frameworks GQM (Solingen and Berghout, 1999), GQ(I)M (Park et al., 1996), GQM Lightweight (Gresse et al., 2003) , ISO/IEC 15939 (2002), PSM (DoD, 2000) were analysed in order to determine whether these methodologies met these requirements (see Chapter 3).

The result was that none of the aforementioned methodological frameworks for defining and implementing measurement programs satisfied the requirements stated in this research. Therefore, based on these requirements, none of the methodologies are completely suitable for SMEs. Figure 5-2 shows a summary of the requirements and the degree to which they are covered by these methodologies.

REQ	GQM	GQ(I)M	GQM light weight	PSM	ISO/ IEC
FPP	No	No	Yes	Average	Average
RUSE	No	No	Yes	Yes	Yes
FSTEP	Average	No	Yes	Average	Yes
GPINT	No	No	No	Yes	No
GPIN	No	No	No	Average	No
GMM	No	No	No	Average	No
GB&P	No	Average	No	Yes	No
EXPL	Average	Average	No	Yes	No

Figure 5-2. Measurement Methodology Requirements Fulfilment.

Basically, GQM, GQ(I)M, GQM lightweight method and ISO/ IEC 15939 focus solely on the methodological area, i.e. on the steps required to define and implement a measurement program, but they do not deal with the supporting areas. In other words, they do not provide specific guidelines to support the definition of the measurement program. In addition, they are not focused on SMEs with the exception of the GQM lightweight method. The most complete framework is PSM (DoD, 2000), which widely deals with the supporting areas with the exception of the measurement maturity issues. However, its methodology must be adapted if it is to be applied in SMEs with regard to the roles and number of people involved. The support information is project-oriented and there is a lack of specific measurement support at all levels of the processes established in a software development and maintenance organization. Additionally, PSM does not provide any support as regards with measurement maturity. Therefore, since none of the popular methodologies in the bibliography completely fulfilled the stated requirements, MIS-PyME was defined (see Chapter 4).

Once MIS-PyME had been defined, we studied how it could integrate a measurement maturity model in order to define measurement programs adapted to the measurement maturity of the company. With this aim in mind, a set of assumption were stated in order to derive the set of requirements that were desired for the selected measurement maturity model. The assumptions were as follows:

1. The measurement maturity of the company is higher since the measurement process is better performed and established.

   3M-REQ1: The measurement maturity model must explicitly indicate the criteria which decides whether a process is in one level or another.

2. The measurement maturity of the organization is higher since more ambitious information needs are successfully obtained from the measurement program.

3M-REQ2: The measurement maturity model must explicitly indicate the information needs that the measurement program is able to satisfy for each maturity level.

3. The measurement maturity of the organization is higher since the support tools, related procedures and other resources are better established so the process will be performed more efficiently.

   3M-REQ3: The measurement maturity model must explicitly indicate the support tools and other resources of the measurement process that must be implemented for each maturity level.

4. The measurement analyst should be able to determine the measurement maturity of the organization.

   3M-REQ4: The measurement maturity model must explicitly provide a process with which to assess the measurement maturity level of the organization.

The popular measurement maturity models found in the bibliography were analysed in order to select a suitable model which would comply with all of the stated requirements. Daskalantonakis et al. (1990), M-CMM (Niessink and Van Vliet, 1998), the measurement maturity model of Comer and Chard (1993), and CMMI (2006) were studied (see Chapter 3). Figure 5-3 summarizes the requirements fulfilment for each model. The results were that the Daskalantonakis et al. model fulfilled all the requirements, and that CMMI practically fulfilled them. However, as regards CMMI, although it clearly defines the measurement process improvement and the measurement needs for each maturity level, these are spread out over all the software processes of the model and their identification is difficult. The Daskalantonakis et al. model, on the other hand, had to be improved in terms of measurement scope and support tools throughout the maturity levels.

In addition, an interface between the measurement maturity model and MIS-PyME had to be defined.

REQ	Daskalantonakis	Niessink & Van Vliet	Comer & Chard	CMMI
3M- REQ 1	Yes	Yes	Yes	Yes
3M-REQ 2	Yes	No	No	Average
3M-REQ3	Yes	Yes	No	Average
3M-REQ4	Yes	No	Yes	Yes

Figure 5-3. Measurement Maturity Model Requirements Fulfilment.

Therefore, an adaptation of the Daskalantonakis measurement maturity model was developed, including aspects of CMMI and Niessink and Van Vliet, into which MIS-PyME was integrated. This measurement maturity model will support the measurement analyst in defining measurement programs adapted to the measurement maturity of the company.

5.2.3 Action-Research Cycle 3

Cycle 3 sought a result. MIS-PyME was therefore tested in order to understand whether this new methodology was suitable for defining and implementing measurement programs in SMEs and defining measurement programs adapted to the measurement maturity of the company. Case Study research was applied in order to explore and validate the suitability of this methodology (see Section 5.3.2). The result was that MIS-PyME was found to be a good methodology for this purpose and this context. The MIS-PyME framework supported STL in:

- Defining better measurement goals from process improvement goals and developing those measurement goals more easily.
- Aligning the measurement program with the measurement maturity of the company.
- Integrating the measurement program into the measurement process.
- Defining the measurement program in a clearer and more reusable manner.

In addition, a set of improvements for the methodology were identified and developed:

- The MIS-PyME Measurement Goals Template had to be improved. More measurement improvement goals had to be identified.
- The MIS-PyME Measurement Maturity Model (MIS-PyME 3M) was redefined in order to make this model capable of assessing the measurement maturity of the company. The fifth requirement was included for the measurement maturity models: Since there is an international standard for software process assessment models (ISO/IEC 15504), the measurement maturity model had to be compliant with it, and therefore had to comply with the requirement stated in this standard:

  3M-REQ5: the measurement maturity model used for defining measurement programs adapted to the measurement maturity of the company must be compliant with the international standard ISO/IEC 15504.

  - 3M-REQ 5.1.The process reference model must be compliant with the requirement stated in the standard for the process reference model.
  - 3M-REQ 5.2. The measurement maturity model must be compliant with the measurement framework of ISO/IEC 15504.
  - 3M-REQ 5.3. The measurement maturity model must represent a mapping of the process reference model and the measurement framework of ISO/IEC 15504.
  - 3M-REQ 5.4. The process assessment of the measurement maturity model must be compliant with the process described in ISO/IEC 15504.

5.2.4 Action-Research Cycle 4

MIS-PyME 3M was used to assess the measurement maturity of the Development Division of STL. A case study was also performed in order to explore whether MIS-PyME 3M was a good methodology for determining the measurement capability level that the software development and maintenance department had, and to understand the measurement goals that the company should easily be able to implement (see Section 5.3.3). It also allowed the company to identify the measurement deficiencies and propose improvement actions.

In addition in this cycle, and after the case study, a review of MIS-PyME was carried out by the supervisors of this research and the last improvement to the MIS-PyME 3M was made. This improvement suggested limiting the maturity levels to three levels in order to better adapt this model to SMEs. As indicated in COMPETISOFT, it is not expected that SMEs will achieve more than three levels. The MIS-PyME 3M, MIS-PyME Measurement Goals Table and MIS-PyME Indicator Templates were therefore modified accordingly.

5.2.5 Action-Research Final Reflection

In short, we started by using a known methodology from the bibliography to define and implement a measurement program in the software development and maintenance department formed of 39 people in which there was a limited measurement knowledge, people were reluctant as regards software measurement initiatives, and the measurement capability of the organization as regards software measurement was quite low.

GQ(I)M was then applied to define a measurement program in this context. A case study (Case Study 1) was developed, but the measurement program was not successful. As a result, GQ(I)M was not considered to be a suitable methodology for SMEs. In Cycle 2 of the Research, a study was performed with the methodologies available in the bibliography in order to analyse whether any of these fulfilled the requirements stated. Since none of these completely fulfilled them, MIS-PyME was developed. The validation of MIS-PyME was performed in Cycle 3 and the result was that MIS-PyME was found to be a useful methodology and that it was more useful than GQ(I)M in the context of SMEs. As a consequence, the measurement program defined using MIS-PyME was implemented. The forth cycle aimed to evaluate whether MIS-PyME 3M was a good methodology for determining the measurement maturity of the company, the measurement improvements and for suggesting the measurement goals that the company could implement. These results were also successful.

Although the quality of the case studies was not good enough to completely validate the MIS-PyME model, we consider it to be a good start. The case studies should be replicated in other companies in order to match the results, and demonstrate the context in which MIS-PyME is useful and where it is not (perhaps in large and/or measurement experienced companies). .

However, the case studies may be useful for those SMEs which aim to define measurement programs, and for future research aimed at the continuous validation of MIS-PyME. The following section provides a more detailed description of the case studies.

5.3 Case Studies

This section shows the plan, design and results of the case studies performed in the cycles presented above.

5.3.1 Case Study 1

The frame for developing the case study is based on the template presented in (Brereton et al., 2008) and is as follows:

Backgroud:

- Identify previous research on the topic: A search in the bibliography was performed in order to seek experiences of using GQ(I)M, particularly in SMEs. Goethert and Hayes, (2001) provide the lessons learnt from three experiences of software measurement program implementation using GQ(I)M. However, none of these experiences particularly addressed the application of GQ(I)M in a small setting.
- Define the main questions being addressed by this study: Is GQ(I)M suitable for defining measurement programs in SMEs?
- Identify any additional research questions that will be addressed: Why is it suitable or not?

Design:

- Multiple or single case study: The case study was a single case study, which signifies that it was carried out in only one company and in one measurement program initiative. The reason for selecting a single case study is that the context of the case study was quite common in terms of measurement aspects in SMEs. Other studies corroborate this (Kasunic, 2006; Laporte et al., 2008; Luna, 2008):
  - Software measurement culture is not usual in this kind of companies. The software measures obtained are quite few and there is no formal process to obtain the full benefit from software measurement.
  - People in these companies are usually reluctant to carry out these initiatives or do not demonstrate any special interest.
  - The resources and budget are limited and it is not possible to contract experts to implement a software measurement process.

Therefore, it is our belief that the results of this case study will be replicated in other companies in the same context. However, one of the improvements proposed for this thesis was to replicate the case studies presented and enforce the theory and the results presented.

- Object of Study: There were two units of analysis. The first unit was the measurement program defined using GQ(I)M in the development and maintenance department of STL. This analysis unit was chosen because, if the participants were satisfied with the resulting measurement program, then the methodology for defining and implementing the measurement program would be a suitable methodology for SMEs. The other unit of analysis was the process defined in GQ(I)M for defining the measurement program.
- The sub-questions were are as follows:

  As regards the resulting measurement program:
  - Are the indicators and measures defined in the measurement program correctly derived from the business and measurement goals?
  - Is the resulting measurement program understandable?
  - Is the resulting measurement program easy to apply since people find it reliable?
  - Is the resulting measurement program easy to apply since it is well integrated into the measurement program?

  As regards GQ(I)M methodology:
  - Does the methodology clearly determine the activities, tasks and roles that should be performed in order to carry out the measurement program?
  - Are these activities and roles suitable for an SME?

The data collected were qualitative data and were based on the answers to the surveys provided by the participants: the measurement analyst, reviewer and the sponsor of the measurement program.

Case Selection: The measurement program that resulted from applying GQ(I)M in the development and maintenance department of STL was chosen because its context matched the usual context of SMEs as regards software measurement. The context in which GQ(I)M was applied is as follows (June, 2006):

- The resources in the department were limited and therefore the department could not spend too much effort on defining and implementing the measurement program. Measurement knowledge was quite limited throughout the company and consequently the person who defined the measurement program (measurement analyst) was a member of the company and was not a software measurement expert.
- Many people were reluctant to use the measurement initiative
- Measurement was not established in the company culture. Few software measures had been collected, there was no established measurement process and therefore the measurement maturity was quite low.
- The department was formed of 39 people.

Procedures and Roles:

- The roles were those identified previously (see Section 5.1): the measurement analyst, the sponsor and the reviewer. The case study is of a participatory nature since the measurement analyst was also one of the researchers.
- The procedure: The Goal-driven software measurement guidebook (Parker et al., 1996) was applied. The measurement analyst defined the measurement program based on the information obtained from the top manager. Once a draft of the measurement program had been defined, the reviewer checked the measurement program and proposed his suggestions to the measurement analyst. At the end of this process, the research questions were put to the reviewer, the measurement analyst herself and the Top manager.

Data Collection: The data collection is as indicated above, and is based on the answers to the research questions put to the participants, which are detailed as follows:

- Are the indicators and measures defined in the measurement program correctly derived from the business and measurement goals?
  - Do you agree with the measurement goals derived from the process improvement goals defined?
  - Do you agree with the indicators derived from the measurement goals?
  - Do you agree with the measures derived from the indicators?
- Is the resulting measurement program understandable?
  - Do you understand the indicators required for achieving each of the goals?
  - Do you understand the measures required for achieving each of the indicators?
- Is the resulting measurement program easy to apply since people find it reliable?
  - Are you confident with the data colleted? Are the tools required to collect the data suitable? Is the process through which the data are introduced into the tool coherent and reliable, etc.
  - Based on a simple example of an indicator, would you be confident to perform the analysis of the indicator based on the data shown in the indicator? Do you perfectly understand the data and are you able to perform an interpretation?
- Is the resulting measurement program easy to apply since it is well integrated into the measurement program?
  - Do you understand when each measure should be collected and when the indicators should be built during the actual management, development or maintenance process?
  - Are indicator and measure templates integrated into the products defined in the management, development or maintenance process of the company?
  - Is it clear which person should collect data, build the indicator, analyse and interpret the indicator, take actions and communicate results, in accordance with the management, development and maintenance process?
- Does the methodology clearly determine the activities, tasks and roles that should be performed in order to carry out the measurement program?
- Are these activities and roles suitable for an SME?

Plan Validity: This is an analysis of the design of the case study and was performed before this plan was applied. The aim was to verify to what extent the results would be true and not biased by the researcher's subjective point of view.

- Construct Validity: This aspect of validity reflects to what extent the operational measures that were studied would really represent what the researcher had in mind and what was investigated according to the research questions.

  Most of the questions put to the participants, which were the basis for data collection and analysis, were based on the assumption that if the methodology (GQ(I)M) was suitable, the resulting product would be good (measurement program).

  On the other hand, most of the questions were based on certain assumptions in order to indicate what a good measurement program is:

  - A good measurement program is one whose measurement goals are well derived from the process improvement goals and business goals.
  - A good measurement program is one whose measure results can be collected.
  - A good measurement program is one whose indicators are comprehensible and the person who analyses the measurement program is confident to answer the questions presented in the indicator in a reliable manner.
  - A good measurement program is one whose indicators and measures are well integrated into the company's other software processes.

  These assumptions were quite basic and acceptable for a good measurement program. Some of them matched the validation criteria for a measurement program presented in PSM part 7 (DoD, 2000). The other questions were based on how the methodology was adaptable to an SME in terms of activities and roles.

- External Validity: This aspect of validity was concerned with to what extent it would be possible to generalize the findings, and to what extent the findings would be of interest to other people outside the investigated case.

  The results of the analysis were that the conclusions of this case study could be generalized to other SMEs whose characteristics as regards measurement match those typically indicated in the design section.

  One threat to validity was that this was a single case study. The conclusions obtained from this study would, therefore, have been reinforced if the study had been applied in multiple cases in the same context and in which the results of each case matched. In addition, the case study should have been repeated in the rival context in order to demonstrate that if these context conditions had not been fulfilled, the results would have been different and therefore the required context conditions would have been demonstrated.

  However, it appeared to be an interesting case study since GQ(I)M was a well known measurement methodology and in most countries more than 70% of the software companies are SMEs . This may, therefore, be of interest to any SME that plans to define software measurement programs using any methodology available in the bibliography or that plans to use GQ(I)M.

- Reliability: This aspect was concerned with to what extent the data and the analysis were dependent on the specific researchers. This case study depended on the researcher to a relatively great extent since she was a participant in the case study. However, data collection was taken from two other sources (the top manager and the reviewer) in order to obtain a triangulation analysis. In addition, the researcher was part of the company and was trying to do a good job from the point of view of a measurement analyst and the conclusions derived were probably not affected by any special research motivation.

  As regards the case study replication, even if the procedures for defining the measurement program had been clearly defined in GQ(I)M and if the questions for collecting the analysed data had been quite simple, clear and understandable, the replication of the measurement program would not have been easy since it depended on the information from the top manager to define the measurement program, and on the measurement analyst's and the top manager's way of thinking, which is quite difficult to replicate. This was one of the reasons for using case studies, since there is no control between causes and effects. However, even if it is difficult to replicate the same case, the results of the exploration should at least be the same.

5.3.1.1 Case Study Analysis Results

The measurement program was defined by following GQ(I)M guidelines. After the measurement program was defined it was revised and the research questions were presented. The answers to the main questions in the study, which are based on the answers provided by each of the participants, are as follows:

- **Are the indicators and measures defined in the measurement program correctly derived from the business and measurement goals?**

  A problem appeared with regard to this. The indicator related to P.I.G 3: "to improve integration testing effectiveness", was not only focused on evaluating and improving the effectiveness of projects with the integration test phase but also on understanding the reliability of the product during the development phases in order to attain the reliability goal once the product was in production. The indicators aimed to collect the defects detected in each test phase, these defects were weighed up depending on their severity, and two focuses were consequently given to the indicator:

  - To understand the effectiveness of the integration phase: The integration reliability measure was compared with the acceptance reliability measure.
  - To evaluate whether the reliability of the product was sufficiently good at the test phases in order to attain a certain reliability goal in production.

  This indicator was neither clear nor well developed, since two goals were implemented rather than one.

- **Is the resulting measurement program understandable?**
  The resulting measurement program was not easily understandable for the top manager and the reviewer. The problem was the way in which it was documented. A template was defined for each business goal, sub-goal and indicator (the indicator includes the measure specification). Each of the documents, which included any goal, contained the specification of the goal (sub-goal), the related entities and questions. The document including the indicator information specified the indicator by following the template proposed in (Goethert and Siviy, 2004). Although the name of the document clearly determined the sub-goal or the indicator which had been specified, it was neither reader-friendly nor comprehensible.
  In addition, the measurement plan was defined in a document whose contents were basically: an introduction, the goals and scope, the implementation in terms of the activities required to implement the measurement program, the plan and schedule; the final section indicated the actions required to carry out the measurement program once it was implemented. However, we found that it would be more understandable and easier to reuse in the future if the plan for implementing the measurement program was defined in the project plan, and if the measurement program was defined in a different format.
- **Is the resulting measurement program easy to apply since people find it reliable?**
  As indicated above, P.I.G 3 was defined in order to evaluate the reliability of the developed product and during the test phases in order to attain a certain reliability goal in production. However, the measurement maturity was not sufficient to be able to evaluate the reliability progress of the product during the test phases in order to achieve a reliability goal in production. There was no historical database of previous projects with which to establish ranges and perform an evaluation.
  The indicator related to P.I.G-4 aimed to evaluate the reliability of the product based on a fixed goal which was settled with the client before the product was in production. Even if we had understood the reliability of the product exploited, our measurement maturity was not sufficient for us to be able to define what kind of fixed reliable goal was appropriate for the new products developed.
- **Is the resulting the measurement program easy to apply since it is well integrated into the measurement program?**
  The measurement program was not adaptable to the software development processes existing at that time. For example:
  - P.I.G-3 required a definition of the thresholds of the product's reliability during the test activities in the test plan definition activity, but the development, quality or management project processes did not contemplate this task and were not sufficiently mature to do so.
  - P.I.G-4 aimed to evaluate the reliability of the products based on a fixed goal which had been negotiated with the client before the product was in production. However, this task was not contemplated during the project management and development processes.

- **Does the methodology clearly determine the activities, tasks and roles that should be performed in order to carry out the measurement program? Are these activities and roles suitable for an SME?**
  GQ(I)M clearly determines the activities to derive measurement goals from process improvement goals or business goals. In addition, it clearly determines how to derive indicators from measurement goals and measures from indicators; but it does not clearly describe a process in order to understand who should perform each task and who the other participants are. GQM, the base methodology of GQ(I)M, does provide this information but it is not adapted to SMEs.

5.3.1.2 Case Study Conclusions and Validity

Based on the case study results, we can assert that GQ(I)M was not a good methodology for defining and implementing our measurement programs in the development and maintenance department of STL. However, we cannot definitively generalize this statement since this was a single case study, and although we believe that these kind of results would be replicated in another SME with the same conditions, we cannot rigorously state that GQ(I)M is not suitable for SMEs.

5.3.2 Case Study 2

Backgroud:

- Identify previous research on the topic: Some of the experiences regarding the definition and implantation of measurement programs have been addressed in Chapter 3. However, none of them was specific to SMEs and none of them used MIS-PyME methodology since it is a new methodology which is validated in this study.
- Define the main questions being addressed by this study: Is MIS-PyME a suitable methodology for defining and implanting measurement programs in SMEs? Is MIS-PyME more suitable than GQ(I)M for defining measurement programs in SMEs?
- Identify any additional research questions that will be addressed: Why?

Design: The design was the same as that of Case Study 1: a single case study, since it was a common case study.

- Object of study: There were two units of analysis, the measurement program which resulted from applying MIS-PyME and the process specified in the methodology for defining the measurement program.
- The sub-questions that were studied are those indicated in Case Study 1, but other questions were proposed in order to compare MIS-PyME and GQ(I)M:
  - Is the resulting measurement program more understandable than that defined using GQ(I)M?

  - Do you think that this measurement program is more reliable than that defined using GQ(I)M?
  - Do you find this measurement program better integrated into the processes of the organization than that defined using GQ(I)M?
  - Is the MIS-PyME process easier and better adapted to SMEs than the GQ(I)M process?
- The Data collection was qualitative data based on the answers to the surveys provided by the participants: the measurement analyst, reviewer and the sponsor of the measurement program.

Case Selection: The measurement context is the same as that indicated in the previous case study: the development and maintenance division of STL.

Procedures and Roles:

- The roles are those used in Case Study 1, thus: the measurement analyst, the sponsor and the reviewer. The case study is also of a participatory nature.
- The procedure: The manner of working consisted of following the MIS-PyME methodology (see Chapter 4).

Data Collection: Data collection was based on the answers to the research questions put to the participants.The detailed questions are the same as those defined for Case 1 but some have been added for the new questions which are as follows:

- Is the MIS-PyME process easier and better adapted to SMEs than the GQ(I)M process?
  - Does MIS-PyME define the required process for defining the measurement program in which roles and activities are clearly identified? Are these roles suitable for SMEs?
- Do you think that this measurement program is more reliable than that defined using GQ(I)M?
  - Does MIS-PyME 3M help to avoid the definition of indicators that require a more mature measurement process?
  - Does MIS-PyME 3M help to avoid the definition of measurement programs which require resources which are not available and/or are difficult to obtain?
  - Does the MIS-PyME 3M help to avoid the definition of measurement programs whose purposes are too ambitious as regards the measurement maturity of the company?

Plan Validity: This is an analysis of the plan of the case study and was performed before this plan was applied. The aim was to verify to what extent the results would be true and not biased by the researcher's subjective point of view.

- Construct Validity: This aspect of validity reflects to what extent the operational measures that were studied would really represent what the researcher had in mind and what was investigated according to the research questions.

  As has already been explained for the Case Study 1, most of the questions put to the participants, and which were the basis for data collection and analysis, were based on the assumption that if the methodology (MIS-PyME) was suitable, then the resulting product would be good (measurement program). The assumptions by which a measurement program was considered to be a good product were stated in Case Study 1.

- External Validity: This aspect of validity was concerned with to what extent it was possible to generalize the findings, and to what extent the findings were of interest to other people outside the investigated case.
  The result was that the conclusions of this case study could be generalized to other SMEs whose characteristics as regards measurement matched the typical ones which have been indicated in the design section of Case Stuy 1.
  One threat to validity was that this was a single case study. The conclusions obtained from this study would, therefore, have been reinforced if the study had been applied in multiple cases in the same context and in which the results of each case matched. In addition, it should have been performed in the rival context in order to demonstrate that if these context conditions had not been fulfilled, the results would have been different and the required context conditions would therefore have been demonstrated.
  However, it was an interesting case study since GQ(I)M was a well known measurement methodology and in most countries more than 70% of the software companies are SMEs. Therefore this may interest any SMEs that plan to define software measurement programs using any methodology available in the bibliography and/or show an interest in GQ(I)M or MI-PyME.
- Reliability: This aspect was concerned with to what extent the data and the analysis were dependent on the specific researchers.
  The main aspect with regard to the reliability validity of the study was that MIS-PyME was applied after GQ(I)M. Therefore the measurement analyst had already been trained in defining the measurement program and this would have led to a more competent definition of the measurement program when using MIS-PyME. In addition, questions related to the comparison between MIS-PyME and GQ(I)M were only answered by the measurement analyst.

5.3.2.1 Applying MIS-PyME

This section presents the full experience as regards the use of MIS-PyME for defining and implementing a measurement program.

The way in which the measurement program, the indicators and measures were defined, how they were instrumented and integrated into the other software processes, etc. and the benefits of the implemented measurement program are shown as follows:

Initiate the measurement program: In this task the Top manager of the development division of STL appointed the measurement analyst and stated the process improvement goals to develop for the measurement program:

- P.I.G 1: To improve project and process monitoring and control. The director was especially keen to improve the monitoring of the project’s progress.
- P.I.G 2: To improve project planning.

- P.I.G 3: To improve integration testing effectiveness.
- P.I.G 4 – To improve client satisfaction in terms of project conformance.

Based on these process improvement goals, MIS-PyME was applied to define the measurement program. The people involved until the verification of the measurement program were the Top manager and the measurement analyst only.

Formalizing the measurement goals and checking whether a measurement program is reused. The MIS-PyME Measurement Goals Table was used to determine the measurement goals that would solve the process improvement goals.

- For P.I.G 1 the measurement goals specified in the MIS-PyME measurement goals Table: Project Management Process -> Project Tracking was used to guide the Top manager with regard to the measurement goals that would satisfy the defined process improvement goal. Based on the Measurement Goals Table, the measurement goals were defined as follows:
  - IND-PRJ-Progress: Monitoring the conformity of the progress of the project as regards the schedule, work performed and cost and what was planned in order to take corrective actions or present problems throughout the project execution
  - IND-PRJ-ReqProg: Monitoring the state of the requirement in order to understand the project's progress

  In addition, it was desired to track the reliability of the product being developed, and therefore the "quality" element in the measurement goal table was selected and the measurement reliability goal was selected:
  - IND-PROD-OpenInc: Monitoring the incidents that still remain open in order to understand the remaining work to improve reliability.
- For P.I.G 2 the measurement goals were taken from "project management process -> project planning" and were as follows:
  - IND-PRJ-InexacSize: Characterizing the accuracy in terms of the estimated size of the product developed and actual size at completion in order to improve the size estimation in future projects
  - IND-PRJ-InexacDuration: Evaluating the accuracy in terms of the estimated duration of the project and actual duration at its conclusion in order to improve the duration estimation in future projects
  - IND-PRJ-InexacCost: Characterizing the accuracy in terms of the estimated cost of the project and actual cost at its conclusion in order to improve the cost estimation in future projects.
  - IND-PRJ-EffortDev: Characterizing the accuracy in terms of the estimated effort of the project and actual effort at its conclusion in order to improve the effort estimation in future projects.
- P.I.G 3: To improve integration testing effectiveness. The measurement goals were taken from the process effectiveness section of the Measurement goals Table but were renamed in order to refer to the integration test:

- INC-PROC-IntEffect: Evaluating integration test effectiveness in order to understand it, help to plan future projects and improve the process.

- P.I.G 4 – To improve client satisfaction in terms of project conformance. The Top manager defined this process improvement goal. However, there was no intention of interviewing the client with regard to that which was defined in the measurement goals table as “client satisfaction”, as the Top manager expected to understand those aspects which most affected client satisfaction. In this development division these aspects were:
  - Deviation from the duration of the project. This measurement goal was the same as that which had been previously defined for the planning improvement goal: IND-PRJ-InexacDuration: Evaluating the accuracy in terms of the estimated duration of the project and eventual actual duration in order to improve the duration estimation in future projects.
  - The product’s reliability once it was implanted in production. The product’s reliability indicator was selected for this measurement goal. However, its intention was not to measure the reliability of the product during development, but once it was implanted: IND-PROD-OpenIncImp - Evaluating the reliability of the product in order to understand whether it should be improved.

A new indicator was created (IND-PRJ-ClientConformance) which included the two aforementioned indicators.

In addition to the above indicators, other indicators were defined to carry out cross-project analysis:

- The planning related indicators were: IND-PRJORG-InexacEffort, IND-PRJORG-InexacSize, IND-PRJORG-InexacCost, Ind-PRJORG-InexacDuration
- The integration test effectiveness indicator was: IND-PRJORG-IntEffect
- The project conformance indicator was: IND-PRJORG-ClientConformance

Since the development division of the company did not have any measurement programs, there was no opportunity to reuse a measurement program.

Specifying project plan : The plan was quite informal. The measurement analyst was assigned to this project on a part-time basis. The measurement program was expected to take around a month for its definition and revision and half a month for the instrumentation, or at least the initial instrumentation.

Defining Indicators: In this activity the indicators required to develop the measurement goals were specified. The MIS-PyME Indicator Templates provided for each of the measurement goals an indicator template in the table which contained the standard information for its implementation.

Some of the templates were directly adopted and few aspects needed to be changed. This was the case of the indicators related to project planning. Other indicators required a greater amount of adaptation. In either case, the name of the reports and the name of the processes in which the indicator had to be integrated were translated to the methodology that was used in this department.

The indicators which required further tailoring were as follows:

- INC-PROC-IntEffect: Evaluating integration test effectiveness in order to understand it, help to plan future projects and improve the process. This template was quite complex since it belonged to the third maturity level and it required more modifications. It was stated that the effectiveness of the integration tests was acceptable if 70% of the total defects and 80% of the sever defects were discovered in the integration tests.
- IND-PROD-OpenIncImp: Evaluating the reliability of the product in order to understand whether the product should be improved. This indicator collected the incidents that had arisen in production, from the first implantation until one month after the last installation related to the project.

The maturity suggested by MIS-PyME was taken into account when the indicators were specified, with the exception of INC-PROC-IntEffect and IND-PROD-OpenIncImp which were two of the three indicators that had to be further modified. In these cases the indicator had been defined in a manner that made its successful implementation impossible since the measurement maturity of the development and maintenance division was not sufficient. Both indicators were initially defined in a manner that required a reliable prediction before the evaluation and STL was not prepared for this. The following section explains how the MIS-PyME 3M questionnaire helped the measurement analyst to realise that the company was not sufficiently mature to define this kind of indicator, and the indicator was redefined.

Define your measures and specify the measure results collection procedure: In this activity, each of the measures required to build the indicator were defined. In the same template it was indicated what data should be collected, what data should not be collected and how it should be collected. In addition, if any operation was necessary to obtain the measure result, this was also specified.

In this experience 25 measures were specified. In addition, there were other elements that had to be specified and which could be classified as criteria and concepts but not measures. As an example, the type of projects developed in the department was specified, since the ranges to evaluate certain indicators such as IND-PRJ-InexacDuration differed in the type of projects. In this experience the type of project was classified in two dimensions. One dimension was the type of project in terms of being external (requested from the client) or internal (not requested by the client). Another dimension was based on the effort that the project implied, and the projects were therefore classified as: low, medium or high based on defined ranges.

Another element was “estimation”, for example the duration estimation of the project. The manner in which this estimation should be performed, the methods used, etc. were also defined in order to unify the means used to estimate and measure. In this experience, 6 estimations and 9 concept criteria were defined.

Integrating the measurement program: Since there was no standard measurement process in the organization, three different sub-processes were identified:

- Project tracking measurement process: this measurement process was aligned with the control and follow up activity of the specific project administration process. The indicators related to this measurement process were those which develop the P.I.G-1 "to improve the project monitoring in contrast with the plan".
- Close of project measurement process: This measurement process was aligned with the closing activity of the specific project administration process. The indicators' data were collected and the indicators were analysed at the end of the project. These indicators were those named as IND-PRJ and related to P.I.G-2 "to understand and manage deviations from the plan to project closure", P.I.G-3 "to improve the efficiency of the integration test phases" and P.I.G-4 "to improve the project's conformance".
- Close of project portfolio measurement process: This process was aligned with the project management process. This measurement process analysed the results of the projects which had been closed in the last 6 months. It analysed those indicators denominated as IND-PRJORG, which were related to the process improvement goals as follows: P.I.G-2 "to understand and manage deviations from the plan to project closure", P.I.G-3 "to improve the efficiency of the integration test phases" and P.I.G-4 "to improve the project's conformance"

As has been indicated, the project tracking measurement, close project measurement and the close project portfolio measurement processes were aligned to the organization's other processes, particularly to the specific project administration process and to the project management process. This means that the specific project administration process and the project management process referred to these measurement processes in order to perform their tasks. In addition, the roles in each measurement process for "collecting data", "building the indicator", "analysing the indicator" were aligned to the roles specified in the specific project administration process and the project management process. In these cases the project manager performed most of the activities.

Finally the measurement results were included in the reports of the organizational process and in its templates. As an example, the indicator analysis results of the project tracking measurement process were included in the "follow-up" report specified in the specific project administration process. The analysis results of the closure of project measurement process were included in the closure of project report specified in the specific project administration process.

Measurement program verification: This activity involved only one project manager. The measurement program was first reviewed in two sessions. In the first session the measurement analyst gave an overview of the measurement program defined and suggestions were only received in the second session after the measurement program had been analysed. In the second session, some complaints regarding the source of data emerged. These complaints addressed the problem of the reliability of the data source, since the severity and the cause of failures in the incident database were not always well assigned by the internal users. It was decided to give some reminders to ensure that internal users followed the procedures correctly.

The reviewer agreed with the rest of the measurement program since he found it to be an enhancement of the measures previously obtained. In addition, the complaints regarding these previous measures were taken into account in this measurement program definition.

Instrumentation: The tools already in use in the department were used to collect all the data required. The measurement program attempted to avoid data that could not be obtained from tools already in use. Some of these tools needed to be tailored to automatically obtain the required data. These tools are the following:

- Microsoft Project Manager: The project management tool used in the company.
- ActiTime: A free tool which registers the effort dedicated to each task
- Remedy: An incident management tool.
- IRQA: A requirement management tool which was also tailored in order to trace the state of each requirement (coding phase, verification phase, acceptance phase, etc.).

Within the frame of tools with which to support the measurement process, we briefly studied certain software measurement tools such as *MetricFlame*, *MetricCenter*, *ProjectConsole*, etc. However, we preferred to use simple rather than complicated tools since complicated tools may make the understanding of the basic measurement process steps more difficult. As a consequence, after weighing up the difference between the efforts needed to study a complex tool, adapt it and train people, and the benefits provided, we chose to develop three simple Spread Excel Sheets and report templates to support each of the measurement sub-processes.

Acceptance: The acceptance was performed by the people who usually play the role of project managers in the organization. In this activity, the measurement program was verified in real scenarios. The closure of project measurement process and the closure of project portfolio measurement process were tested by means of analysing the related indicators of the projects which had been completed in the last six months. In general, both measurement processes were accepted but certain considerations arose:

- IND-PROD-OpenIncImp: this indicator only showed the number of incidents (failures) detected in one of the products that was implemented, but it did not provide information about their severity. This indicator was therefore modified in order to analyse and evaluate it based on the severity of the incident
- There were some errors in the closure of project report with regard to the measurement information, which were amended.

The project tracking measurement process was also tested in a pilot project, and the considerations that emerged were the following:

- It was still not possible to build the indicators IND-PRJ-ReqProg because people were not experienced in using IRQ since this tool was new. They had enough with its basic use and the process could not be complicated with the other attributes needed to build these indicators.

The remaining indicators of the process were accepted. However, since some of the indicators could not be used in relation to the project tracking measurement process, it was decided to repeat the test in 6 months in another pilot project. The project tracking measurement process was, however, optional and to be used at the project manager's discretion.

5.3.2.2 Case Study Analysis Results

- **Are the indicators and measures defined in the measurement program correctly derived from the business and measurement goals?**
  The top manager, the reviewer and the measurement analyst were satisfied with the measurement goals, indicators and measures defined. The indicators defined aimed to satisfy the measurement goals proposed and not others, as had occurred in the previous experience.
- **Is the resulting measurement program understandable?**
  The measurement program defined was easy to understand and reuse since it separated the project plan and the elements of which the measurement program was formed: measures, indicators and the measurement process.
- **Is the resulting measurement program easy to apply since people find it reliable?**
  The measurement program defined using MIS-PyME was reliable; however, certain issues were modified during the review of the measurement program. MIS-PyME 3M helped to perform this task. This is better explained in Case Study 3 (see below).
- **Is the measurement program easy to apply since it is well integrated into the measurement program?**
  The measurement program defined was successfully integrated into the remaining software processes without any great changes in the organizational processes. It was also integrated into the specific and management project processes. The indicator analyses were included in the templates of the follow-up reports and the closure of project reports.
- **Does MIS-PyME define the process required to define the measurement program in which roles and activities are clearly identified? Are these roles suitable for SMEs?**
  Yes, the processes proposed in MIS-PyME, including the roles that take part in each activity, were well adapted in our company and were easy to follow.

MIS-PyME vs. GQ(I)M The reasons which could have led to an unsuccessful measurement program definition when using GQ(I)M and to a successful measurement program when using MIS-PyME were described by the measurement analyst when answering the following questions:

- **Are the indicators and measures defined in the measurement program correctly derived from the business and measurement goals?** When using GQ(I)M it was difficult to specify the indicators which would fulfil the measurement goal and it was therefore easy to be mistaken when defining the indicators which support process improvement goals. Additionally, GQ(I)M pays a great deal of attention towards creating questions through which to derive sub-goals from business goals, which is time consuming for inexperienced measurement stakeholders and measurement analysts. This time could have been reduced by using some kind of guidelines such as those proposed by the MIS-PyME: Measurement goals table and the guidelines provided in the MIS-PyME Indicator Templates.

- **Is the resulting measurement program more understandable than that defined using GQ(I)M?** The measurement plan proposed in GQ(I)M does not separate the measurement program project plan from the definition itself, and it does not separate the measurement process from the indicator and measure elements, which makes its integration and reuse more difficult. MIS-PyME, however, defines the measurement program in a manner which makes its reuse simple. It separates the project plan (with which to define the measurement program), the elements of the measurement program: indicators, measures and other elements (e.g: value criteria, guidelines), and the measurement process, taking into account the fact that a measurement process can implement more than one measurement program or parts of different measurement programs.
- **Do you find this measurement program better integrated into the organization's processes than that defined using GQ(I)M?** GQ(I)M does not pay a great deal of attention towards how the measurement program can be integrated into the rest of the software management, development and quality processes or whether this integration is possible according to the maturity of these processes. In the first experience, we failed to define two indicators related to this issue. MIS-PyME, however, provides guidelines in its indicator templates with which to integrate these indicators into the other software processes and explicitly makes the user think about this important issue since a task is dedicated to this.
- **Is the process provided in MIS-PyME easier to use and better adapted to SMEs than the process provided in GQ(I)M?** GQ(I)M does not explicitly determine the roles that should take part in each activity nor the people which should be involved in a small sized company. MIS-PyME provides this information. Another drawback of GQ(I)M is that it does not deal with the implantation of the measurement program, and activities such as instrumentation verification and acceptance are not therefore taken into account.
- **Do you think that this measurement program is more reliable than that defined using GQ(I)M?**

  The measurement program defined using MIS-PyME was reliable and the participants agreed to implement it, in contrast to what had happened in the previous experience. However the measurement program was not reliable from the beginning.

  One of the indicator goals aimed to evaluate the reliability of the product developed. This indicator was intended to evaluate the reliability of the company based on a fixed value meant to be a goal (the number of failures registered in production after the product had been installed). The intention of this indicator was to "evaluate" based on a predicted reliability goal, and this was at level 4 according to the MIS-PyME measurement maturity interface of the methodology (see Chapter 4). The focus was "reliability" and the entity was the “product”, which were at lower levels. However, even if the company had had experience and had (more or less) known the reliability of the products in production thanks to the MIS-PyME measurement capability maturity, the measurement analyst realized that the company’s experience was not sufficiently mature for determining, with a fixed goal, what the reliability of the product would be based on the characteristics of the product and project being developed. The MIS-PyME 3M questionnaire caused the analyst to reflect on this, since she was not able to answer the following questions: “do you measure the reliability of the product, and other aspects that may have a relationship with the reliability of the product, in a rigorous, frequent and organized fashion?”, “could you set reliable goals based on the available data?”, “Are there reliable and defined methods with which to control dirty data in the historical measurement database of the organization?”, etc. The answers to these questions were negative. She therefore decided to evaluate indicators based on a range of values (good, normal, not too good, not acceptable). In this case, although the maturity level required was still high (level 3) as regards reliability in production, she felt able to implement this indicator successfully and was therefore able to answer the questions in an affirmative manner: it was possible to accurately define the ranges of reliability of the product developed, which would depend on the type of project: high, medium, low.

The indicator INC-PROC-IntEffect was also modified to better adapt it to the maturity of the company. This indicator was defined so as to achieve the third process improvement goal: Improving the effectiveness of the integration tests. Initially, this indicator assessed the effectiveness of the test phases based on the failures detected during each test phase and compared with a threshold. However, the company was not sufficiently mature to define a threshold for each testing activity and, depending on the product and project developed, the company failed to answer level 4 questions. However, they were sufficiently mature to define a normal percentage ratio between test phases (e.g. more than 70% and 80%of the sever failures should be detected during integration test). The first indicator purpose was at level 4 and that finally defined was at level 3.

These two examples positively answer the research questions defined as regards the usefulness of MIS-PyME 3M. However, one indicator was defined which could not be implemented: IND-PRJ-ReqProg. MIS-PyME suggests this indicator at Level 3, quite high at this time for the company. However, since a requirement management tool had been applied in the company, the measurement analyst considered that it was possible to apply it. The result was that this was a mistake which was detected in the acceptance activity.

5.3.2.3 Case Study Conclusion and Validity

The problems detected when using GQ(I)M were not caused solely by the fact that GQ(I)M was used. If a measurement expert had defined the measurement program, s/he might have succeeded, but when bearing the aforementioned issues in mind, it became evident that the program could quite easily fail if GQ(I)M was used. Since SMEs do not have many resources with which to define measurement programs at their disposal, and since the people assigned may be from within the company, it would appear that MIS-PyME is more suitable for SMEs than GQ(I)M since the methodology is complete and is focused on SMEs in terms of roles and activities, and it also provides support guidelines which allow the (probably) inexpert user to define the measurement program faster and in a more reliable manner.

As has already been commented on in the validation plan, in spite of the fact that both methodologies were strictly followed, the MIS-PyME experience may have been affected by previous experience since the participants had received more training in defining the measurement program. However, MIS-PyME had already provided MIS-PyME workproducts which would have influenced the measurement analyst and the top manager. The measurement analyst was guided by the information provided by the Top manager and the workproduct, and not by the measurement program defined by GQ(I)M. If GQ(I)M had been applied after MIS-PyME, the bias would have been higher since GQ(I)M does not provide work support products on which to base the measurement program.

The measurement analyst stated that GQ(I)M lacked certain aspects of methodology (roles, verification activities, reuse of measurement programs) and that, thanks to the MIS-PyME support modules, it was more difficult to be mistaken than when using GQ(I)M.

However, this case study cannot be used as a formal validation of MIS-PyME since, as was indicated in the validation plan, this was a single case study. It would be advisable to reinforce the results by performing other case studies in other companies in the same context in order to demonstrate the usefulness of MIS-PyME. In addition, the measurement analyst was one of the researchers and bias could therefore have occurred when assessing the usefulness of the measurement program. However, two other people were also questioned (the Top manager and the reviwer) and they agreed with the conclusions.

Therefore this case study served to preliminary validate the MIS-PyME methodology and encouraging results were obtained.

5.3.3 Case Study 3

The last Case Study performed aimed to explore whether MIS-PyME 3M was suitable for assessing the measurement capability of the company, identifying measurement improvements and understanding the measurement goals that the company was able to implement.

The person who performed the assessment was the measurement analyst who had defined the previous measurement program. The assessment was performed a year after the measurement program was implanted.

Background:

- Identify previous research on the topic: Chapter 3 shows the measurement maturity models that exist in the bibliography and presents an analysis of them with regard to certain requirements identified in this research.

- Define the main questions being addressed by this study: Is MIS-PyME 3M suitable for determining the measurement capability of the company?
- Identify any additional research questions that will be addressed: Why?

Design: The design was the same as that of Case Studies 1 and 2: a single case study since it was a common case study. The unit of analysis was MIS-PyME 3M.

- The sub-questions that were studied are as follows:
  - Is MIS-PyME 3M suitable for defining the measurement capability of the company?
  - Is MIS-PyME 3M suitable for identifying measurement improvements?
  - Is MIS-PyME 3M suitable for understanding the measurement goals that the company is able to implement?
- Data collection was qualitative data based on the answers to the surveys provided by the participant: the measurement analyst.

Case Selection: The assessment to be performed using MIS-PyME 3M in the development and maintenance department of STL is the case selected.

Procedures and Roles:

- In this case study there was just one role, the organization's measurement analyst. The measurement analyst was also the same person who belonged to the research group.
- The procedure: The means of working consisted of following the MIS-PyME assessment process (see Annex E). The measurement analyst used the information to perform the assessment as follows: List of projects performed in the organization during that period of time, their project and quality plan (and its various versions) the tracking of project reports, the closure of project reports, the measurement reports performed every six-months as defined by the measurement processes, and the measurement processes defined in the company.

  The measurement analyst had quite a good knowledge of the measurement activities performed in the software and development department, she asked the questions proposed in the questionnaire with regard to the measurement assessment process, and she indicated the reasons for her answers and addressed the documentation in which the evidence could be found.

Data Collection: Data collection was based on the answers to the research questions provided by the measurement analyst. These answers were qualitative data but were based on the results of the assessment, which were quantitative data.

- Is MIS-PyME 3M suitable for defining the company's measurement capability?
  - Does it provide a set of measurement related attributes and their value?
  - Does it provide a capability level based on these attributes?
  - Was the assessor able to reliably determine the measurement maturity of the company?
- Is MIS-PyME 3M suitable for identifying measurement improvements?
  - Was the assessor able to identify measurement related improvement?
- Is MIS-PyME 3M suitable for understanding the measurement goals that the company is able to implement?

- Was the assessor able to determine the measurement goals that the company was able to implement?

Plan Validity: This was a previous analysis of the case study's plan through which to verify to what extent the results were valid and not biased by the researcher's subjective point of view.

- Construct Validity: This aspect of validity reflects to what extent the operational measures that would be studied really represented what the researcher had in mind and what was investigated according to the research questions.

  The case study results were based on qualitative data which were the answers to the research questions. The research questions directly addressed the assessment process and model, and therefore the results directly addressed what was being investigated.
- External Validity: This aspect of validity is concerned with to what extent it was possible to generalize the findings, and to what extent the findings were of interest to other people outside the investigated case.

  The results of this case study could not be generalized. More case studies would have to be performed in SMEs in order to definitively state that MIS-PyME 3M was a suitable model. In addition, the results of this case study were only based on one person, the measurement analyst. It is necessary to involve more people in the case study in order to collect different opinions and contrast the results.

  This single study would, however, be useful if the research was continuous and more cases studies were performed ("to discover the suitability of MIS-PyME 3M"). In addition, theoretically, MIS-PyME 3M was quite a complete measurement maturity model since it fulfilled certain requirements stated in this research (see Section 5.3.3), which none of the measurement maturity models found in the bibliography were able to do. Therefore, we consider this model to be of interest to the measurement area. In addition, other researchers or companies may exist which are interested in applying and studying MIS-PyME 3M.

  Although this case study was not robust, the results of using MIS-PyME 3M in the development and maintenance department of STL were consistent and presented.
- Reliability: This aspect is concerned with to what extent the data and the analysis were dependent on the specific researchers.

  This was the main threat to validity, since the researcher was the same person who performed the assessment and answered the research questions. This case study should be repeated using independent participants.

5.3.3.1 Case Study Analysis Results

- **Is MIS-PyME 3M suitable for assessing the measurement capability of the company?**

The structure of the measurement assessment attributes was consistent and the assessment process was easy to follow. Since MIS-PyME 3M is based on the COMPETISOFT assessment model and complies with ISO/IEC 15504, it also appears to be reliable. MIS-PyME was considered suitable since, from the point of view of the measurement analyst, it correctly supported the department's measurement assessment. The measurement maturity level was Level 2. The measurement maturity of the development division was characterized by a set of attributes and these were rated. The results of the assessment were as follows:

- Level 1 maturity attributes were fully achieved, since a basic measurement process for tracking the reliability of the products in production had existed for quite some time. In addition, basic project tracking was performed in all projects. Project managers had to include the schedule deviation, its causes and the actions to be performed in their project tracking reports.
- Level 2 was largely achieved: the measurement program was clearly specified, the closure of project reports asked project managers to sum up the results of the development of the project, and other measures such as cost, and effort, duration deviation, reliability etc. therefore had to be indicated. Other indicators were also analysed during the project in order to track reliability, the project's progress, etc.

  However, one important aspect in relation to the PA 2.1 (P) Performance management attribute and the PA 2.2 (P) Workproduct management attribute must be improved: responsibilities are clear with regard to the collection of data, analysis of results and feedback, but in some cases it is not clear who the person responsible for, and the main parties interested in the measurement process are, or what their responsibilities are. In the measurement process for project tracking, measurement analysis results are used for decision making purposes and the project manager is aware of this. But this is not the case in other measurement results concerning, for example, product reliability tracking in production. In these measurement processes, indicators are analysed and improvement issues are identified, but it is not clear who is responsible for carrying out these improvement initiatives, so it is not therefore clear who the main person interested in and responsible for the measurement process is. As a consequence, measurement results in these cases are not usually used to improve development projects and the maintenance process.
- With regard to Level 3, one attribute was largely achieved: the PA 3.1(P) Process definition attribute. The PA 3.6 (R) Resources deployment attribute was partially achieved. However, the remaining Level 3 attributes were not achieved. Some of the issues identified were the following: there were standard measurement processes that defined indicators for cross-product and project analyses, there were other more advanced indicators for monitoring tracking: requirement status, etc. However, the standard process for project monitoring purposes was not rigorously applied as specified in real-life projects. With regard to cross analyses, the analyses were not performed in such profundity as they should have been, responsibilities for analyses and feedback were not clear and results were not used to carry out improvement actions.

  Table 5-2 shows the measurement attributes profile of the development department.

Attribute	Assessment Results
PA 1.1 (P) Process performance attribute	Fully achieved
PA 1.2 (I) Basic project and product focus performance attribute	Fully achieved
PA 1.3 (R): Basic management tools implemented attribute	Fully achieved
PA 2.1 (P) Performance management attribute.	Largely achieved
PA 2.2 (P) Workproduct management attribute	Largely achieved
PA 2.3 (I) project and product focus management attribute	Largely achieved
PA 2.4 (R): management and development tools implemented attribute	Fully achieved
PA 3.1(P) Process definition attribute	Largely achieved
PA 3.2 (P) Process deployment attribute	Not achieved
PA 3.3 (I) Standard and advanced measurement	Not achieved
PA 3.4 (I) Advanced product tracking attribute	Partially achieved
PA .3.5 (R) Resources deployment attribute	Largely achieved

Table 5-2. Development Division of STL measurement assessment profile

- **Is MIS-PyME 3M suitable for identifying measurement improvements?**

  MIS-PyME 3M was suitable for identifying the improvement actions in STL:

  - To strengthen the knowledge of the measurement processes implemented in the company in order to attain a better understanding of the responsibilities, the goals and the measurement process itself. The intention of this is to encourage the person responsible for the measurement program to use measurement results for her/his purposes and to apply the standard process correctly. In order to achieve this improvement goal, a training program will be defined and given to the headquarters of the company, the quality department and to the project managers.
  - To formalize and supervise the improvement initiative programs in the company. In order to achieve this goal, a presentation will be given at the company's headquarters in order to highlight the importance of the improvement programs as projects and also to encourage the project management office to supervise and support improvement programs.

  The measurement analyst stated that even if a company understands its measurement deficiencies, a measurement capability maturity model is important since it objectively identifies these problems and encourages their resolution and the carrying out of process improvement initiatives. In addition, improvement initiatives are suggested in a feasible and progressive manner, rather than promoting just any improvement initiative, in no particular order and without taking into account whether or not the company is prepared to successfully carry it out.

- **Is MIS-PyME 3M suitable for understanding the measurement goals that the company is able to implement?**

Based on the MIS-PyME 3M interface, a further conclusion of the assessment is that the software development and maintenance department is at level 2 of the measurement capability. The company should not therefore have any problems in defining level 1 and 2 measurement goals, but they must take care when they wish to implement a level 3 measurement goal. Level 4 and 5 measurement goals are not recommended.

It was therefore stated that MIS-PyME 3M was a good model for identifying the measurement goals that the company is able to implement.

5.3.3.2 Case Study Final Results and Validity

Although the design of the case study was not sufficiently robust, it was demonstrated that, in the case studied, MIS-PyME 3M was useful for determining the measurement maturity level of the development and maintenance department of STL. A set of attributes are presented and evaluated, measurement improvement issues have been identified and the limitations as regards the measurement goals that the company is able to implement are also identified.

However, more case studies must be performed in order to reinforce the results obtained in this research.

5.4 Lessons Learned

The contents included in this section are the benefits of the measurement program being defined in STL, the improvements made, the lessons learned and the effort required to implement the measurement program.

5.4.1 Measurement Program Benefits

The benefits of the measurement program identified by the top manager are as follows:

- Improving the plan for future projects in terms of duration, effort and cost.
- Understanding the reliability of the products being developed and implemented in production and attempting to improve them.

- Improving the client satisfaction in terms of the duration of the projects and the reliability of the products.
- Improving the effectiveness of the integration tests.
- Using measurement information to track projects and to provide objective information concerning progress, problems, etc.

In addition, there were a set of indicators, measures and measurement processes that were well defined and unified in the software development and maintenance division of the company which permitted cross project analysis, tracking and control of the measurement process, and integration into the organization's other processes.

To illustrate these benefits, an example of the results of the IND-PRJORG-IntEffective indicator is shown in Table 5-3.

Analysis Date	Prj	Relation Total	Relation Grv	Conform_Total	Conform_Grv
31/12/2006	PR1	86,36/13,64%	83,33/16,67%	YES	YES
31/12/2006	PR2	50,00/50,00%	50,00/50,00%	NO	NO
31/12/2006	PR3	0,00/100,00%	0,00/100,00%	YES	YES
31/12/2006	PR4	47,37/52,63%	47,06/52,94%	NO	NO
31/12/2006	PR5	0,00/0,00%	0,00/0,00%	YES	YES
01/07/2007	PR6	72,73/27,27%	71,43/28,57%	YES	YES
01/07/2007	PR7	50,00/50,00%	60,00/40,00%	NO	NO
01/07/2007	PR8	60,66/39,34%	65,38/34,62%	NO	NO

Table 5-3. Example of IND-PROCORG-TESTCONFORMANCE results.

The analysis of this indicator is derived from the data collected during the acceptance phase of this measurement program and those collected six months later, once the measurement program had been successfully implemented. This indicator was analysed in eight projects. The results show that only half of these projects achieved their goals: the total relation of failure between the integration and acceptance test phases should be 70% / 30%, and 80% / 20% for severe failures.

5.4.2 Improvement

Although the measurement program was accepted and was being used, it was quite difficult to understand the measurement program well. It was difficult to access the indicator and measure specifications, and the measurement process. The measurement program was therefore specified by using Software Process Engineering Metamodel (SPEM). The SPEM representation of the measurement program facilitated its management. For instance, this measurement program could be displayed by means of an automatically generated WEB site (see Figure 5-4).

Note that in Chapter 4 we indicated that the MIS-PyME Framework was also defined in SPEM, but in this case it is not MIS-PyME but the result of applying MIS-PyME, and thus the measurement program that was specified in STL. Nevertheless, the indicators and the measurement process templates provided by the MIS-PyME Framework are easily reusable since they are specified in SPEM. The user can reuse these elements and ease the definition of the measurement program in SPEM.

The advantages of defining the measurement program in SPEM are as follows:

- Access to the related measurement information is easier and faster.
- Measurement training is improved if there is an accessible and friendly WEB which includes the measurement program specification.
- It is easier to carry out the measurement process, and this can be done in a more reliable manner.
- Activities, tasks, roles, templates, guidelines are accessible and centralized.
- Reuse is enhanced. It is easier to reuse the indicator, measure elements or even part or the whole measurement process. SPEM provides a mechanism with which to easily reuse each element of the measurement specification.
- It is easier to manage and improve the measurement process

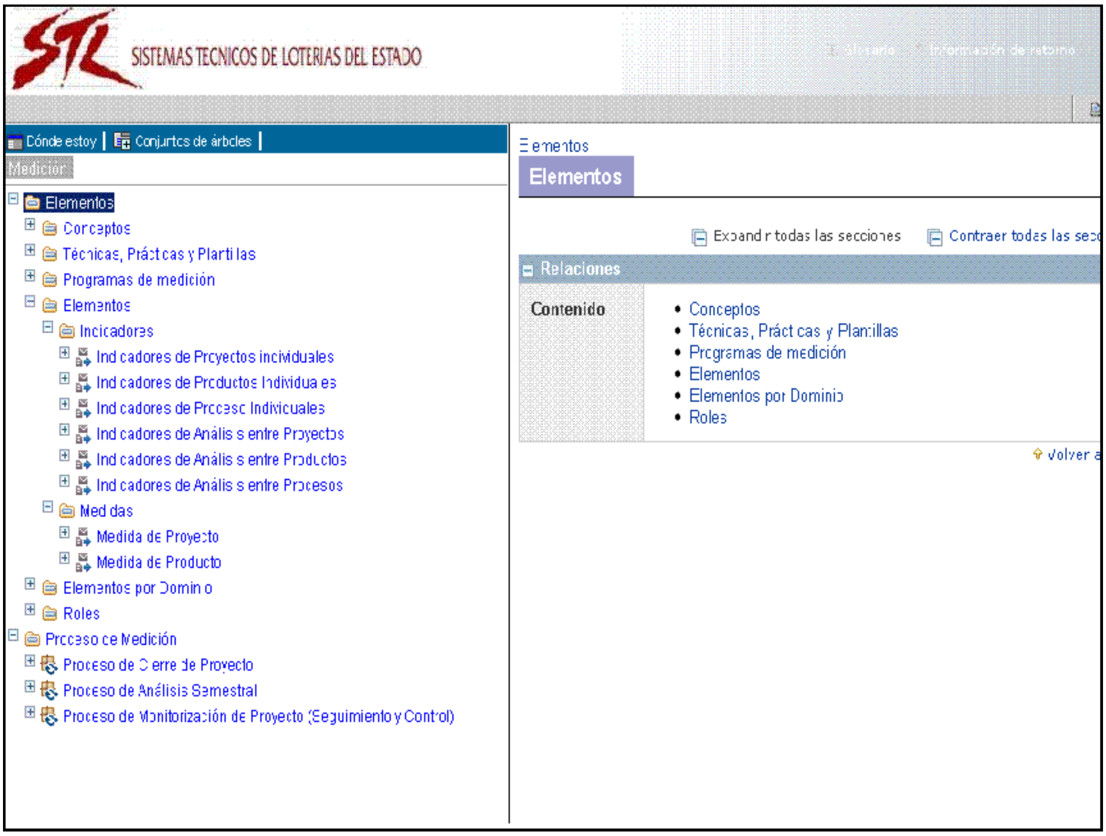


Figure 5-4. Measurement program of STL in SPEM

5.4.3 Success Factors for Measurement Programs in SMEs

This experience led to certain success factors that should be taken into consideration when the measurement program is intended to be implemented in SMEs with limited resources, measurement knowledge and maturity.

Some authors have already identified the practices suggested below as being success factors in the implementation of measurement programs (see Chapter 3). In accordance with these practices, we propose others which make the definition and implementation of measurement programs easier with regard to SMEs' restrictions.

- **Reusing measurement models and defining measurement programs for the organization's use.** The definition of the measurement program should focus on defining the measurement program in a reusable manner, and an effort should be made to reuse the existent measurement programs already defined and implemented.
- **Measurement for supporting process improvement goals not business goals.** Attempt to define measurement programs which focus on supporting software process improvement goals rather than business goals. Low measurement maturity level settings cannot afford measurement programs from just any business goal. If the aim is to define successful measurement programs in an effortless, accurate and consistent manner, these measurement initiatives should be aligned to process improvement goals to which the organization is already committed.
- **Measurement programs adapted to the measurement maturity of the company.** Measurement Program definition should be adapted to the measurement maturity of the company. The definition should be of what is possible and reliable, even if it is not the best.
- **Define a measurement program definition draft involving few people.** Define a draft of the measurement program involving few people: the top manager (the main stakeholder) and the measurement analyst. This means of working reduces the time spent on defining measurement programs, and the reviewers make more suitable suggestions and are more motivated if a draft of the measurement program has already been defined. Since the aim is to seek common useful measurement goals that support software process improvement, the top manager is able to support the measurement analyst in defining the first approach of the measurement program.
- **Use of Excel Spread Sheets or familiar databases.** Use these kinds of tools first. Before having powerful tools it is better to understand the process and to control the essential activities. Furthermore, the benefits that the tool provides may not make up for the cost of evaluating the tool and training people. Once the company is sufficiently mature, other more powerful tools can be acquired.
- **Use of data which is easy to collect.** Try to define measures which are already available or easily available to the company. Take advantage of the data provided by already existing development tools and attempt not to collect ambiguous or difficult (as regards data collection) data.

- **Make the person responsible for maintaining and improving the measurement program aware of any changes in the organization's processes.** This person must be at least concerned with the new tools that are to be acquired and could substitute old ones. S/he should verify the changes made in configuration, development and management tools since any change in these tools may influence or disturb the measurement activity. S/he should, moreover, be aware of any changes in the quality, management or development management processes.
- **Try to define the measurement program using SPEM.** As indicated above, specifying the measurement program in SPEM eases access to the measurement information, eases the reuse of the measurement program, improves measurement training, centralizes the measurement information, improves measurement process management and improvement, etc.

5.4.4 Effort Required

The definition and implementation of the measurement program took two months and 201 hours of effort (see Table 5-4).

Phase	Effort
Definition of the measurement program and integration	156 hours approx. (6 hours for the top manager and 150 for the measurement analyst)
Verification and acceptance	20 hours (the trial test with the pilot project is not included - 16 hours for project managers and 4 hours for measurement analyst)
Instrumentation	25 hours approx. (for measurement analyst)

Table 5-4. Measurement program definition and implementation effort (Phase 2)

This effort is not apparently high, if we take into account the fact that the measurement program supported four process improvement goals, which signifies that it took around 50,25h for each process improvement goal and that this was the first time MIS-PyME had been used. It is expected that the definition and implementation time will be reduced.

6. Conclusions

This chapter shows the final results of this research, and their evaluation. It starts with an analysis of the attainment of the research goals and the main results (corroborated by the publications shown), and future research lines are then stated.

6.1. Research Goals Analysis

Industry recognizes software measurement as an accepted requirement for the effective management of software projects and organizations. However, the successful implementation of measurement programs is still a difficult task, and more so for small and medium companies since some of the common characteristics of these companies become obstacles for the implementation of software measurement programs: limited resources, limited software measurement knowledge, etc.

In this thesis, a methodological framework for defining and implementing measurement programs adapted to the restrictions and maturity of small and medium companies is proposed.

In the first Chapter of this thesis, the research goal was presented:

To define a methodological framework in order to support small and medium-sized enterprises in establishing measurement programs. This methodology should be a lightweight methodology which takes into account the maturity and the limitations of the company.

In addition, a set of sub-goals were also stated which had to be solved in order to achieve the goal of the research. A summary of the achievement of each of these sub-goals is explained hereafter:

Goal 1: To carry out an in-depth study in the Measurement Program Area. This is tantamount to studying the known standards and methodologies that support the definition of measurement programs.

Chapter 3: State of the Art, describes the main methodologies for defining software measurement programs and presents an analysis of them based on the phases proposed by SWEBOK(2008) for software measurement. In addition, other related existing approaches which complement the main methodological frameworks are presented and analysed.

Goal 2: To study literature describing experiences in the implementation of measurement programs in companies so as to know how the measurement program was implemented and what the results were.

Chapter 3: State of the Art, presents some experiences in defining and implementing measurement programs, with a particular emphasis on the goals, the results and the methodology used. However, there are not many experiences relating to the development and implementation of measurement programs in SMEs.

Goal 3: To study the suitability of the above standards and methodologies to support SMEs in defining measurement programs.

Chapter 5 shows a case study in which GQ(I)M, one of the popular methodologies analysed, was studied in order to understand its suitability for SMEs. However, the result was that GQ(I)M was apparently not suitable for SMEs. Consequently, a set of requirement were stated for a measurement methodology which would be suitable for SMEs. The methodological frameworks popular in bibliography, were then analysed based on these requirements in order to theoretically understand whether these other methodologies were suitable for SMEs (see Chapter 3). The results were that none of these methodologies completely fulfilled these requirements. The motivation and the basis of the research in this thesis were therefore justified.

Goal 4: To thoroughly develop a methodological framework with which to support SMEs in defining software indicators according to their maturity and limitations.

Chapter 4 thoroughly describes the methodological framework developed in this thesis, denominated as MIS-PyME. It describes the principles from which MIS-PyME was designed, and shows the activities, tasks and roles of the process for defining and implementing a measurement program which is consistent with the size of a particular company. This methodology is based on GQM (Solingen, 2002) and GQ(I)M (Park et al., 1996). In addition, it defines the supporting modules with which to ease the definition of the measurement programs in SMEs: the MIS-PyME measurement goals Table, the MIS-PyME Indicator Templates, and MIS-PyME Indicator Database. A measurement maturity model (MIS-PyME 3M) was developed and integrated into MIS-PyME Framework, and is used to define measurement programs according to the measurement maturity of the company, assess the measurement maturity level of the company and identify feasible improvement measurement suggestions.

A tool has, moreover, been developed in order to ease the use of MIS-PyME Framework.

Goal 5: To validate the methodological framework in a small or medium company. Two case studies through which to achieve this goal have been performed. The aim of the first case study was to validate MIS-PyME in its definition and implementation of measurement programs adapted to the restrictions and maturity of SMEs. This case study provides a step-by-step description of how MIS-PyME was applied in the software development and maintenance division of STL. It shows why MIS-PyME was a suitable methodology, why it was considered to be more suitable than GQ(I)M and why MIS-PyME 3M was useful for defining a measurement program adapted to the measurement maturity of the company..

The benefits of the resulting measurement program are presented, and the lessons learned from the experience are also addressed.

The second case study aimed to validate MIS-PyME 3M as regards this model's capacity to determine the measurement maturity level of the company. The results of the exploratory case studies were that MIS-PyME 3M useful for identifying the measurement maturity of the division, which was at level 2, identifying the measurement improvements, and identifying the measurement goals which were suitable for application in the company's Development Division.

However, the case studies lack robustness if they are to completely validate MIS-PyME. More case studies should be performed in other SMEs in order to confirm results and prove the context in which MIS-PyME is a suitable a methodology. More case studies are therefore required if this goal is to be fully achieved.

Main Goal: To define a methodological framework in order to support small and medium-sized enterprises in establishing measurement programs. This methodology should be a lightweight methodology which takes into account the company's maturity and limitations.

Based on the attainment and fulfilment of the sub-goals, the main goal of this research is fulfilled. MIS-PyME provides a useful framework through which to define and implement measurement programs adapted to the restrictions and maturity of SMEs. MIS-PyME consists of a lightweight methodology, support modules with templates, guides and examples which ease the definition of the measurement program, and a measurement maturity model, which is used as a reference for the methodology in order to define measurement programs adapted to the measurement maturity of an SME.

6.2. Results Support

The main results of the research have been published in software engineering related forums. Table 6-1 shows a schema of the relevant book chapters, conference publications and journals. Those papers which are currently at the acceptance stage are marked with the symbol "+".

Type of publication	Number
International Journals	2 +1
Chapters in Books	1
International Conferences	3
National Conferences	1
Latin American Conferences	2
TOTAL	9 + 1

Table 6-1. Summary of Publications

Table 6-2 shows the published papers classified by each theme of the research.

Main Research Theme	Specific Research Theme	Reference
State of the art	Analysis of the existent methodologies for defining and implementing measurement programs	2006-SISOFT 2008-Ra-ma
Basis of the Research	Empirical analysis of the suitability of GQ(I)M for SMEs	2007-PROFES 2008-IET 2008-CLEI
	Analysis of the suitability of the existent methodologies for SMEs.	2008-JISBD 2008-IET
MIS-PyME methodology	MIS-PyME definition and experience	2007-MENSURA 2007-PROFES 2008-IET 2008-CLEI
	MIS-PyME definition and validation	2008-IET
	MIS-PyME Measurement Goals Table	2009-SQP
MIS-PyME Measurement maturity Model	Model presentation and validation as regards usefulness for defining measurement programs adapted to the measurement maturity of a company	2008-PROFES 2008-IET 2009-AES
	Model presentation and validation as regards the usefulness for determining the measurement maturity of a company, identify measurement improvements and suitable measurement goals.	2009-AES

Table 6-2. Papers classified by research themes.

Book Chapters

- (2008-RA-MA). Diaz-Ley, M., García, F., and Piattini, M. (2008). Capítulo 8. Implantación del programas de medición. In Ra-Ma (Ed.), *Medición y Estimación del Software: Técnicas y métodos para mejorar la calidad y la productividad* (pp. 217-234).

Journals

- (2008-IET). Diaz-Ley, M., García, F., and Piattini, M. (2008). Implementing a Software Measurement Program in SMEs- A Suitable Framework. *IET Proceedings Software. 2*(5), 417-436. **Impact Factor = 0,4 (2007 JCR)**
- (2009-SQP). Diaz-Ley, M., García, F., and Piattini, M. (2009). MIS-PyME Software Measurement Goals Table- Supporting the Selection of Measurement Goals based on Measurement Organizational Maturity. *Software Quality Professional (SQP).11*(3)
- (2009-AES). Diaz-Ley, M., García, F., and Piattini, M. (2009). MIS-PyME Software Measurement Capability Maturity Model- Supporting the Definition of Software Measurement Programs and Capability Determination. *Advances in Engineering Software (AES).* **Impact Factor = 0,529 (2007-JCR)** (*submitted).*

International Conferences

- (2007-PROFES). Diaz-Ley, M., García, F., and Piattini, M. (2007). Software Measurement Programs in SMEs - Defining Software Indicators: A methodological framework. In Springer/Heidelberg (Ed.), *Product Focused Software Development and Process Improvement (PROFES'07)* (**LNCS-4589**, pp. 247-261). Riga, Latvia. Conference Rankings[1]: B (2007).
- (2007-MENSURA). Diaz-Ley, M., García, F., and Piattini, M. (2007). Implementing Software Measurement Programs in Non mature Small Settings. In Springer/Heidelberg (Ed.), *MENSURA* (**LNCS- 4895,** pp.154-167). Palma de Mallorca.
- (2008-PROFES). Diaz-Ley, M., García, F., and Piattini, M. (2008). MIS-PyME Software Measurement Maturity Model- Supporting the Definition of Software Measurement Programs. In Springer/Heidelberg (Ed.), *Product Focused Software Development and Process Improvement (PROFES'08)* (**LNCS-5089**, pp. 19-33). Frascati - Monteporzio Catone, Rome, Italy. Conference Rankings[1] B (2007).

Latin American Conferences

- (2006-SISOFT).Díaz, M., García, F., and Piattini, M. (2006). *Defining, Performing and Maintaining Software Measurement Programs: State of the Art.* Paper presented at the IV Simposio Internacional de Sistemas de Información e Ingeniería del Software en la Sociedad del Conocimiento, Cartegena de Indias (Colombia).

---

[1] CORE Conference Rankings http://www.core.edu.au/rankings/Conference%20Ranking%20Main.html

- (2008-CLEI). Diaz-Ley, M., García, F., and Piattini, M. (2008). *MIS-PyME - Un marco metodológico para la definición de Programas de Medición en PyMEs.* XXXIV Conferencia Latinoamericana de Informática (CLEI'08), pp.499-508, Santa Fe, Argentina.

National Conferences

- (2008-JISBD). Diaz-Ley, M., García, F., and Piattini, M. (2008). Metodologías para definir Programas de Medición en PyMEs: El marco MIS-PyME. Paper presented at the 13th Conference on Software Engineering and Databases (JISBD'08), pp.265-274, Gijón, Spain. ISBN: 978-84-612-5820-8 Acceptance Rate: 25%.

6.3. Research Contributions

The result of this research is MIS-PyME, a methodological framework for defining and implementing measurement programs in a small or medium company or in a small or medium division of a large company.

Based on a systematic search in the bibliography, MIS-PyME is the only methodological framework adapted to small or medium sized companies or units which includes complete guidelines for defining and implementing measurement programs. It also comprises a unique methodology which integrates a measurement maturity model in order to define measurement programs adapted to the companies' measurement maturity.

MIS-PyME provides a methodology whose roles and tasks are adapted to the size and characteristics of SMEs, such as the extent to which people are measurement reticent (this is typical in these companies), and the fact that their measurement knowledge is usually limited.

SMEs need the type of support provided by guidelines and templates if they are to be able to apply standards and methodologies (Laporte et al (2008)). Each of the tasks of the MIS-PyME methodology is supported by guidelines which are:

- The MIS-PyME Measurement Goals Table which suggests the common measurement goals which support common process improvement goals.
- The MIS-PyME Indicator Templates which support the development of each of the above measurement goals in order to facilitate the building of the indicators, the analysis of the measurement goals, to integrate the related measurement into the company's other processes and to give advice regarding the measurement maturity required for the company in order to define each common indicator.
- The MIS-PyME Indicator Database which provides examples of indicator implementations

MIS-PyME Framework also provides a measurement maturity model (MIS-PyME 3M) which support the development of measurement programs aligned to the measurement maturity of the company and also aims to assess the measurement maturity of the company and identify improvement measurement suggestions.

6.4. Future Lines of Research

This section presents the future research lines related to this thesis.

Enhance the robustness of MIS-PyME validation: As has already been mentioned, more case studies must be performed in order to replicate the results observed in the research carried out for this thesis. MIS-PyME should be applied in different SMEs which will fulfil the context for which MIS-PyME was designed. MIS-PyME should also be applied in other companies in a different context in order to understand to what extent MIS-PyME is useful in measurement mature companies, in large companies, or in large companies whose measurement maturity is low, etc.

MIS-PyME 3M should also be applied in other companies and the results as regards the capability of MIS-PyME 3M should be explored to determine the measurement maturity of the company, identify the measurement improvement issues and understand the measurement goals that the company is able to implement.

Refining MIS-PyME measurement goals table and indicators templates: The measurement goals proposed in MIS-PyME to support process improvement goals and the related indicator templates are designed to assist SMEs in defining common measurement programs, and thus those that are commonly required in companies and whose goals are common process improvement goals. The measurement goals and templates presented in the first version of MIS-PyME are those that the research group, based on their experience, considered to be the most suitable. However, these should be refined by observing the common process improvement needs in SMEs. Therefore, more experiences which use MIS-PyME in order to refine the measurement goals and templates as regards the common needs of these companies are necessary.

MIS-PyME templates should, moreover, be tailored based on the improvement suggestions reported by the companies which apply this methodological framework. In addition, the MIS-PyME database should include examples of each of the indicator templates in order to understand how the template was tailored for implantation in the company. The first version of MIS-PyME provides few examples, and one of the goals for a future version will be to extend the number of examples.

Integrating MIS-PyME with ISO/EC 29110: Once MIS-PyME has been refined and its usefulness has been demonstrated in a robustness manner, it might be worth analysing the integration of MIS-PyME into ISO/IEC 29110 (ISO/IEC, 2007). This standard is still under construction, but will specify a software process improvement model for very small companies and is based on COMPETISOFT. MIS-PyME could complement ISO/IEC 29110 since it would support the companies which use this standard in defining and implementing a required measurement framework for carrying out an efficient software life cycle processes management.

Appendix

A: COMPETISOFT

B: MIS-PyME Measurement Goals Table

C: MIS-PyME Indicator Templates

D: Measurement Process

E: MIS-PyME 3M Assesment Process and Questionnaire

F: MIS-PyME 3M and ISO 15504 Conformity

G: MIS-PyME and other Methodologies

A. COMPETISOFT

The COMPETISOFT (Oktaba et al., 2007) Software Process Model defines the best practices of software management and engineering for improving its software process. This mode is focused on Latin American SMEs and It conforms to ISO 15504.

COMPETISOFT is based on MoProsoft (Oktaba et al., 2005) which was developed to be set as Mexico's Software Process Standard in order to support SMEs in software process improvement and to make them more competitive in the national and international market. In addition these models are also focused for being applied in small projects of large-sized companies or on small divisions of large-sized companies.

The motivation of MoProsoft was that many of businesses deploy reference models proposed by the Software engineering Institute (SEI), the Capability Maturity Model Institute (CMMI), or the International Organization for Standardization (ISO) was especially difficult to apply in SMEs. These reference models provide complex recommendations and significant time and resource commitment which make their application difficult for small organizations. The situation is especially troublesome for small Latin American organizations due to the absence of tailor-made process reference models, and the adoption of models defined in other countries without suitable adaptation.

In 2005, several researchers and practitioners recognized the importance of an improvement and certification framework for small organizations. They proposed COMPETISOFT to the Ibero-American Science and Technology Development Program (Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo), a group created in 1984 for multilateral scientific and technological cooperation and supported by 21 Latin American countries plus Spain and Portugal. The Participants in COMPETISOFT fell into two main categories:

- researchers from universities in Argentina, Brazil, Chile, Colombia, Costa Rica, Cuba, Ecuador, Mexico, Peru, Portugal, Spain, Uruguay, and Venezuela; and
- the critical reference group, consisting of the Argentinian Institute for Standardization and Certification, the government of Argentina's Neuquén region, and small companies, including five from Colombia, four from Peru, three from Spain, and one each from Argentina, Chile, Ecuador, Mexico, and Uruguay.

COMPETISOFT can be view as an improvement of MoProSoft. In addition COMPETISOFT has become the base model for the new international software process improvement norm for SMEs: ISO 29110.

The main advantage of these models is that they are specific for software development and maintenance. Additionally, it has been defined as a set of processes which are easy to understand and to implement, and these are aimed at improving software processes instead of being mere accreditation frameworks.

The process capacity model establishes 6 maturity levels: Incomplete, Performed, Managed, Established, Predictable, and Optimizing as shown in Figure 3-11.

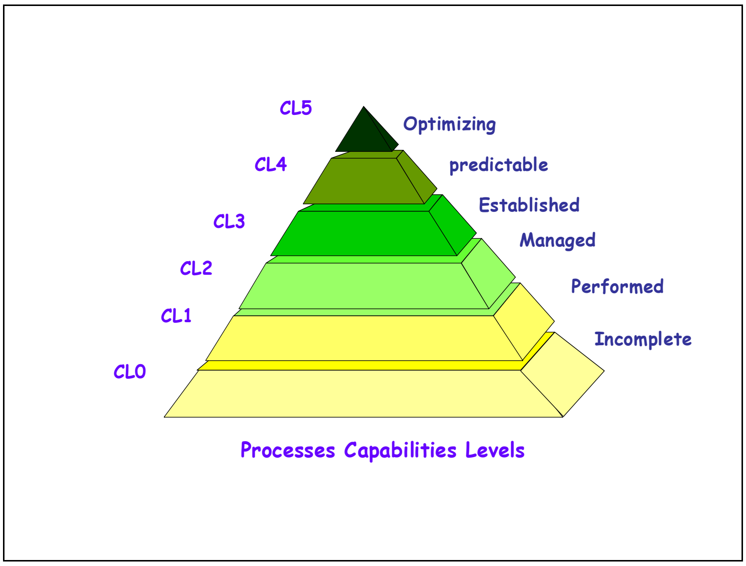


Figure A.1: .COMPETISOFT Process Capability Levels

The software process model is defined by means of six processes organized into three layers (see Figure A.1):

- Top Management layer: It just includes one process, which is the Business Management Process. Its aim is to establish the business goals and the conditions to achieve them. It provides resources and policies to react to an improvement environment and to work focused on the business objectives. Members in this category receive reports from middle management
- Middle Management layer: Members of this category deal with process-, project-, and resource-management practices in line with top management's business goals. They provide elements for the performance of the operational processes, receive and evaluate the information which those processes generate, and inform the top management of the results. It consists of five processes:
    - Process Management Process: It establishes organizational processes based on the required processes as identified in the strategic plan. It also specifies plans and implements activities for process improvement.
    - Portfolio Project Management Process: It ensures that projects meet the objectives and fit in the organizational strategies.

- Resources Management Process: It provides the organization with human resources, infrastructures, providers and a good work atmosphere. It also creates and maintains the organizational experience factory. The aim is to support the achievement of the goals included in the organizational strategic plan.
- Goods, Services and Infrastructure Management: The aim of this process is to provide the goods, services and infrastructure that satisfy the acquisition process and project requirements.
- Knowledge Management: The purpose of this process is to keep available and administrate the knowledge database and the products created in the organization.

- Operational Layer: It includes two processes: Operations. Members of this category address the practices of software-development and -maintenance projects. They perform activities using elements management provides and deliver reports and the software products generated.

  - Specific Project Administration Process: It systematically establishes and carries out those activities which enable the achievement of project goals in terms of costs and time.
  - Software Development Process: It systematically performs the analysis, design, construction, integration and test activities of software products according to requirements as specified.
  - Maintenance Process: The aim of this process is to perform the requested modifications in the software product detailing what has to be done, when, how and by whom.

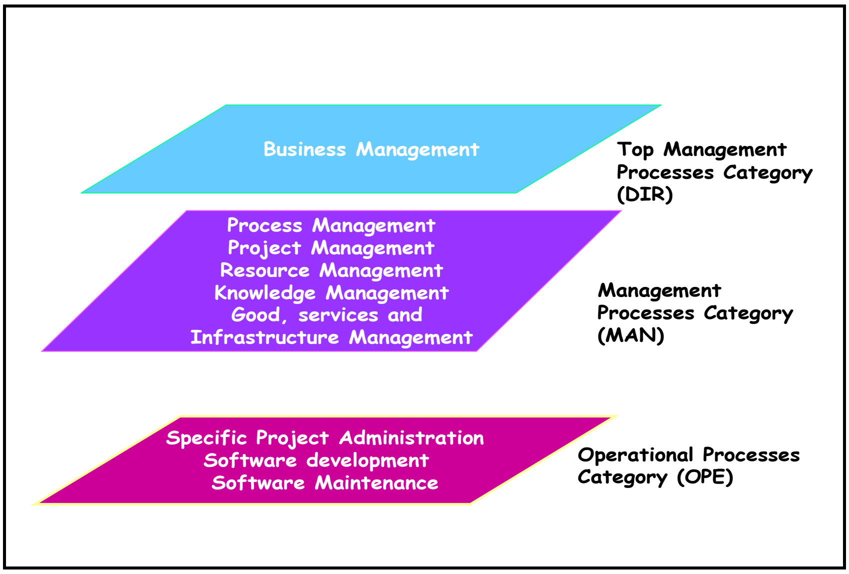


Figure A.2: COMPETISOFT Processes

All processes are clearly defined in templates. This template includes the following information: process category, purpose, description, objectives, indicators, responsibilities, sub-processes, related processes, inputs, outputs, best practices, the specific activities and tasks, resources and infrastructure and adjusting practices.

Additionally, the activities described in the template are coloured depending on the maturity to which these activities correspond.

The roles defined in COMPETISOFT are as follows:

- The Client: The one who asks the software product and funds the project.
- The User: The one who make use of the product.
- The board group: They lead and manage the organization and are responsible of the successful organization performance.
- Process responsible: S/he is the responsible of performing the tasks of a process and the achievement of its goals.
- Involved: There are other required roles for the performance of the specific activities and tasks such as: analyst, programmer, reviewer, and others.

COMPETISOFT also provides with an evaluation and accreditation framework which indicates the state of an organization over a given period of time (PvalCOMPETISOFT). Consequently it provides and improvement and evaluation model. The COMPETISOFT evaluation model is based on the EvalProSoft model. The aim was to help small organizations carry out their assessments by reducing subjectivity and making the process more formal.

The COMPETISOFT improvement model is based on agile SPI, which establishes the elements necessary for economically running improvement programs in small organizations. The model defines PmCOMPETISOFT, an improvement process that follows the process pattern defined in COMPETISOFT. Designed to be easier and more intuitive for small software organizations. PmCOMPETISOFT is a lightweight process that follows an iterative and incremental approach to guide the implementation of an improvement cycle. It is composed of one or more improvement cycles, each one involving five activities: initiating the cycle, diagnosing the process, formulating improvements, executing improvements, and revising the cycle. The model clearly defines these activities by describing the roles involved, the expected work products, and, for each work product, a fully detailed self-content template.

B. MIS-PyME Measurement Goals Table

Level 1	Level 2	P.I.G	Measurement Goals	Indicators
Resources Management	Good, Services and Infrastructure management Process	Improving provider satisfaction	Characterizing the deviation in time, receiving the required service or product in order to understand the tendency and improve it	IND-RES-DelTimeDev
			Characterizing the reliability or suitability of the resources delivered by the each provider in order to understand the tendency and improve it	IND-RES-TimeProblemResolution
Project Management Process	Specific Project Administration Process	Improving project planning	Characterizing the accuracy of the size estimated for the project at its completion in order to improve the size estimation in future projects.	IND-PRJ-InexacSize
			Characterizing the accuracy of the duration estimated for the project at its completion in order to improve the duration estimation in future projects.	IND-PRJ-InexacDuration
			Characterizing the accuracy of expended cost in the project in order to improve this factor in future projects	IND-PRJ-InexacCost
			Characterizing the accuracy of the effort estimated for the project at its completion in order to improve the effort estimation in future projects.	IND-PRJ-EffortDev
		Improving Project Tracking	Monitoring the conformity of the project's progress as regards the schedule, work performed and cost, and what is planned in order to take corrective actions or present problems throughout project execution	IND-PRJ-Progress
			Monitoring the requirements or CU that have passed through the development process's main activities (analysis, design, construction, acceptance, etc.)	IND-PRJ-ReqProgress
			Monitoring requirement stability in order to understand the project's progress	IND-PRJ-ReqStabilityProg

Level 1	Level 2	P.I.G	Measurement Goals	Indicators
Project Management Process	Specific Project Administration Process	Improving Process effectiveness	Monitoring / Evaluating Development process effectiveness, in order to understand it, help to plan future projects and improve the process	IND-PROC-Effectiveness
		Improving Process Compliance	Monitoring / Evaluating the process compliance in order to improve processes and/or take corrective actions	IND-PROC-ProcessCompliance
		Improving Process Efficiency	Monitoring/evaluating process productivity in order to take corrective actions, re-plan the project, improve future plan estimations and improve the process	IND-PROC-Productivity
		Improving client satisfaction	Characterizing/Evaluating the client's satisfaction with the project and the product developed in order to improve it.	IND-PRJ-ClientSatisfaction
			Monitoring the productivity of the projects being developed in order to improve planning of future projects and carrying out process improvements	IND-PRJORG-ProcessProductivity
			Characterizing the projects developed in terms of effort, cost and duration deviation in order to detect common problems, identify improvements and meet business goals.	IND-PRJORG-GlobalProjectManagement
	Software Development Process	Improving the reliability of the product	Monitoring/Evaluating the reliability of the product in order to understand whether the product is sufficiently reliable to go though to the next phase or be delivered to the client or to improve it.	IND-PROD-DensityFailures
			Monitoring the number of accepted open defects during the software development activities, in order to understand the work remaining to improve reliability	IND-PROD-OpenInc
			Monitoring the fault tolerance of the product in order to understand it and improve recovery functions when system breakdowns occur	IND-PROD-FaultTolerance
			Monitoring/Evaluating the test coverage in order to understand how many of the products have been tested	IND-PROD-TestCoverange

Level 1	Level 2	P.I.G	Measurement Goals	Indicators
Project Management Process	Software Development Process	Improving the maintainability of the product	Monitoring/evaluating the % of repeated code in order to take corrective actions	IND-PROD-RepeatedCode
			Monitoring/Evaluating the Cyclomatic complexity of the modules in order to take corrective actions	IND-PROD-CyclomaticComplexity
			Monitoring the module size in order to take corrective actions	IND-PROD-ModuleSize
			Monitoring/evaluating the dependencies between modules in order to take corrective actions	IND-PROD-ModuleDependency
			Monitoring/Evaluating the unit test coverage in order to understand how much of the product has been tested	IND-PROD-TestCoverange
		Improving the efficiency of the product	Monitoring the user time for the most frequently used or critical requests in order to take corrective actions	IND-PROD-TImeRequest
			Monitoring the transaction throughput of the system in order to take corrective actions if necessary	IND-PROD-TransactionThroughput
	Software Maintenance Process	Improve maintenance planning	Monitoring/Evaluating the accuracy of the effort estimated for each maintenance action in order to take corrective actions	IND-PRJ-MeanEfforDev
		Improving the quality of the product	See the development of process improvements: Improve the reliability of the product; Improve the maintainability of the product; Improve the efficiency of the product	
		Improving the maintenance process efficiency	Monitoring/Evaluating the mean effort spent on fixing urgent defects, planned defects and requests for change	IND-PRJ-MeanEffort
			Monitoring/Evaluating mean maintenance downtime in order to take corrective actions	IND-PRJ-MeanDownTime
		Improving the client satisfaction	Characterizing/Evaluating the client's satisfaction with the project and the product developed in order to improve it.	IND-PRJ-ClientSatisfaction

Table B-1. MIS-PyME Measurement Goals Table

C. MIS-PyME Indicator Templates

This section shows the indicator templates proposed by MIS-PyME which follow the structure proposed in the MIS-PyME measurement goals table.

Resource Management Process

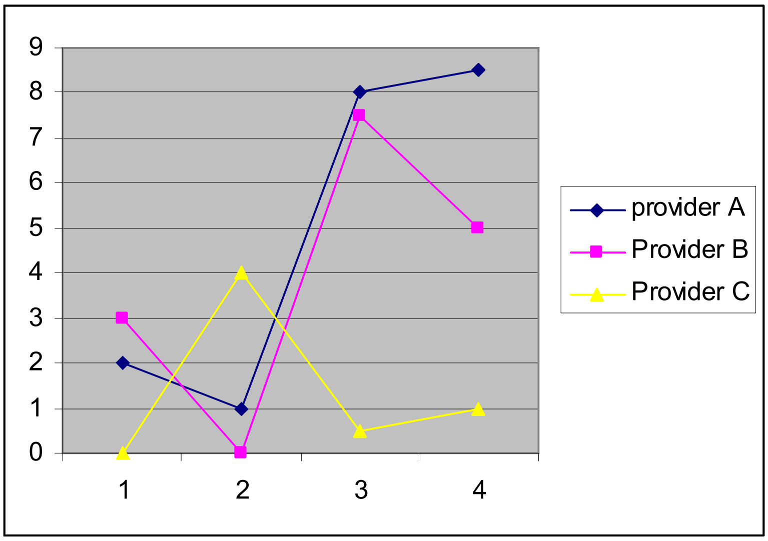
Goods, Services and Infrastructure Management process

IND-RES- DelTimeDev
➢ **Description:** Characterizes the deviation in time needed to receive the required service or product from the provider in order to take corrective actions if required. ➢ **Entity to be measured:** This indicator includes related information for each provider, the analysis is performed individually (RES). ➢ **Point of view**: resources manager and services and infrastructure manager. ➢ **Context:** Software development projects and maintenance in the company. ➢ **Questions**: o Q1: What is the mean time each provider takes to deliver the required resources or products? o Q2: Is it necessary to take action to improve this aspect? ➢ **Inputs:** o MED-RES-DEV: Date received - date required (Unit: days). This derived measure is collected when the service or resource is provided and it is saved in the providers' catalogue. ➢ **Algorithms**: Q1: Delivery Mean time for each provider= Sum([(MED-RES-DEV) of the analysed period] / total of the measure results. ➢ **Measuring activity information:** o **Those responsible for analysis and feedback:** resources manager and goods, services and infrastructure manager. o **Analysis and feedback frequency:** Ej. Every month or every four months, depending on the needs. o **Destination of the analyses and interpretation results, audience:** Processes management process, knowledge management process and resources management.

- **Indicator results report location: "**services and infrastructure resources report", "measures and improvement suggestions report", and lessons learned.
- **Integration:** This indicator is necessary in the services and infrastructure process reporting activity.
- **Resources required:**
  - The knowledge base (providers' catalogue / database).

➢ **Analysis / Interpretation:** This indicator helps the resources manager to objectively understand the provider's tendency when delivering the required software, license or whatever resource is required. The resources manager can control the provider, improve it, or even suggest that the executive business group take serious actions with regard to this issue. S/he can also suggest improving certain issues of the process.

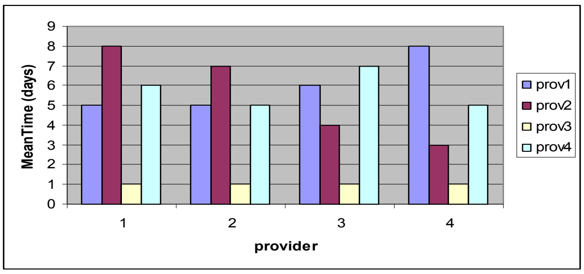
➢ **Output Display.** The graphical display provides the mean time results for each provider and its evolution in time as regards previous analyses, at a glance.



IND-RES-TimeProblemResolution

- **Description:** Characterizing the mean time to resolve problems or failures per each resource and each provider.
- **Entity to be measured:** This indicator includes related information for each resource and provider, the analysis is performed individually (RES).
- **Point of view**: resources manager and services and infrastructure manager.
- **Context:** Software development projects and maintenance in the company.
- **Questions**:
  - Q1: What is the mean time each provider takes to resolve a problem for each resource?
  - Q2: What is the tendency?
  - Q3: Should any action be taken to improve results?
- **Inputs:**
  - MED-RES-ProbSolving: Date reported - date when the problem is completely resolved. (Unit: natural days). This measure is obtained when the problem is completely resolved
- **Algorithms**:

  Q1. MED-RES-ProbSolving [during the period analysed]/#problems.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** resources manager and goods, services and infrastructure manager.
  - **Analysis and feedback frequency:** Ej. every month or every four months, depending on the number of resources and the needs.
  - **Destination of the analyses and interpretation results, audience:** Processes management process, knowledge management process and resources management.
  - **Indicator results report location:** “services and infrastructure resources report”, “measures and improvement suggestions report” and lessons learned.
  - **Integration:** This indicator is required in the services and infrastructure process “reporting” activity.
  - **Resources required:**
    - The knowledge base/ information provider database
    - Problem, defects tracking system.
- **Analysis / Interpretation:** This indicator helps the resources manager to monitor the providers, improve them, or even suggest that the executive business group take serious action with regard to this issue.
- **Output Display.** The graphical display provides the mean time results for each resource and provider, at a glance.



Project Management Process

IND-PRJORG-GlobalProjectManagement

- **Description:** Characterizing the projects developed in terms of effort, cost and duration deviation in order to detect common problems, identify improvements and meet business goals.
- **Point of view**: Person responsible for portfolio project process.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What was the deviation in time of the projects developed in the period of time?
  - Q2: What was the deviation in effort of the projects developed in the period of time?
  - Q3: What was the deviation in cost of the projects developed in the period of time?
  - Q4: What was the deviation in size of the projects developed in the period of time?
  - Q5: For each of the above aspects, are there any common problems? Are there any common improvement suggestions?
  - Q6: Understanding previous results, what is the tendency for each aspect?

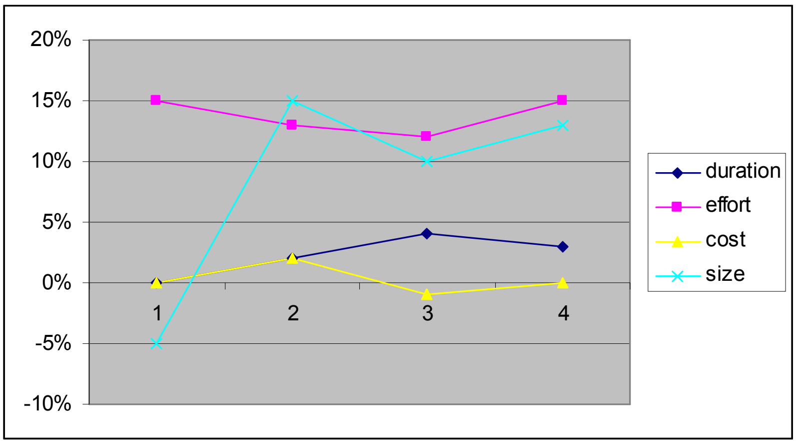
This analysis can be performed for each type of project, e.g.: small, medium or big.

- **Inputs:** For all the closed software development projects:
  - IND-PRJ-InexacSize.
  - IND-PRJ-InexacDuration.
  - IND-PRJ-InexacCost.
  - IND-PRJ-EffortDev.
  - Previous IND-PRJORG-GlobalProjectManagement.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for portfolio project process.
  - **Destination of the analyses and interpretation results, audience:** person responsible for process management, person responsible for business management.
  - **Analysis and feedback frequency:** e.g. every six months but this depends on the number of projects.
  - **Indicator results report location:** Measurements and improvement suggestions report, lessons learnt, qualitative and quantitative report.

- **Integration:** This indicator should be analysed for the project portfolio management process, and for the follow-up activity.
- **Resources required:** Organizational database.

➢ **Analysis:** This information would be useful to understand the deviations in projects, common causes of problems and common improvement suggestions.

➢ **Graphic Display:** This would display the progress of the results of these analyses in terms of duration deviation, cost deviation, effort deviation and size deviation.



IND-PRJORG-ProcessProductivity

➢ **Description:** Monitoring the productivity of the projects being developed in order to improve the planning of future projects and carrying out process improvements.

➢ **Point of view**: Portfolio project manager.

➢ **Context:** Software development projects.

➢ **Questions**:

- Q1: What is the median of the productivity for UC/requirements specification of the projects completed in the period analysed?
- Q2: What is the median of the productivity for coding phase of the projects completed in the period analysed?
- Q3: What is the median of the productivity for testing phase of the projects completed in the period analysed?
- Q5: Is it necessary to take any improvement action in the process?

This analysis could be performed for type of product being developed in the project, or type of project.

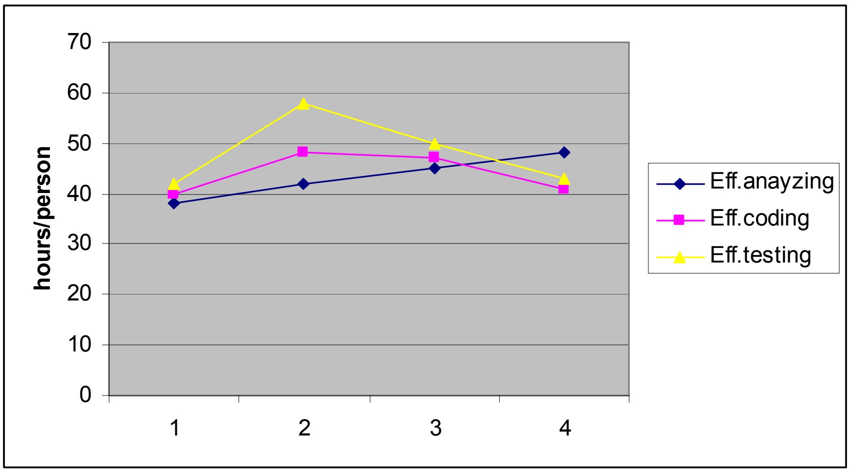
➢ **Inputs**

- IND-PRJ-ProcessProductivity.
- IND-PRJORG-ProcessProductivity of previous analyses.

- **Algorithm**
  - At the end of the project, the indicator IND-PRJ-ProcessProductivity should contain the median of the productivity in the project.
  - A median of the above values provided in IND-PRJ-ProcessProductivity of each project completed in the project analysed is performed.

The median is the number separating the higher half of the productivity values of each project.

- **Improvement of the indicator**
  - Once the company is experienced in analysing this information, and there is sufficient historical data and reliable data, normal ranges of productivity can take place and evaluation (good, normal, bad) can be performed.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for portfolio project process.
  - **Destination of the analyses and interpretation results, audience:** person responsible for process management, person responsible for business management.
  - **Analysis and feedback frequency:** e.g. every six months, but this depends on the number of projects.
  - **Indicator results report location:** Measurements and improvement suggestions report, lessons learnt, qualitative and quantitative report.
  - **Integration:** This indicator should be analysed for the project portfolio management process, and for the follow-up activity.
  - **Resources required:** Organizational database.
- **Analysis:** This information will be useful for the person responsible for project portfolio management in order to improve estimation and to suggest any process improvement if the productivity is considered low.
- **Graphical Display**
  - There will be an indicator for the median time progress of the following: UC analyses, UC coding and UC testing.



Specific Project Administration Process

Improve Project Planning

IND-PRJ-INEXACSIZE
➢ **Description:** Characterizing the accuracy in terms of the estimated size of the product developed and actual size at its completion in order to improve the size estimation in future projects. ➢ **Point of view**: Project manager. ➢ **Context:** Software development projects. ➢ **Questions**: o Q1: What was the total deviation in terms of size of the product developed in the project? o Q2: What were the causes of the deviation? New requirements? Problems using estimation methods? Poor historical information? o Q3: Are any improvements necessary? ➢ **Inputs:** o Estimated Size (which could be measured in function points (IFPUG, Fisma, COSMIC, etc.), use case points (LOC, etc.) o Actual Size (which should be measured with the same unit of measure). ➢ **Algorithms**: Q1. (size_actual - size_estimated) * 100 / size_estimated. ➢ **Measuring activity information:** o **Those responsible for analysis and feedback:** Project manager. o **Destination of the analyses and interpretation results, audience** person responsible for process management, person responsible for portfolio project process. o **Analysis and feedback frequency:** At the project's completion. o **Indicator results report location:** Measurements and improvement suggestions report. o **Integration:** This indicator should be analysed for the specific project management process, at its completion, and included in the reports indicated above. o **Resources required:** project estimation methods. ➢ **Analysis:** This information would be useful for future projects and for the person responsible for process management in order to understand whether any improvement to the process should be performed.

IND-PRJ-INEXACDURATION

- **Description:** Characterizing the accuracy of the duration estimated for the project at its completion in order to improve the duration estimation in future projects.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What was the total deviation in terms of time of the project?
  - Q2: What were the causes of the deviation?
  - Q3: Are any improvements necessary?
- **Inputs:**
  - Date when the project formally started. The starting point of the project should be formally defined. This could be the date when the client and the project manager start working on the requirements once the project has been formally assigned to the company.
    - Date_planned_start: Date indicated in the last plan, signed with the client, to start the project.
  - Date when the project formally finished. The starting point of the project should be formally determined. This could be the date when the last realization of the product related to the project is delivered to the client and the starting point of the maintenance phase.
  - Two pieces of data must be obtained:
    - Date_planned_finish: Date indicated in the last plan, signed with the client, to complete the project.
    - Date_actual_finish: Real date that the project finished.
- **Algorithms**:

  Q1. (date_actual_finish - Date_planned_start) - (date_planned_finish - date_planned_start) * 100 / (date_planned_finish - date_planned_start).
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience:** person responsible for process management, person responsible for portfolio project process.
  - **Analysis and feedback frequency:** At the project's completion.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the specific project management process, at its completion, and included in the reports indicated above.
  - **Resources required:** project estimation methods.
- **Analysis:** This information would be useful for future projects and for the person responsible for process management in order to understand whether any improvement to the process should be performed

IND-PRJ-EffortDev

- **Description:** Characterizing the deviation in terms of the project effort in order to improve management in future projects.
  - It is not necessary to use this indicator if the above is being used.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What was the total deviation in terms of the project effort?
  - Q2: What were the causes of the deviation?
  - Q3: Are any improvements necessary?
- **Inputs:**
  - E1: Planned effort (measured from the last plan signed with the client).
  - E2: Actual effort.
- **Algorithms**:

Q1. (E2 - E1) * 100 / E1

- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience** person responsible for process management, person responsible for portfolio project process.
  - **Analysis and feedback frequency:** At the project's completion.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the specific project management process, at its completion, and included in the reports indicated above.
  - **Resources required:** project estimation methods.
- **Analysis:** This information would be useful for future projects and for the person responsible for process management in order to understand whether any improvement to the process should be made.

IND-PRJ-COSTDEV

- **Description:** Characterizing the deviation in terms of the project's cost in order to improve management in future projects.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What was the total deviation in terms of the project's cost?
  - Q2: What were the causes of the deviation?
  - Q3: Are any improvements necessary?
- **Inputs:**
  - C1: Planned cost (included in the contract with the client).
  - C2: Actual cost. This measure can be obtained with the following formula: SUM([hours spent]n *cost_n/hour * n_resources]); n=each human resource category + [other material and resources required for the project].
- **Algorithms**:

  Q1. (C2 - C1) * 100 / C1
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience** person responsible for process management, person responsible for portfolio project process.
  - **Analysis and feedback frequency:** At the project's closure.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the specific project management process, at its conclusion, and included in the reports indicated above.
  - **Resources required:** project estimation methods.
- **Analysis:** This information would be useful for future projects and for the person responsible for process management in order to understand whether any improvement to the process should be made.

IND-PRJ-REQSTABILITY

- **Description:** Characterizing the deviation in terms of the project requirements in order to improve management in future projects.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
    - Q1: What was the total deviation in terms of the project requirements?
    - Q2: What were the causes of the deviation? More time and new techniques should be dedicated to the capture of requirements phase …
    - Q3: Are any improvements proposed?
- **Inputs:**
    - N1: Number of requirements that were changed after the requirements specifications had been verified and/or accepted.
    - N2: Number of requirements at the project's completion.
- **Algorithms**:

Q1. (N2 - N1) * 100 / N1

- **Measuring activity information:**
    - **Those responsible for analysis and feedback:** Project manager.
    - **Destination of the analyses and interpretation results, audience** person responsible for process management.
    - **Analysis and feedback frequency:** At the project's completion.
    - **Indicator results report location:** Measurements and improvement suggestions report.
    - **Integration:** This indicator should be analysed for the specific project management process, at its completion, and included in the reports indicated above.
    - **Resources required:** project estimation methods.
- **Analysis:** This information would be useful for future projects and for the person responsible for process management in order to understand whether any improvement to the process should be made.

Improve Project Tracking

IND-PRJ-Progress

- **Description:** Monitoring the conformity of the progress of the project as regards the schedule, work performed and cost and what is planned in order to take corrective actions or present problems throughout the project execution.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What is the schedule performance variance for each phase / iteration?
  - Q2: What is the effort performance variance for each phase / Iteration?
  - Q3: What is the total schedule performance variance?
  - Q4: What is the total effort performance variance?
  - Q5: Is it necessary to take any corrective action? What?
- **Inputs:**
  - Budgeted Cost of Work Schedule (BCWS): Effort planned for the work scheduled at the time of the analysis.
  - Budgeted Cost of Work Performed (BCWP): Effort planned for the work performed at the time of the analysis.
  - Actual Cost of Work Performed (ACWP): Actual effort of the work performed at the time of the analysis.
- **Algorithms**:
  - Q1. (BCWP - BCWS) / BCWS
  - Q2: (BCWP - ACWP) / ACWP
  - Q3. (BCWP - BCWS) / BCWS (taking into account the work scheduled for that iteration or phase)
  - Q4: (BCWP - ACWP) / ACWP (taking into account the work scheduled for that iteration or phase)
- **Improvement of the indicator**

Once the company is experienced in performing and analysing this indicator, other indicators should be added as input to this indicator in order to better understand the work performed and left:

  - IND-PRJ-ReqProgress.

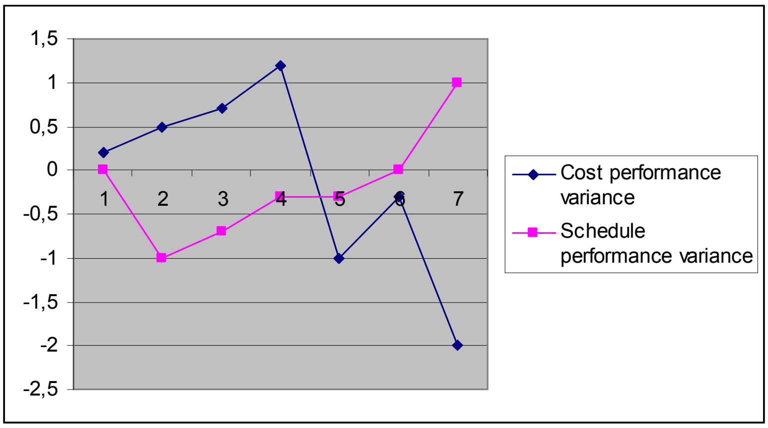
- **Measuring activity information:**

- **Those responsible for analysis and feedback:** Project manager.
- **Destination of the analyses and interpretation results, audience:** Person responsible for project portfolio management; person responsible for software development.
- **Analysis and feedback frequency:** Periodically (e.g. once a week, once a month, depending on the project).
- **Indicator results report location:** project follow-up report.
- **Integration:** This indicator should be analysed for the specific project management process during the evaluation and control activity.
- **Resources required:** project management tool / effort management tool (to include actual effort), requirement management tool.

➢ **Analysis:** This information will be useful for the project manager and the person responsible for project portfolio management in order to understand the progress of the project and take action in the case of problems.

Results near zero indicate that the project is proceeding according to plan. Negative results are an indication that the project is behind schedule or over budgeted cost. Positive results indicate the project is ahead of schedule or under budgeted cost.

Graphical Display. This should show at a glance the schedule performance variance and the effort performance variance (for the whole project and for each phase/iteration)



IND-PRJ-ReqProgress

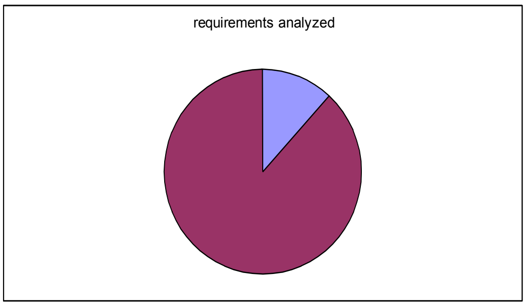
- **Description:** Monitoring the state of the requirement in order to understand the progress of the project
- **Point of view**: Project manager
- **Context:** Software development projects
- **Questions**:
  - Q1: How many requirements or UCs have passed each phase: analyses and design, coded and verified and accepted (or for each iteration)?
  - Q2: What are the corrective actions?
- **Inputs:**

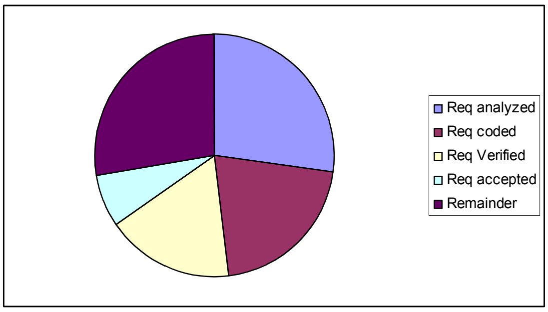
Requirement_state (The requirements management tool should have an attribute for each requirement or UC in order to manage their state).

- **Algorithms**:

P1. #requirements_n * 100/ #requirements (total number of req); "n" is the state which is being analysed: analysing + designing, coding + verifying, accepting. If the development process is performed by iterations, "n" would be the iteration.

- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience:** Person responsible for project portfolio management; person responsible for software development.
  - **Analysis and feedback frequency:** once a week or once a month.
  - **Indicator results report location:** project follow-up report.
  - **Integration:** This indicator should be analysed for the specific project management process during the evaluation and control activity.
  - **Resources required:** Requirements management tool.
- **Analysis:** This information will be useful for the person responsible for project portfolio management and for the specific project in order to understand the progress of the project and take action in the case of problems.
- **Graphical Display.** This should show, at a glance, the progress of the requirements when going through the controlled activity. UC could be used as an alternative.

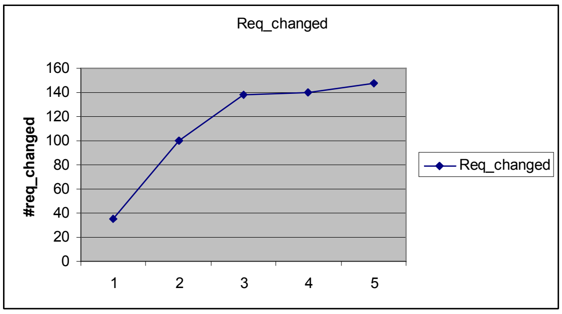




IND-PRJ-ReqStabilityProgress

- **Description:** Monitoring the stability of the requirements in order to understand the progress of the project.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What percentage of requirements has been added, modified or deleted?
  - Q2: What is the progress of requirements change (added, modified or deleted)? After the requirement specifications were verified and accepted?
  - Q3: Is it necessary to take any corrective action?
- **Inputs:**
  - Req_changed: Number of requirements that were added, modified or deleted once the iteration or project had passes to the design phase.
  - Req_total: total number of requirements in the project.
- **Algorithm**
  - # req_changed /req_total
  - req_changed
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience:** Person responsible for project portfolio management; person responsible for software development.
  - **Analysis and feedback frequency:** every fifteen days, once a week; at least the same frequency as the follow-up report of the project has to be delivered.
  - **Indicator results report location:** Measurements and improvement suggestions report and/or project follow-up report.
  - **Integration:** This indicator should be analysed for the specific project management process during the evaluation and control activity.
  - **Resources required:** Requirements management tool.
- **Analysis:** This information will be useful for the project manager and the person responsible for portfolio management in order to understand the possible problems in the project and take corrective actions.
- **Graphics**

The indicator should show the progress of requirements change throughout the project. For each follow-up meeting, the number of requirements changed is shown.



Software Development Process Administration

IND-PRJ-Effectiveness

- **Description:** Monitoring/evaluating the effectiveness of the development process in order to understand it and improve future projects.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - What is the progress performance of the development process like, and what are the consequences of poor performance?
    - P1.1: How many defects have been detected in the integration tests? (classify the defects as low, medium and high).
    - P1.2: How many defects have been detected in the acceptance tests? (classify the defects as low, medium and high).
    - P1.3: What action could be taken in order to improve this aspect?
  - What is the performance of the test phases like, and what are the consequences of poor performance?
    - P2.1: How many defects were detected once the product was delivered to the client?
    - P2.2: What action could be taken in order to improve this aspect?
- **Inputs:**
  - DTI: Defects (high, medium and low) detected in integration tests.
  - DTA: Defects (high, medium and low) detected in acceptance tests.
  - DP: Defects detected in the client site for two months (more or less) since the product was delivered to the client.
- **Algorithms**:
  - **P1.2 :** DTI.
  - **P1.3 :** DTA.
  - **P2.1:** DP.
- **Improvement of the indicator**

Once the company is experienced in analysing this information, and there is sufficient historical data, normal ranges should be identified and therefore evaluations (good, normal, bad) can be performed.

- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience:** Person responsible for project portfolio management; person responsible for software development.
  - **Analysis and feedback frequency:** After the analysed phase has been completed.
  - **Indicator results report location:** Measurements and improvement suggestions report and/or

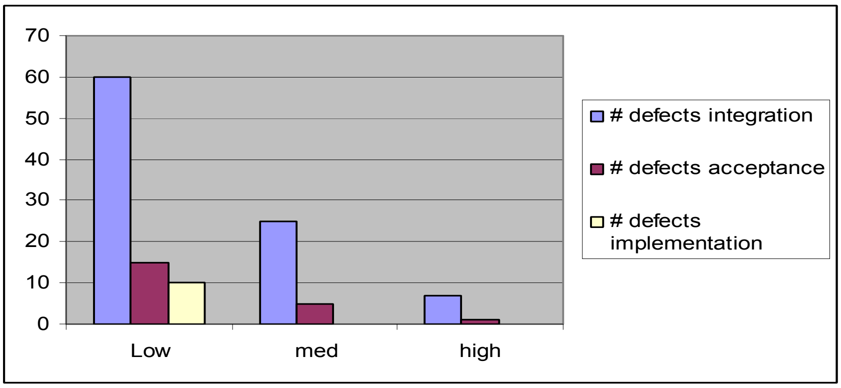
project follow-up report.

- **Integration:** This indicator should be analysed for the specific project management process during the evaluation and control activity.
- **Resources required:** defect tracking tool.

➢ **Analysis:** This information will be useful for the project manager and person responsible for project portfolio management in order to understand the efficiency of the process. This information helps to suggest improvements to the process.

➢ **Graphical Display.**

This can display the number of defects detected in each test phase: integration and acceptance, and once the product is in production and classified by impact.



IND-PRJ-ProcessCompliance

➢ **Description:** Monitoring the compliance of the process performed in the project against the process defined and planned.

➢ **Point of view**: Project manager.

➢ **Context:** Software development projects.

➢ **Questions**:

- Q1: By following the standard process defined, have the specified tasks been performed? Which have not? Why have these tasks not been performed? Are they necessary? Is it necessary to take any corrective action?
- Q2: Are there any tasks that are not being properly performed, and therefore any output products that have not been developed or not developed as they should have been? What are these activities and products? Are they necessary? If they are not necessary explain why, otherwise, explain the corrective actions.
- Q3: Has the project manager clearly identified and communicated the role and the responsibilities of the role to each person in the project team?
- Q4: Are roles understood by the project team? Are they undertaking their responsibilities as they

should?

- Q5: At the end of the project: what is the profile of the process performance?
- Q6: What are the improvement suggestions of this evaluation?

- **Inputs:**
  - Standard process specification in the company.
  - Reference model evaluation (e.j: ISO/IEC 15504).
- **Algorithms**:
  - In order to understand the process compliance throughout the project, the project manager should compare the reference process and the process performed in the project, understand the difference and take corrective actions.
  - In order to assess the process at its conclusion, the evaluation process specified for the reference process model should be followed
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager
  - **Destination of the analyses and interpretation results, audience:** Person responsible for project portfolio management; person responsible for process management.
  - **Analysis and feedback frequency:** Periodically, and this frequency depends on the type of project and the project's completion.
  - **Indicator results report location:** Measurements and improvement suggestions report and project follow-up report.
  - **Integration:** This indicator should be analysed for the specific project management process during the evaluation and control activity.
- **Analysis:** This information will be useful for the project manager and person responsible for project portfolio management in order to take corrective actions in the process. In addition, it will be useful for the person responsible for process management in order to improve or adapt the standard process, train people in those activities in which there are performance problems, oblige project managers to follow the standard process, etc.

IND-PRJ-ProcessProductivity

- **Description:** Monitoring/evaluating the productivity of the project in order to take corrective actions if the productivity is under planned, and improve future project estimations.
- **Point of view**: Project manager.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What is the productivity progress for UC/requirements specification?
  - Q2: What is the productivity progress during the coding phase?
  - Q3: What is the productivity progress during the test specification and performance phase?
  - Q4: At the end of the project, what is the final productivity for each of the above activities?
  - Q5: Is it necessary to take any corrective action?
- **Inputs**
  - Eff.anayzing: Effort required to specify each UC with client.
  - Eff.coding: Effort required to code each UC and perform unit test.
  - Eff.testing: Effort required to specify and perform each UC.

The above questions could be related to Function Points or components. We do not recommend the use of LOC.

- **Algorithm**
  - The median of the above measures [eff.anayzing]; [eff.coding]; [eff.testing] is obtained for each of the developers analysers or testers (depending on development phase) and the period analysed.
  - At the end of the project a median of the above values is calculated, taking into account each development phase.

The median is the number separating the higher half of the effort values received from the measures.

- **Improvement of the indicator**

Once the company is experienced in analysing this information, and there is sufficient historical data and reliable data, normal ranges of productivity can take place and evaluation (good, normal, bad) can be performed.

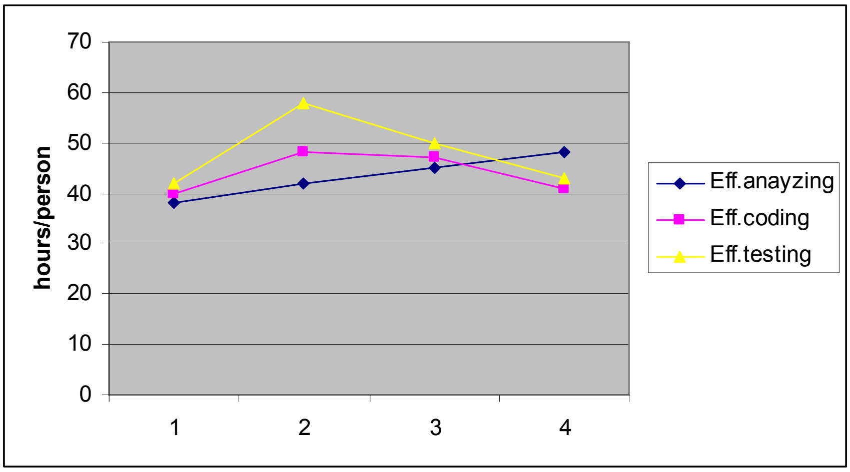
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Project manager.
  - **Destination of the analyses and interpretation results, audience:** Person responsible for project portfolio management; person responsible for software development.
  - **Analysis and feedback frequency:** Periodically during the project and at the project’s closure.

- Indicator results report location: Measurements and improvement suggestions report and project follow-up report.
- **Integration:** This indicator should be analysed for the specific project management process during the evaluation and control activity.
- **Resources required:** Project database; project management tools for tracking effort.

➢ **Analysis:** This information will be useful for the project manager and person responsible for project portfolio management in order to improve estimation, and to understand whether the estimations performed at the beginning of the project were good and the project will be able to meet the duration of the project and efforts planned.

➢ **Graphical Display**

There will be an indicator for the median time progress of the following: UC analyses, UC coding and UC testing.



Client Satisfaction

IND-PRJ-ClientSatisfaction

➢ **Description:** Characterizing the client satisfaction with the project and the product developed .

➢ **Point of view**: Project manager .

➢ **Context:** Software development projects.

➢ **Questions**: Some of the questions that should be put to the client are as follows:

- Q1: Are you satisfied with the achievement of the milestones in the project in terms of time and suitability? (Specific data of the project with regard to this should be included. Indicating what the milestone were, the deviation in time for achieving those milestones, etc.)
- Q2: Are you satisfied with the fulfillment of the requirements of the project? (Specific data of the project with regard to this should be included. Requirements which were modified or deleted since there were expensive to implement, etc.)

- Q3: Are you satisfied with the communication channels? Was it easy to find the right person to solve the problem …
- Q4: Are you satisfied with the customer support? As regards the way in which doubts were resolved, whether solutions were adapted to the company, etc.
- Q5: Do you find the product friendly and easy to manage?
- Q6: Are you satisfied with the quality of the product? (Quality information about the product should be provided: number and impact of the failures which have appeared in production before this survey should be provided.)
- Q7: Are you satisfied with the general expectations of the product?
- Q8: What aspects do you think we should improve?

➢ **Inputs:**

The survey results

➢ **Algorithms**:

From question 1 to 7, the client should indicate the importance of this aspect and the satisfaction value. For example, levels of importance could be (IL):

- Not too important: 3.
- Important: 2.
- Very important: 3.

And the satisfaction values could be (SV):

- Very satisfied: 5.
- Satisfied:4.
- Neutral: 3.
- Not satisfied: 2.
- Dissatisfied: 1.

The value of this indicator would be: SUM(SV*IL)n / #questions answered with a value. "n" is each of the question in which a satisfaction value was indicated.

➢ **Measuring activity information:**

- **Those responsible for analysis and feedback:** Project manager
- **Destination of the analyses and interpretation results, audience:** portfolio project manager, business manager, headquarters of the company, person responsible for software development.
- **Analysis and feedback frequency:** This indicator should be analysed a month after the product has been delivered to the client.
- **Indicator results report location:** Measurements and improvement suggestions report.
- **Integration:** This indicator should be analysed for the project management process, during the closing phase.

➢ **Analysis:** This information will be useful for understanding the satisfaction of the customer with the company management and the product developed. This indicator is very useful for understanding weakness and improvement issues.

Software Development Process

Product Reliability

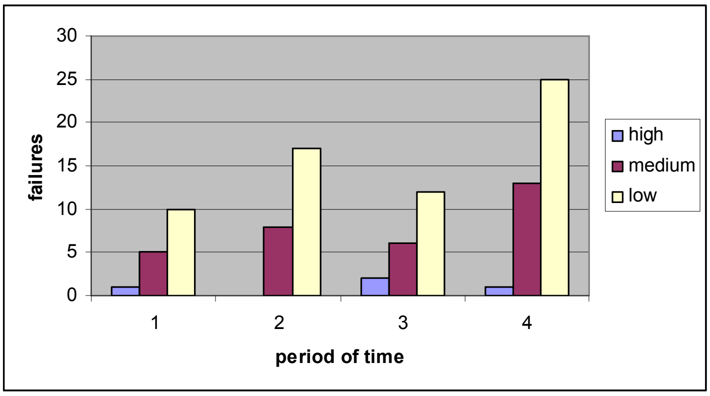
IND-PROD-DensityFailures
➢ **Description:** Monitoring/Evaluating the density of failures of the product in order to understand whether the product is sufficiently reliable to pass to the next phase or be delivered to the client or improved. ➢ **Point of view**: o In development: person responsible for Software development o In production: Project maintenance manager. ➢ **Context:** Software development and maintenance projects ➢ **Questions**: o Q1: What is the progress of failures density? o Q2: Is it necessary to take any corrective action? ➢ **Inputs:** o Production: Failures found in the product in production in a unit of time (e.g. in a month in production). This measure will return the total number of failures, classified by impact (low, medium or high). o Development: Failures found during the test phases in a unit of time (e.g. in a week of testing). This measure will return the total number of failures and will be classified by impact (low, medium or high). ➢ **Improvement of the indicator** An improvement of the indicator would be to understand the normal ranges of the product's reliability when in production and to take into account the type of product . ➢ **Measuring activity information:** o **Those responsible for analysis and feedback:** Person responsible for development and maintenance. o **Destination of the analyses and interpretation results, audience:** ▪ In development: Project manager. ▪ In production: project portfolio manager, business manager o **Analysis and feedback frequency:** Periodically and based on the needs. o **Indicator results report location:** ▪ In development: Measurements and improvement suggestions report. This report could

also be included in the project follow-up report.

- In production: qualitative and quantitative report, measurements and improvement suggestions report.

- **Integration:** .
  - In development: This indicator should be analysed during the development process and especially during the acceptance phase.
  - In production: This indicator should be analysed during the specific maintenance management process and during the portfolio project management process.
- **Resources required:** Incident / defect tracking tool.

- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product is ready to be delivered to the client. If the density of failures is still high and it has not been reduced, the reliability of the product is not still good enough, etc. In addition, in the maintenance process this indicator will be useful for identifying problems with the product, planning the maintenance team, etc.
- **Graphical Display.** This should show at a glance the progress reliability of the product.



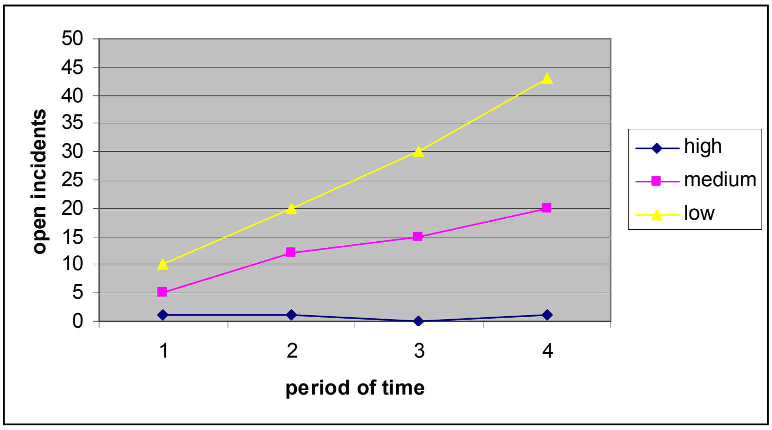
IND-PROD-FaultTolerance

- **Description:** Monitoring the fault tolerance of the product in order to understand it and improve recovery functions when breakdowns occur in the system.
- **Point of view**: Project manager and person responsible for software development.
- **Context:** Software development and maintenance projects.
- **Questions**:
  - Q1: How many system breakdowns occurred during the product tests or in production?
  - Q2: For how many of the above failures was there a solution to recover the system?
  - Q3: Is it necessary to take any corrective action?
- **Inputs:**
  - F1: BreakDown failures.
  - F2: BreakDown failures that could be recovered.
- **Algorithms**:

  P1. F2/ F1.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development and maintenance.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically and based on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed in the development process and especially during the acceptance phase and in the maintenance process.
  - **Resources required:** Incident / defect tracking tool.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand the actual reliability of the product and whether recovery methods should be developed to improve the product's reliability.

IND-PROD-OpenInc

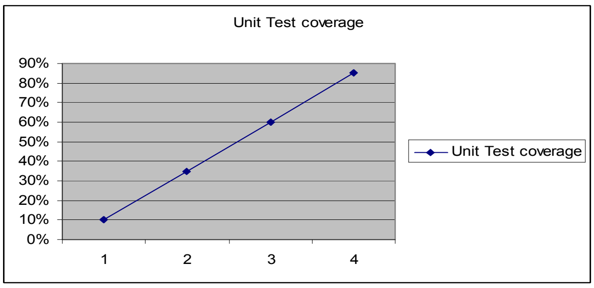
- **Description:** Monitoring the incidents that remain open in order to understand the remaining work to improve reliability.
- **Point of view**: Project manager and person responsible for software development.
- **Context:** Software development and maintenance projects.
- **Questions**:
  - Q1: How many incidents continue open?
  - Q2: Is it necessary to take any corrective action?
- **Inputs:**
  - Q1: Number of open incidents. This measure will return the number of open incidents and classify them by impact.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development process.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically and based on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed in the development process and especially during the acceptance phase.
  - **Resources required:** defect tracking tool.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand the actual reliability of the product and the remaining work that should be performed in order to improve the reliability.
- **Graphical Display.** This should show the progress of open incidents at a glance.



IND-PROD-TestCoverage

- **Description:** Monitoring/evaluating the test coverage in order to understand how many of the products have been tested.
- **Point of view**: Project manager and person responsible for software development.
- **Context:** Software development projects.
- **Questions**:
  - Q1: What is the test coverage for each module of the code?
  - Q2: Are you satisfied with the coverage? Does any module have an especially low coverage? Is it necessary to take any corrective action?
- **Inputs:**
  - BC: Branch coverage (% of branches covered): This value is obtained automatically. It measures the branches or decisions which are executed by the tests and by module.
  - LC: Lines of code (% of LOCcovered): This value is also obtained automatically and it means the number of lines of code which have been executed by the test and by module.
- **Algorithms**:

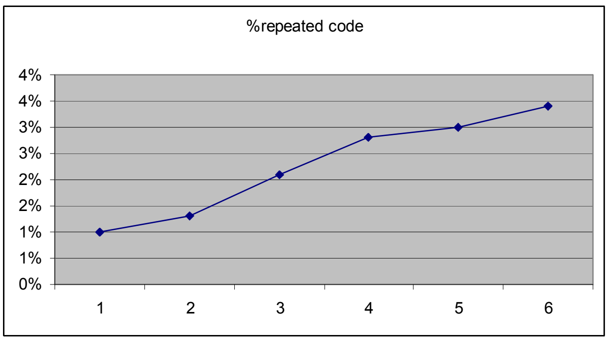
  P1. BC + LC /2 *100.
- **Improvement of the indicator**
  - An improvement of the indicator would be to understand the normal test coverage that previous similar products developed in the company had and therefore be able to perform evaluations.
  - The recommended value is 95-100% and the minimum is 80% for unit tests.
  - The recommended value is 80% for integration tests.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically at the end of the coding phase and at least once at the end of it.
  - **Indicator results report location:** Measurements and improvement suggestions report and/or project follow-up report.
  - **Integration:** This indicator should be analysed for the development process during the unit and integration test phases.
  - **Resources required**: Static quality analyser tools.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand the actual reliability or maintainability of the product.
- **Graphic Display** It should show the progress of the unit testing coverage.



Product Maintainability

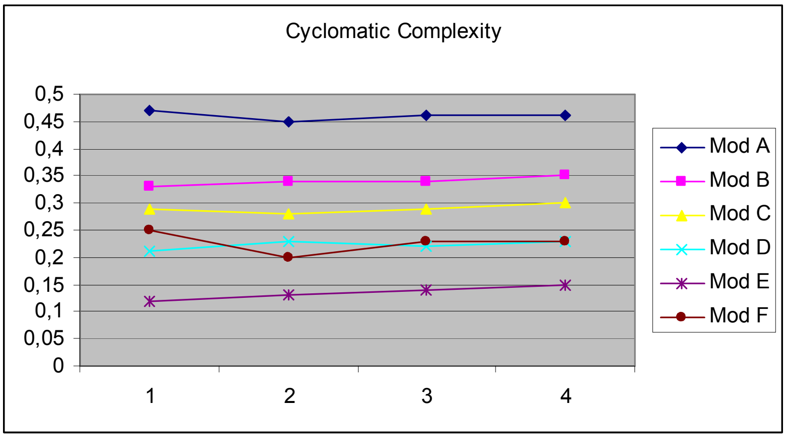
IND-PROD-RepeatedCode

- **Description:** Monitoring/evaluating the repetition of code in order to take corrective actions.
- **Point of view**: Project manager and person responsible for software development and maintenance.
- **Context:** Software development and maintenance projects.
- **Questions**:
  - Q1: What is the progress of the repeated code?
  - Q2: Is it necessary to take any corrective action?
- **Inputs:**
  - RC: % of repeated code.
- **Improvement of the indicator**
  - An improvement of the indicator would be to understand the normal ranges of repeated code of the product and be able to perform evaluations.
  - The recommended value is $<$ 3% * NCSS (Number of non commented code sentences) and the maximum value may be 5% * NCSS.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development and maintenance
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically and depending on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the development and maintenance process and especially during the coding activity.
  - **Resources required:** Static quality analyser tools.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product being developed is of sufficient quality to be easily maintained. High repeated code makes the code difficult to maintain since one defect may be fixed in two places. It also makes the code difficult to understand.
- **Graphical Display.** This should show at a glance the progress of % of repeated code (different lines could be shown to represent different modules).



IND-PROD-CiclomaticComplexity

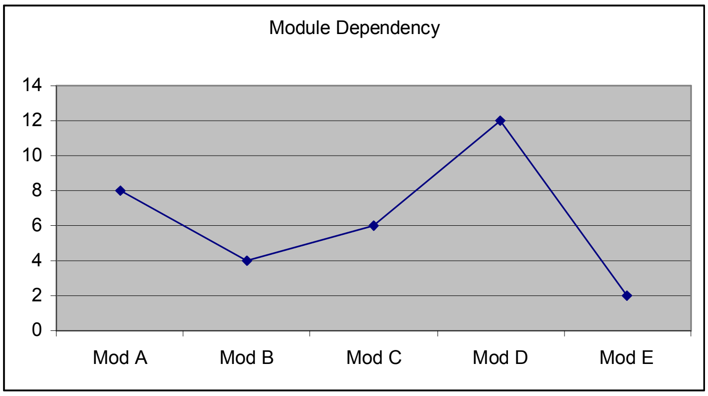
- **Description:** Monitoring/evaluating the Cyclomatic complexity of the modules in order to take corrective actions.
- **Point of view**: Project manager and person responsible for software development.
- **Context:** Software development and maintenance projects.
- **Questions**:
  - Q1: What is the cyclomatic complexity of the modules (classes) being developed?
  - Q2: Is it necessary to take any corrective action?
- **Inputs:**
  - C1: cyclomatic complexity of the module/class .
- **Improvement of the indicator**
  - An improvement of the indicator would be to understand the normal ranges of the cyclomatic complexity of the product's modules and be able to perform evaluations.
  - The recommended range is < 10 and the maximum is 15.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development and maintenance.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically and depending on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the development and maintenance process and especially during the coding activity.
  - **Resources required:** Static quality analyser tools.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product being developed is of sufficient quality to be easily maintained. High cyclomatic complexity makes the module difficult to understand and test.
- **Graphical Display.** This should show at a glance the progress of the cyclomatic complexity of each module/class developed.



IND-PROD-ModuleDependency

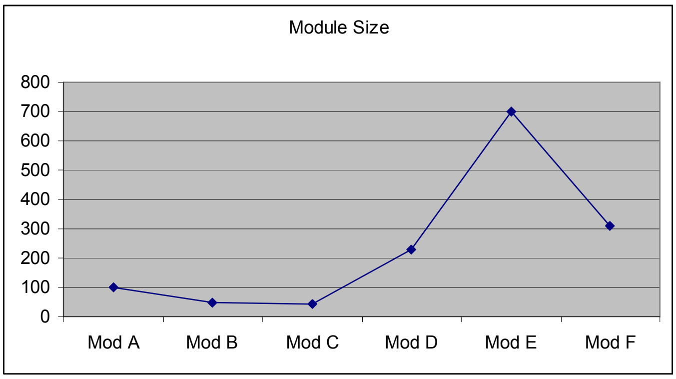
- **Description:** Monitoring the dependency of the modules developed in order to take corrective action.
- **Point of view**: Project manager and person responsible for software development.
- **Context:** Software development and maintenance projects.
- **Questions:**
  - **Q1:** What is the dependency of the modules being developed?
  - **Q2:** Is it necessary to take any corrective action?
- **Inputs:**
  - **D1:** Dependency of the module/ class (this information should be automatically provided by the software development tool or software configuration management)
- **Algorithms:**

  P1. D1 (for each module or class)
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development and maintenance.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically and depending on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the development and maintenance process and especially during the coding activity.
  - **Resources required:** Static quality analyser tools.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product being developed is of sufficient quality to be easily maintained. High dependency makes the code difficulty to compile and maintained since one change requires to check and maybe to change the dependent modules.
- **Graphical Display.** This should show at a glance the dependency of each module/class developed.



IND-PROD-ModuleSize

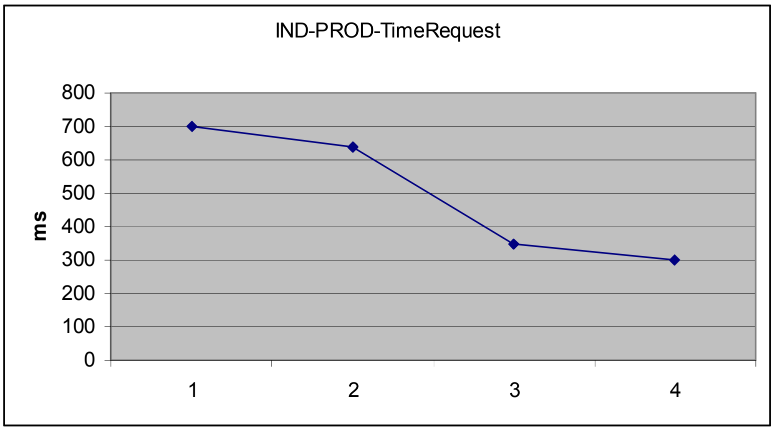
- **Description:** Monitoring/evaluating the size of the modules developed in order to understand the maintainability of the module developed.
- **Point of view:** Project manager and person responsible for software development.
- **Context:** Software development and maintenance projects.
- **Questions:**
  - **Q1:** What is the size of the modules being developed?
  - **Q2:** Is it necessary to take any corrective action?
- **Inputs:**
  - **S1:** Size of the module/class .
- **Improvement of the indicator**
  - An improvement of the indicator would be to understand the normal ranges of the size of the product's modules, taking into account the historical data and the type of product .
  - The recommended value is 35 LOC/ method and < 300 LOC/class.
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for development and maintenance
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** Periodically and depending on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the development and maintenance process and especially during the coding activity.
  - **Resources required:** Static quality analyser tools.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product being developed is of sufficient quality to be easily maintained. High size in modules makes the code less understandable.
- **Graphical Display.** This should show at a glance the size of each module/class developed. If the normal ranges are known, these should be indicated in the graph.



Product Efficiency

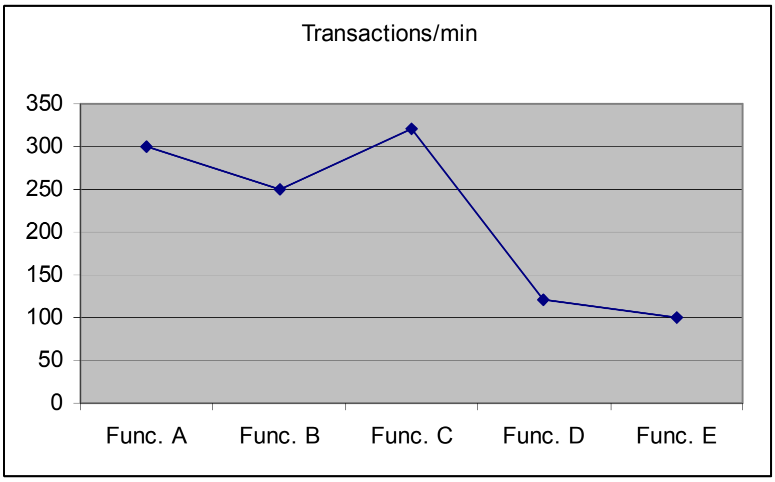
IND-PROD-TimeRequest

- **Description:** Monitoring the time it takes to perform the critical request to the system and that most used in order to understand the efficiency of the product.
- **Point of view**: Project manager and person responsible for software development and maintenance.
- **Context:** Software development and maintenance projects.
- **Questions**:
  - Q1: For each of the most critical and most used requests to the system, how long does it take to obtain the result of a request from the entry command until the user receives the result?
  - Q2: Is it necessary to take any corrective action?
- **Inputs:**
  - T1: Time taken to obtain result - time the entry command finishes.
- **Measuring activity information:**
  - **Person responsible for analysis and feedback:** Those responsible for development and maintenance.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** At least once during the integration or system tests.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the development and maintenance process and especially during the integration or system tests.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product being developed is of sufficient efficiency for the critical and most-used requests.
- **Graphical Display.** This should show at a glance the most used and most critical requests and the time that they take.



IND-PROD-TrancThroughput

- **Description:** Monitoring the throughput of the critical functions of the system in order to understand the efficiency of the product and improve it if necessary.
- **Point of view**: Project manager and person responsible for software development.
- **Context:** Software development and maintenance projects.
- **Questions**:
  - Q1: For each of the required and most critical functions of the system, what is the maximum number of transactions that can be completed per unit of time?
  - Q2: Is it necessary to take any corrective action?
- **Inputs:**
  - T1: Number of transactions completed/ unit of time (e.g. minute).
- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Those responsible for development and maintenance.
  - **Destination of the analyses and interpretation results, audience:** Project manager.
  - **Analysis and feedback frequency:** At least once during the integration or system tests.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the development and maintenance process and especially during the the integration or system tests.
  - **Resources required:** simulation tools may be required and an environment similar to that used in production.
- **Analysis:** This information will be useful for the project manger and the person responsible for the development process in order to understand whether the product being developed is of sufficient efficiency for the critical and most-used requests.
- **Graphical Display.** This should show at a glance the most critical functions that require this analysis and their transaction throughput.



Software Maintenance Process

Maintenance planning

IND-PRJ-MeanEffortDev

- **Description:** Monitoring/Evaluating the accuracy of the effort estimated for each maintenance action in order to improve cost estimations.
- **Point of view**: Person responsible for specific maintenance project.
- **Context:** Software maintenance.
- **Questions**:
  - Q1: In the period of time analysed, what was the deviation mean in terms of effort between the effort plan to develop the fixing defect request and the real effort made? Only non-urgent requests should be planned.
  - Q2: In the period of time analysed, what was the deviation mean in terms of effort between the effort plan to develop the request for changes and the real effort made.
  - Q3: Is it necessary to take any corrective action or make any improvements to the process?
- **Inputs:**
  - E1: Planned effort (the planned effort is assigned when the maintenance request is first reviewed and classified).
  - E2: Actual effort.
- **Algorithms**:
  - Q1. SUM[(E2 - E1)/ E1]n / #n where "n" is the defect fixing request that was not urgent and thus planned in the analysed period of time
  - Q2: SUM[(E2 - E1)/ E1]m / mn where "m" is the request for change in the analysed period of time.

  For these calculations it may be that the extreme values of (E2-E1), if these are not usual, cannot be taken into account.
- **Improvement of the indicator:**

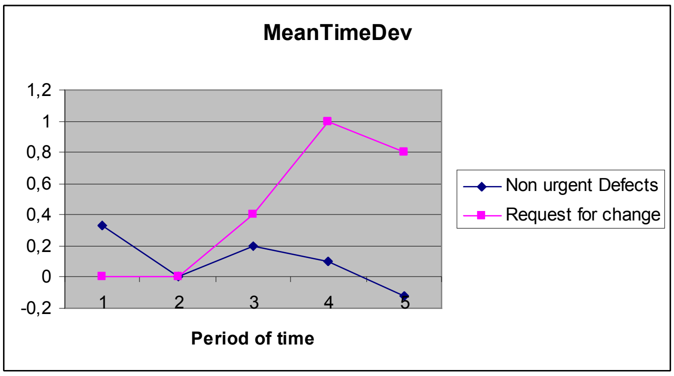
Once the company has experience with the above analyses, normal ranges of effort deviation can be discovered and reliable evaluation results can be provided

- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for specific maintenance project
  - **Destination of the analyses and interpretation results, audience:** person responsible for process management, person responsible for portfolio project management, headquarters.

- **Analysis and feedback frequency:** Periodically and depending on the needs.
- **Indicator results report location:** Measurements and improvement suggestions report, and qualitative and quantitative report.
- **Integration:** This indicator should be analysed for the maintenance process, for the follow-up activity, and during the portfolio project management process.
- **Resources required:** Incident management tool.

➢ **Analysis:** This information would be useful to improve effort estimation for the maintenance process and monitor the maintenance service.

➢ **Graphic Display:** This should show the mean effort deviation of the fixing of defect requests and the mean time effort deviation of request for change requests. It will show the evolution of the indicator.



IND-PRJ-MeanEffort

➢ **Description:** Monitoring/Evaluating the mean effort in performing maintenance actions in order to improve cost estimations and the service provided.

➢ **Point of view**: Person responsible for specific maintenance project.

➢ **Context:** Software maintenance.

➢ **Questions**:

- Q1: In the period of time analysed, what was the mean effort for fixing non urgent defect requests?
- Q2. In the period of time analysed, what was the mean effort for completing request for change?
- Q3. In the period of time analysed, what was the mean effort for fixing urgent defects?
- Q4: Is it necessary to take any corrective action or make any improvements to the process?

➢ **Inputs:**

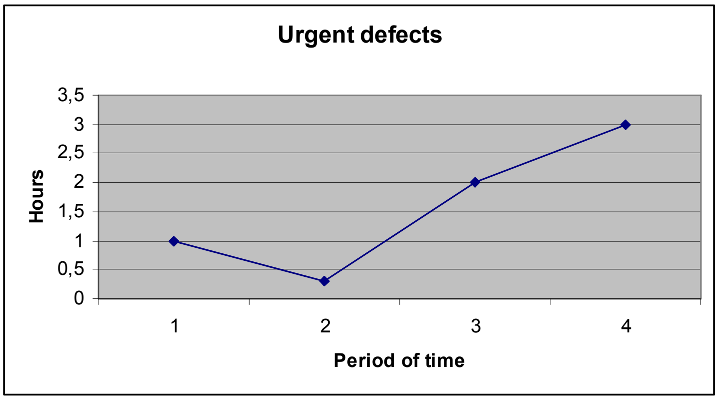
- AE: Actual effort from when the “client” registered the problem until the problem was solved.

Urgent effort can be measured in hours and request for change and non urgent defects can be measured in days.

- **Algorithms**:
  - Q1. SUM[AE]n / #n where "n" is the non urgent defects fixing request that was completed in the analysed period of time
  - Q2. SUM[AE]m / #m where "m" is the request for changes that were completed in the analysed period of time
  - Q3. SUM[AE]r / #n where "r" is the urgent defects fixing request that was completed in the analysed period of time
- **Improvement of the indicator:**

Once the company has experience with the above analyses, normal ranges of effort can be discovered and reliable evaluation results can be provided.

- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for specific maintenance project.
  - **Destination of the analyses and interpretation results, audience:** person responsible for process management, person responsible for portfolio project management, business manager.
  - **Analysis and feedback frequency:** Periodically and depending on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report, and qualitative and quantitative report.
  - **Integration:** This indicator should be analysed for the maintenance process, for the follow-up activity, and during the portfolio project management process.
  - **Resources required:** Incident management tool.
- **Analysis:** This information would be useful to improve effort estimation for the maintenance process, and identify problems in maintenance productivity, and to take corrective actions and control the maintenance service.
- **Graphic Display:** This graph should show the evolution of the mean effort to resolve each of the types of maintenance requests (urgent defect, non urgent defects fixing request and request for change).



Maintenance Efficiency

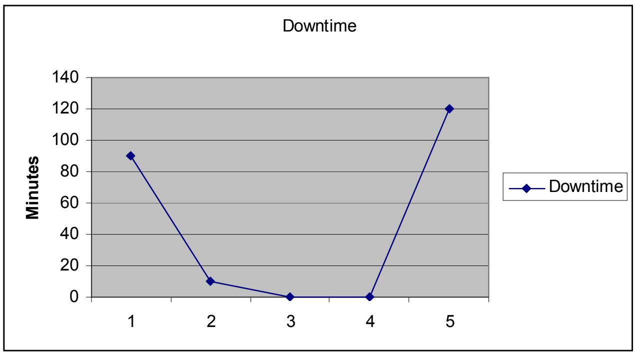
IND-PRJ-MeanDownTime

- **Description:** Monitoring/Evaluating mean maintenance downtime in order to take corrective actions.
- **Point of view**: Person responsible for specific maintenance project.
- **Context:** Software maintenance.
- **Questions**:
  - Q1: In the period of time analysed, what was the mean of service downtime for the product analysed?
  - Q2: Is it necessary to take any corrective action or make any improvements to the process?
- **Inputs:**
  - DT: downtime of each of the services provided by the product/system analysed. This can be measured in minutes or hours, depending on the type of system and service.
- **Algorithms**:
  - Q1. SUM [(DT)t]s where "s" is each service provided by the product/system analysed and t are the different moments at which there has been a service down time.
- **Improvement of the indicator:**

Once the company has experience with the above analyses, normal ranges of mean downtime can be discovered and reliable evaluation results can be provided.

In addition, this indicator can be classified by services, therefore showing the data for each service.

- **Measuring activity information:**
  - **Those responsible for analysis and feedback:** Person responsible for specific maintenance project.
  - **Destination of the analyses and interpretation results, audience:** person responsible for process management, person responsible for portfolio project management
  - **Analysis and feedback frequency:** Periodically and depending on the needs.
  - **Indicator results report location:** Measurements and improvement suggestions report.
  - **Integration:** This indicator should be analysed for the maintenance process, for the follow-up activity, and during the portfolio project management process.
  - **Resources required:** Incident management tool.
- **Analysis:** This information would be useful to improve the efficiency of the process.
- **Graphic Display:** This will show the evolution of the total of mean downtime per period analysed



D. MIS-PyME: Measurement Process

Name	MIS-PyME measurement process
Purpose	To carry out the measurement program defined in the company and evaluate its usefulness and suitability
Outcomes	1. measure results are calculated based on measure specifications 2. indicators are built and analysed. 3. decisions and improvements are identified based on the analyses and interpretations 4. decisions and improvements are communicated 5. The benefits, usefulness of and possible improvements to the measurement program are analysed.
Base Practices	1. Collecting and Processing the data. The data required to generate the measure results is collected. The indicator is built based on the measure results 2. Analysing and Interpreting the data: The indicator base data are analysed and interpreted, questions are answered and decisions are therefore made according to the indicator goal 3. Communicating the results: Analyses and interpretation results are communicated, reviewed and updated if necessary
Management Practice	- Evaluating the measurement program. The measurement process is evaluated and modified in order to improve it, and it is verified whether past action plans were carried out in order to understand the benefits of the measurement process. Lessons learned are also identified and included in a report for the knowledge management process
Inputs	Standard measurement process specification
Output	Standard measurement process specification (updated) The measure results Indicator.data Indicator.analyses and interpretation results Decisions and Improvement Suggestions Feedback (performed) Lessons learned report

Activity 1: Prepare the measurement process:	
Task 1.1.Verifying and adapting the measurement process to be applied: The roles and measure data collection procedure defined for the standard measurement process of the organization, are adapted if the environment requires it. If any of the indicators appear not to be useful, and there is not any constrain, these indicators can be ignored.	
Role: Person Responsible for Measurement	**Input**: Standard measurement process specification **Output**: Standard measurement process specification (updated)
Activity 2. Collecting and Processing the data: The data required to generate the measure results is collected. The indicator is built based on the measure results.	

Task 2.1. Collecting the data: The data required to generate the measure results is collected and processed, if any operation over the data is required in order to obtain the measure result.	
Role: Measurement Data Collector	**Input:** Measure specifications. **Output:** The Measure results.
Task 2.2. Building the indicators: The measure results required to build the indicator are verified and processed as specified in the indicator algorithm. Graphic displays are created if necessary, thus the indicator is prepared to be analysed and interpreted.	
Role: The person responsible for measurement	**Input:** The measure results, Indicator specifications. **Output:** Indicator.data
Activity 3. Analysing and Interpreting the data: The indicator base data are analysed and interpreted, questions are answered and decisions are therefore made according to the indicator goal.	
Task 3.1 Analysing and Interpreting the data: The indicator base data are analysed and interpreted, questions are answered.	
Role: Person responsible for Measurement	**Input:** Indicator.data **Output:** Indicator.analyses and interpretation results
Task 3.1 Determining suggestions for improvement and making decisions: Based on the analysis and interpretation results, suggestions for improvement and decisions are made according to the indicator goal.	
Role: Person responsible for Measurement	**Input:** Indicator.analyses and interpretation results **Output:** Decisions and Improvement Suggestions
Task 4. Communicating the results: Analyses and interpretation results are communicated, reviewed and updated if necessary	
Task 4.1. Communicating the results and suggestions for improvement: The results of the analyses and the interpretations, the suggestions for improvement and/or the decisions made are communicated to the interested parties.	
Role: Person Responsible for Measurement	**Input:** Indicator.analyses and interpretation results, Decisions and Improvement Suggestions **Output:** feedback (performed)
Task 4.1. Reviewing and updating the measurement results and suggestions for improvement if necessary: The results of the analyses and the interpretations, the suggestions for improvement and/or the decisions made are reviewed and modified if necessary.	
Role: Person Responsible for Measurement	**Input:** Indicator.analyses and interpretation results, Decisions and Improvement Suggestions **Output:** Indicator.analyses and interpretation results (updated), Decisions and Improvement Suggestions (updated).
Task 5. Evaluating the measurement program	
Task 5. Evaluating the measurement program. The measurement process is evaluated and modified in order to improve it, and it is verified whether past action plans were carried out in order to understand the benefits of the measurement process. Lessons learned are also identified and included in a report for the knowledge management process.	
Role: Measurement Analyst, Person Responsible for Measurement and Measurement Data Collector	**Input:** measurement process specification **Output:** measurement process specification (updated), lessons learned report.

Table D-1. MIS-PyME Measurement Process

E. Measurement Assessment Process and Questionandire

MIS-PyME 3M Assessment Process

The roles that take part in the assessment are the following:

- The sponsor of the assessment, whose responsibility is to spread the goal of the assessment and formalize the assessment, ensure that the person who is going to carry out the assessment is a competent assessor, build the assessment agreement, provide the required human and material resources, resolve the conflicts that may appear during the assessment, understand the conformity with the assessment process and results, and evaluate the assessment process.
- The assessor is the person who will carry out the assessment process. S/he should understand the measurement assessment model, the MIS-PyME measurement capability maturity model and the measurement process. S/he will be responsible of the execution of the process: planning, performing the assessment, determining the results and closing the assessment. In addition s/he has to report the results and the conformity of the assessment performed with the reference assessment process. Since usually there are costs and resources limitations, and this assessment should no be understood as a formal assessment for a formal certification, this person may be from inside the assessed company.
- The assessed people are responsible to answer to the questions and give the assessor the required information. The assessed people should be the main involved in the measurement process: the people who define and maintain the measurement process, the main interested in the measurement process results, people involved in the data collection, etc.

The description of the assessment process is as follows

Purpose	To determine the measurement capability of the company as regards the measurement process in order to identify improvement measurement issues that will benefit software process improvement and business goals, and identify which are the measurement goals that the company is able to successfully implement
Goals	- O1: Achieving a formal evaluation process by means of asking to the suitable people, understanding and checking the inputs, verifying the results and answers, etc. - O2: Determining a reliable measurement capability maturity level of the company. - O3: Determining the set of measurement goals suitable to be implemented in the company. - O4: Specifying the set of the most prior process improvement goals.
Indicators	- The assessment agreement and the evaluation plan exist and it is formally signed. - The questions asked comply with the ones specified in MIS-PyME measurement maturity model. - There are evidences for each answer in order to proof them. - There is a formal document where the result of the assessment is described. - The results of the assessment were reported to the sponsor and other interested in the assessment.
Inputs	- Organization information ▪ Name of the organization or the unit to be assessed ▪ The name of the competent assessor ▪ The expected measurement capability maturity level thus the highest measurement capability maturity model to be investigated. - Needs of the organization and cause of the measurement assessment - The measurement information: ▪ The standard measurement process definition of the organization if exist ▪ The list of projects performed in the organization during the last two years. ▪ The information of each project as follows: the project and quality plan (and its various versions), the tracking project reports, the close of project reports, etc. ▪ The measurement reports performed as a result of a measurement practice. - The process improvement plan. - MIS-PyME Measurement Capability Maturity Model Assessment questionnaire. - MIS-PyME assessment model standard process specification
Outputs	- The assessment Agreement: o Organization information (indicated above). o The owner of the assessment results and the restrictions of their

	use. o controls on information resulting from a confidentiality agreement o The units of the company to be assessed o MIS-PyME 3M version o The expected date to start and to finish the assessment. o Any restriction such as the availability of the resources. - The assessment plan: o Measurement information plan: The specification of the measurement information to evaluate: projects to be evaluated, measurement reports to be evaluated, etc. o The assessment participants names and roles o The calendar of the activities - The set of questions and answers of the MIS-PyME 3M questionnaire (standard questionnaire). - The assessment report: o Organization information (see above) o The date of the assessment o the set of profiles rating o the maturity capability level of the organization o a suggestion of the measurement goals that the company is ready to implement according to its maturity. o The set of issues to improve which will be the input to the measurement improvement process. o Lessons learned and possible improvements of the assessment process

<table>
<tr><td colspan="2">Step1. Prepare the assessment: The aim of this activity is to define the assessment agreement</td></tr>
<tr><td>Role: Sponsor</td><td>Input: Needs of the organization and cause of the measurement assessment, organization information.
Output: Assessment agreement</td></tr>
<tr><td colspan="2">Step 1.1: Determine the scope of the assessment, the start and end dates and any restriction of the assessment. In this step it should be specified the units of the organization to be assessed as regards the purpose of the assessment (ej. the whole organization, only one unit of the organization, etc). The start and end dates should be determined as regards the needs of the assessment and any other restriction as regards the assessment such as the restriction of the resources: material to perform the assessment, holidays of key people, etc.</td></tr>
<tr><td>Role: Sponsor</td><td>Input: Needs of the organization and cause of the measurement assessment
Output: Scope of the assessment, start and end dates, assessment restrictions</td></tr>
<tr><td colspan="2">Step 1.2 Determine who will perform the assessment. The selected person should understand the measurement assessment model, or should be able to study it. In addition s/he should have experience in measurement activities and should understand the company culture and organization. S/he should know who are the best people to answer the assessment questions and should be capable to determine the answers of the measurement assessment questionnaire</td></tr>
<tr><td>Role: Sponsor</td><td>Input: Needs of the organization and cause of the measurement assessment, scope of the assessment, start and end dates, assessment restrictions
Output: Assessor assignment</td></tr>
<tr><td colspan="2">Step 1.3. Specify the confidentiality agreement. Specify the owner of the assessment results and the restrictions of their use and the controls that will be applied to the assessment information based on the confidentiality agreement.</td></tr>
<tr><td>Role: Sponsor</td><td>Input: Needs of the organization and cause of the measurement assessment.
Output: Confidentiality agreement</td></tr>
<tr><td colspan="2">Step 2: Planning the assessment. The assessment is planed as regards the measurement information plan, the measurement participants names and roles and the calendar of activities</td></tr>
<tr><td>Role: Assessor</td><td>Input: Needs of the organization and cause of the measurement assessment, assessment agreement.
Output: The assessment plan:</td></tr>
<tr><td colspan="2">Step 2.1. Select the source of information for the assessment. A study of the organization should be performed in order to determine whether it exist a standard measurement process definition of the organization; if it exists it will be used as an input of the assessment. It should be studied what are the projects performed in the last two years, which will be the projects selected for the assessment, what kind of information of the project will be selected for the assessment, what other measurement activities performed in the company or in the unit should be assessed and what kind of information can be used for the assessment. In addition the set of tools used for the development, maintenance and management processes are listed.</td></tr>
<tr><td>Role: Assessor</td><td>Input: Needs of the organization and cause of the measurement assessment, assessment agreement
Output: The measurement information plan</td></tr>
<tr><td colspan="2">Step 2.2. Specifying the assessment participants. As regards the selected information which will be used as the source of the assessment, the people who will be asked the questions of the assessment are selected.</td></tr>
<tr><td>Role: Assessor</td><td>Input: The measurement information plan
Output: The assessment participants names and roles in the company</td></tr>
<tr><td colspan="2">Step 2.3. Specifying the calendar and activities to perform: A calendar for activities performed for the assessment should be planned</td></tr>
</table>

<table>
<tr><td>Role: Assessor</td><td>Input: MIS-PyME assessment model standard process specification, the measurement agreement, the measurement information plan.
Output: Assessment calendar</td></tr>
<tr><td colspan="2">Step 2.4. Brief Lecture: The assessor and the sponsor will explain in a brief meeting the aim of the assessment, the method used, and the calendar. The people involved in the assessment may indicate any restriction, comment or disagreement to the plan.</td></tr>
<tr><td>Role: assessor</td><td>Input The assessment plan
Output. The assessment plan (updated)</td></tr>
<tr><td colspan="2">Step 2.5. Acceptance of the assessment plan. The assessment plan is reviewed and accepted</td></tr>
<tr><td>Role: Sponsor</td><td>Input: Assessment plan
Output: Assessment plan (accepted)</td></tr>
<tr><td colspan="2">Step 3. Performing the assessment. The questions to be asked are tailored and organized in order to meet the required people to answer these questions. In addition the assessor analyse the questions and answer to the standard questionnaire provided by MIS-PyME</td></tr>
<tr><td>Role: Assessor</td><td>Input: Assessment plan, Assessment agreement, MIS-PyME 3M questionnaire.
Output: Tailored questionnaire (filled out); MIS-PyME 3M questionnaire(filled out).</td></tr>
<tr><td colspan="2">3.1. Adjust the questionnaire. The assessor will adjust the questionnaire if required based on the projects assessed, the measurement information assessed, the culture of the company, the level they estimate that the company will achieve (it is not required to adjust the whole standard questionnaire), etc. He will also understand to whom ask each question.</td></tr>
<tr><td>Role: Assessor</td><td>Input: Assessment plan, Assessment agreement, MIS-PyME 3M questionnaire.
Output: Tailored questionnaire;</td></tr>
<tr><td colspan="2">3.2 Deliver the questionnaire. The assessor will ask the questions to the people required and keeping the evidence of the answer. Afterwards the assessor will confirm to that person the answer and the evidence.</td></tr>
<tr><td>Role: Assessor</td><td>Input: Tailored questionnaire.
Output: Tailored questionnaire (filled out).</td></tr>
<tr><td colspan="2">3.3 Consolidate the questionnaire. The assessor will analyse all the answers to the questions and s/he will fill out the standard questionnaire of MIS-PyME measurement maturity capability model. S/He will ask the required people in case of doubts.
If the question can not be answered, since it is not applicable in the context, this question is not taken into account.</td></tr>
<tr><td>Role: Assessor</td><td>Input: Tailored questionnaire filled out.
Output: MIS-PyME 3M questionnaire(filled out).</td></tr>
<tr><td colspan="2">4 Determining the results of the assessment. In this step the assessor determines whether the measurement attributes are fulfilled and what the measurement maturity of the company is. In addition s/he will determine the most important process improvement issues.</td></tr>
<tr><td>Role: Assessor</td><td>Input: MIS-PyME 3M questionnaire(filled out).
Output: measurement company profile, measurement maturity level of the company , set of improvement issues</td></tr>
<tr><td colspan="2">4.1 Final result of the questionnaire. The assessor will rate each of the attributes indicated in MIS-PyME 3M based on the assessment indicators specified in MIS-PyME 3M, and check if the company complies with the required attributes to achieve the expected level.</td></tr>
<tr><td>Role: Assessor</td><td>Input: MIS-PyME 3M questionnaire(filled out).
Output: measurement company profile, measurement maturity level of the company</td></tr>
<tr><td colspan="2">4.2 Determining the measurement improvement areas: The assessor will determine which are the</td></tr>
</table>

<table>
<tr><td colspan="2">most important and feasible measurement issues that the company should improve.</td></tr>
<tr><td>Role: Assessor</td><td>Input: MIS-PyME 3M questionnaire(filled out), measurement company profile
Output: Set of improvement issues</td></tr>
<tr><td colspan="2">5 Close of the assessment. At this state the final report of the assessment is described, the results are explained to the sponsor and other involved people and the assessment material is delivered to the sponsor.</td></tr>
<tr><td>Role: Assessor</td><td>Input: measurement company profile, set of improvement issues, MIS-PyME 3M questionnaire(filled out)
Output: the assessment report, the MIS-PyME 3M questionnaire(filled out with no compromised information), reporting meeting performed.</td></tr>
<tr><td colspan="2">5.1 Specifying the assessment report. The final assessment report is specified which includes the following:
- Organization information:
o Name of the organization or the unit to be assessed
o The identity of the sponsor of the assessment and the sponsor's relationship to the organizational unit being assessed.
o The name of the competent assessor
o The expected measurement capability maturity level.
- The date of the assessment.
- The set of profiles of the measurement process based on the attributes rating.
- Tthe measurement maturity capability level of the organization.
- A suggestion of the measurement goals that the company is ready to implement according to its maturity (see MIS-PyME 3M).
- The set of issues to improve which will be the input to the measurement improvement process.
- Lessons learned and possible improvements of the assessment process</td></tr>
<tr><td>Role: Assessor</td><td>Input: Measurement company profile, set of improvement issues, MIS-PyME 3M questionnaire(filled out)
Output: The assessment report.</td></tr>
<tr><td colspan="2">5.2 Reporting: The assessor will call up a meeting in order to explain the sponsor and the people involved in the assessment the results of the assessment, the use of the results and the main process improvement issues</td></tr>
<tr><td>Role: Assessor</td><td>Input: Measurement company profile, set of improvement issues, measurement maturity of the company
Output: Reporting meeting performed.</td></tr>
<tr><td colspan="2">5.3 Delivery of the assessment information: The assessor will delete all the information that addresses any person or project from the assessment questionnaire and the evidences of the answers, and s/he will give the sponsor these questionnaires and the assessment report.</td></tr>
<tr><td>Role: Assessor</td><td>Input: The assessment report, MIS-PyME-3M questionnaire (filled out)
Output: The assessment report, MIS-PyME-3M questionnaire (filled out and with no compromised information).</td></tr>
</table>

Table E-1. MIS-PyME 3M Assessment Process

MIS-PyME 3M Questionnaire

Level 1: Performed Process.
PA 1.1 (P) Process performance attribute.
a) 1. Are there any rules in order to collect the measures or any protocol defined? Is it determined when the measure should be collected? Or do data collectors understand more-less in the same way how to collect the data? 2. Are the indicators built using the results of the measures? Are these indicators properly analysed by the right people, those people who are able to perform correct analysis on the data? 3. Are measurement results clearly communicated to the interested people? Are these results discussed and updated if required? 4. Is the measurement program evaluated in order to identify defects or suggest improvements?
PA 1.2 (I) Basic project and product focus performance attribute
a) At least in largest or more important projects in the company, does the project manager tracks the schedule and identifies deviation against planned? b) Does the organization tracks the failures of each of the critical product implanted in production?
PA 1.3 (R): Basic management tools implemented attribute
a) Is there an incident tool implemented in the organization in order to keep the identified failures detected in production, and the defect detected during test activities? b) Has the organization established any tool in order to track schedule progress of the projects?
Level 2: Managed process
PA 2.1 (P) Performance management attribute.
a) Are the measurement goals clearly determined in the measurement program, or in the project or quality plan? b) Does the quality, development or management processes applied in the project take clearly into account the measurement activities? Are these activities planned with regard the time when they should be performed and the effort that it takes? Are these measurement activities checked in order to survey that they have been correctly performed? c) Are measurement activities adjusted to meet project, quality and measurement plans? d) At the beginning of the project ¿Are the measurement responsibilities such as data collection, analyses preparation, analysis and interpretation, feedback clearly assigned? And even out of projects, Are the measurement responsibilities such as data collection, analyses preparation, analysis and interpretation, feedback clearly assigned? e) Do people involved in the measurement process understand their responsibilities and the responsibilities of the others? Is communication between these people fluent?
PA 2.2 (P) Workproduct management attribute
a) Are work products which should be obtained from the measurement activities determined? b) Is the way by which work products (measure results, data collected, analysis results, decisions taken, improvement suggestions, etc.) will be stored, and documented determined? Are rules to verify the measurement results specified? c) Are measure results appropriately obtained and documented? Are these work products communicated to the interested people? Are work products effectively managed to bring benefits such as taking corrective actions in projects using measurement results, or improving

future estimations and plans, etc.?

d) Are analyses results and interpretation, and decisions and the action plan communicated, discussed and adjusted if required? Are analyses results and interpretation, and decisions and the action plan communicated to the people interested in the measurement process?

PA 2.3 (I) Basic project and product focus management attribute

a) Are the specific and defined measures in order to control the deviations against plans in terms of cost, duration and effort phase-by-phase and globally at the end of the project?

b) Is the reliability of the developed product being tracked by checking the defects and problems reports observed by the testing team or the client? Does the company perform any action when the reliability does not seem too good? Does the company take into account the reliability observed on the product at the testing phase in order to decide whether the product can be delivered or not?

c) At least when the product is being delivered to the client. Are there specific indicators or measures in order to understand the customer satisfaction?

PA 2.4 (R): management and development tools implemented attribute

a. Has the organization established any tool in order to track defects during development projects?

b. Has the organization established any tool in order to track cost and effort of the projects?

Level 3: Established Process

PA 3.1(P) Process definition attribute

a) Is there a standard measurement process which is adapted when it is applied in real projects or in other measurement cases? Are there guidelines in order to ease the adaptation and explains which parts of the process can not be modified and which parts can be modified, guidelines as regards the size and scope of projects, etc.?

b) Is the measurement process completely incorporated it in the standard software quality, development and management processes? Do the report templates of these processes include measurement information as required? Such as project tracking reports, close project reports, quality reviews reports, etc. Are the data collection, analyses and feedback activities included into these processes?

c) Are the needed competencies and roles for performing a process identified as part of the standard process?

d) Does the measurement process specifies the tools and infrastructure required in order to collect data, keep measurement results and analyses results information etc.?

e) Are there any standard methods in order to check that measurement activities are correctly performed and the measures, analyses and interpretation results are correctly obtained?

PA 3.2 (P) Process deployment attribute

a) Are the measurement processes applied according to standard process?

b) When performing the measurement process, are the required roles assigned as defined in the standard process?

c) Do people understand the goals of the measurement process, their responsibilities, the utility of the measurement tasks? Are people trained in order to perform their responsibilities?

d) Are there actions plans applied in order to improve organization processes, such as project management process? Is the measurement process assessed in order to improve it? Are measurement results of projects shared across projects?

PA 3.3 (I) Advanced project tracking attribute

a) Does the company manage the progress of the project by understanding the number of requirements or use cases, developed and/or tested?

b) Are there cross-project analyses about their results in terms of cost, effort, duration deviation, etc.?

c)	Does your company performs any re-planning and understands the effects of the requirements changes?
PA 3.4 (I) Advanced product tracking attribute	
a)	Does the company perform static analyses on the product in order to understand its quality in terms of maintainability such as inheritance of classes, dependence between classes or modules, cyclomatic complexity, repeated code, etc.? When these results considerably exceed the recommended thresholds by the standards, or by the company thresholds, does the company perform any action?
b)	Does the company measure and use the same product related indicators (quality indicators) for all the products maintained and developed in the organization? Is then the company able to perform cross-product analyses and does the company detect common cusses of problems related to the product and, does the company proposes improvement organization initiatives in order to avoid these problems?
PA 3.5 (I) Process tracking attribute	
a)	Is the effectiveness of the test phases measured? Are data as regard with productivity collected? In terms of the time required completing the analysis, construction and the test of a Use Case?
b)	Are normal productivity ranges known?
c)	Does the company measure that the activities and tasks of the process defined have been well followed in the project? Does the company measure if the output products have been created as required? Does the company measure if the roles required in the process have been well identified in the project?
d)	Does the company track the time it takes to fix a failure in production? are normal ranges in terms of the time for resolving failures in production understood?
e)	Does the company measure the downtime of the service? Does the company understand common ranges for the downtime of a service due to reparation, maintenance actions or failure? Does the company perform any action when the maintenance action is not in this range?
PA .3.6 (R) Resources deployment attribute	
a)	Is there any organizational database in order to store the results of the measures?
b)	Is there a life cycle configuration management tool for the requirement and code configuration management?
c)	Are Training Program performed as needed for the people involved in measurement to correctly perform their roles?
d)	Does the company use any tool to perform the quality analyses?
e)	Does the company measure the mean time than the provider takes to solve a problem? Does the company measure the deviation in providing the tool or the component?
f)	Do the project managers understand the existent estimation mechanisms and are they trained on this? COCOMO II, Function Point, Use Case Point, etc.
g)	Are there protocols which guide people using the tools and make them introduce the information in the same way (e.g in the incident tool or in the measurement database)?

Table E-2. MIS-PyME 3M Questionnaire

F. Conformity of MIS-PyME-3M and ISO/IEC 15504

This section shows the conformity of MIS-PyME-3M with ISO/IEC 15504. ISO/IEC 15504 indicates that in order to assure that the assessment results are translatable into an ISO/IEC 15504 process profile in a repeatable and reliable manner, Process Assessment Models shall adhere to certain requirements. A Process Assessment Model shall contain a definition of its purpose, scope and elements; its mapping to the measurement framework and specified Process Reference Model(s); and a mechanism for consistent expression of results.

As regards the process model scope the requirements are as follows:

6.3.2.1. A Process Assessment Model shall relate to at least one process from the specified Process Reference Model(s). *6.3.2.2 A Process Assessment Model shall address, for a given process, all, or a continuous subset, of the levels (starting at level 1) of the Measurement Framework for process capability for each of the processes within its scope.* *6.3.2.3 A Process Assessment Model shall declare its scope of coverage in the terms of:* *a) the selected Process Reference Model(s);* *b) the selected processes taken from the Process Reference Model(s);* *c) the capability levels selected from the Measurement Framework*

Table F.1. Requirements for a Process Assessment Model Scope ISO/IEC 15939 (ISO/IEC, 2002)

- 6.3.2.1: The process reference model is COMPETISOFT and MIS-PyME-3M relates to the measurement process.
- 6.3.2.2: MIS-PyME-3M addresses levels 1 to 3.
- 6.3.2.3: The selected process reference model is COMPETISOFT, MIS-PyME-3M is focused on the measurement process and the capability levels selected are the capability levels from 1 to 3.

As regards Process Assessment Model indicators:

A Process Assessment Model shall be based on a set of indicators that explicitly addresses the purposes and outcomes, as defined in the selected Process Reference Model, of all the processes within the scope of the Process Assessment Model; and that demonstrates the achievement of the process attributes within the capability level scope of the Process Assessment Model. The indicators focus attention on the implementation of the processes in the scope of the model.

Table F.2. Requirements for Process Assessment Model Indicators ISO/IEC 15939 (ISO/IEC, 2002)

The indicators are defined in three dimensions: the performance and establishment of the process indicators, the indicators related with information needs obtained from the measurement process and the indicators related with the resources which support the measurement process. The two first indicators directly deal with the purpose and scope of the measurement process. The third type of indicators deals with the resources required to achieve those outcomes and purposes.

As regards the mapping between the process assessment models to process references modes ISO/IEC requires:

A Process Assessment Model shall provide an explicit mapping from the relevant elements of the model to the processes of the selected Process Reference Model and to the relevant process attributes of the measurement framework. *The mapping shall be complete, clear and unambiguous. The mapping of the indicators within the Process Assessment Model shall be:* *a) the purposes and outcomes of the processes in the specified Process Reference Model;* *b) the process attributes (including all of the results of achievements listed for each process attribute) in the measurement framework.* *This enables Process Assessment Models that are structurally different to be related to the same Process Reference Model.*

Table F.3. Requirements for Mapping the Process Assessment Models to Process References Modes ISO/IEC 15939 (ISO/IEC, 2002)

MIS-PyME 3M attributes are designed in three views:

- Performance and establishment (P). These attributes determine whether the measurement process meets its outcomes of the measurement process and the extent of which the process is established.
- Input (I): These attributes determine the type and difficulty of the information needs which are obtained from the measurement process, thus the type of measurement indicators of the measurement process. Measurement indicators are the main workproduct of the measurement process and as more mature the process is, the more difficult information can be obtained.
- Resources (R): These attributes are based on the tools and infrastructure that support the measurement process.

MIS-PyME-3M was designed in this way since it is more easy and clear in order to understand the measurement maturity of the company. However it can be viewed in one dimension and therefore it perfectly maps with ISO/IEC 15504.

The “indicators” related to “I” and “R” attributes, can be considered as a way for determining whether the measurement process has fulfil its purpose in each capability level.

Therefore the mapping of MIS-PyME-3M and the measurement framework ISO/IEC 15504 is shown in Table F.4

MIS-PyME MMM Code	**Indicator Name (MIS-PyME-3M)**	**Maps to (ISO/IEC 15504)**
PA 1.1 (P) Process performance attribute		PA 1.1
PA 1.1. (P).a	The process achieves its defined outcomes	PA 1.1.a
PA 1.2 (I) Basic project and product focus information attribute		PA 1.1
PA 1.2 (I).a	Tracking project schedule/plan (phase by phase)	PA 1.1.a
PA 1.2 (I).b	Tracking critical product reliability in production	PA 1.1.a
PA 1.3 (R): Basic management tools implemented attribute		PA 1.1.a
PA 1.3 (R).a	Incidents tool (production)	PA 1.1.a
PA 1.3 (R).b	Basic project management tool	PA 1.1.a
PA 2.1 (P) Performance management attribute		PA 2.1
PA 2.1 (P).a	Objectives for the performance of the process are identified	PA 2.1.a
PA 2.1 (P).b	Performance of the process is planned and monitored	PA 2.1.b
PA 2.1 (P).c	Performance of the process is adjusted to meet plans.	PA 2.1.c
PA 2.1 (P).d	Responsibilities and authorities for performing the process are understood, assigned and communicated.	PA 2.1.d
PA 2.1 (P).e	Interfaces between the involved parties are managed to ensure both effective communication and clear assignment of responsibilities.	PA 2.1.f
PA 2.2 (P) Workproduct management attribute		PA 2.2
PA 2.2 (P).a	requirements for the work products of the process are defined	PA 2.1.a
PA 2.2 (P).b	requirements for documentation and control of the work products are defined	PA 2.1.b
PA 2.2 (P).c	work products are appropriately identified, documented, and controlled	PA 2.1.c
PA 2.2 (P).d	work products are reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements	PA 2.1.d
PA 2.3 (I) project and product focus management attribute		PA 2.1
PA 2.3 (I).a	Manage planning information	PA 2.1.a,b,c
PA 2.3 (I).b	Manage reliability of the products during development	PA 2.1.a,b,c
PA 2.3 (I).c	Manage customer satisfaction	PA 2.1.a,b,c
PA 2.4 (R): management and development tools implementation attribute		PA 2.1
PA 2.4 (R).a	Test cases management tools	PA 2.1.e
PA 2.4 (R).b	Project management tools for tracking effort and cost are in place	PA 2.1.e
PA 3.1(P) Process definition attribute		PA 3.1
PA 3.1(P).a	A standard process, including appropriate tailoring guidelines, is defined which describes the fundamental elements that must be incorporated into a defined process	PA 3.1.a

PA 3.1(P).b	The sequence and interaction of the standard process with other processes is determined	PA 3.1.b
PA 3.1(P).c	Required infrastructure and work environment for performing a process are identified as part of the standard process	PA 3.1.d
PA 3.1(P).e	Suitable methods for monitoring the effectiveness and suitability of the process are determined	PA 3.1.e
PA 3.2 (P) Process deployment attribute		PA 3.2
PA 3.2 (P).a	A defined process is deployed based upon an appropriately selected and/or tailored standard process	PA 3.2.a
PA 3.2 (P).b	required roles, responsibilities and authorities for performing the defined process are assigned and communicated	PA 3.2.b
PA 3.2 (P).c	The personnel performing the defined process are competent with regard to appropriate education, training, and experience.	PA 3.2.c
PA 3.2 (P).d	Appropriate data are collected and analysed as a basis for understanding the behaviour of the process, to demonstrate the suitability and effectiveness of the process and to evaluate where continuous improvement of the process can be made.	PA 3.2.d
PA 3.3 (I) Standard and advanced measurement		PA 3.1
PA 3.3 (I).a	Cross-project analyses available	PA 3.1.b
PA 3.3 (I).b	Manage development progress	PA 3.1.b
PA 3.3 (I).c	Requirements stability is managed	PA 3.1.b
PA 3.4 (I) Advanced product tracking attribute		PA 3.1.b
PA 3.4 (I).a	Manage quality information	PA 3.1.b
PA 3.4 (I).b	Cross-product analysis available	PA 3.1.b
PA 3.5 (I) Process tracking attribute		PA 3.1
PA 3.5 (I).a	Process efficiency and effectiveness	PA 3.1.b
PA 3.5 (I).b	Productivity normal ranges	PA 3.1.b
PA 3.5 (I).c	Manage the process compliance	PA 3.1.b
PA 3.5 (I).d	Cross-processes analysis	PA 3.1.b
PA 3.5 (I).c	Fix defect time	PA 3.1.b
PA 3.5 (I).d	Downtime	PA 3.1.b
PA 3.6 (R) Resources deployment attribute		PA 3.1
PA 3.6 (R).a	Organization Measure Database	PA 3.1.d
PA 3.6 (R).b	Life cycle configuration management tool	PA 3.1.d
PA 3.6 (R).c	Training Program	PA 3.2.e
PA 3.6 (R).d	Life cycle configuration management tool	PA 3.1.d
PA 3.6 (R).e	Static quality analysers tools	PA 3.1.d
PA 3.6 (R).f	Resources Management	PA 3.1.d
PA 3.6 (R).g	Project estimation techniques	PA 3.1.d
PA 3.6 (R).h	Procedures in the use of these tools are well understood and are standardized throughout the organization	PA 3.2.e

Table F.4. Mapping MIS-PyME 3M to ISO/IEC 15939(ISO/IEC, 2002)

As regards the expression of assessment results,

A Process Assessment Model shall provide a formal and verifiable mechanism for representing the results of an assessment as a set of process attribute ratings for each process selected from the specified Process Reference Model(s).

Table F.5. Requirements of the Expression of Assessment Results of 15939 (ISO/IEC, 2002)

MIS-PyME 3M presents the results as ISO/IEC 15504 does. Therefore the company is characterized by a set of profiles which consist on the evaluation of each of the attributes defined in MIS-PyME-3M which determine its measurement maturity level. In addition a set of suggestions of improvement should be identified and presented as the result of the assessment too.

G. MIS-PyME and Other Contributions

Table G-1 maps the popular methodologies, GQM, GQ(I)M, ISO/IEC 15939, PSM, and GQM lightweight method and MIS-PyME in order to show the differences with regard the methodology steps. Besides, in addition to the main contribution of MIS-PyME explained in Chapter 4, we show the aspects that makes this methodology different from the others.

- **Measurement process definition**: MIS-PyME highlights the formal definition of the measurement process which will carry out the measurement program. This is an issue with which the above methodologies do not deal in detail. This aspect gains importance in MIS-PyME since it is focused on defining measurement programs which should become a standard measurement program used in the company and therefore will be reused.
- **Special attention to the integration**: Special importance is giving to the integration of the measurement program into the other organization processes. PSM and ISO/IEC 15939 also give special importance to this task since they define a specific task for this issue. MIS-PyME guides in defining the measurement process in accordance with the other processes of the organization and integrating the indicator results into the reports of the other processes of the organization. GQM lightweight method highlights the importance of integration but it does not propose any new practice for doing this.
- **The roles required for performing each task are determined**: MIS-PyME defines the roles of the people who should perform each of tasks proposed. In addition it also determines the number of people who would play that role and its profile in the organization. None of the methodologies explicitly indicates the role who performs each of the measurement tasks defined and only GQM lightweight method indicates the number of people involved in the measurement program and focused on SMEs.
- **Support modules**: MIS-PyME provides support modules for defining the whole measurement program till its implementation: MIS-PyME measurement goals table, MIS-PyME indicator templates, MIS-PyME database. Only PSM provides such kind of support. However MIS-PyME measurement goals table provides some benefits as over and against PSM. The main benefits are as follows.
    1. The MIS-PyME software measurement goals table relates software measurement goals to software process improvement goals. MIS-PyME and other methodologies such as GQM (Solingen and Berghout, 1999) considers that measurement programs should support process improvement, this not being an end in itself but rather a means to achieve a clear goal for the process. The MIS-PyME software measurement goals table places the user in this context and guides him/her towards the definition of a measurement program with the process improvement goal in mind. PSM, however, only

relates measures to project issues, so the scope of MIS-PyME is wider, since it specifies the indicators required to support all levels of an organization that develops and maintains software products. As an example, there are some indicators which measure individual projects, labelled as "PRJ", but others which make cross-projects analyse "PRJORG", in order to understand the general tendency in the organization, and others are related to all the products of the organization, "PRODORG", in a quest to understand the value and other characteristics of the services provided in general.

2. PSM does not address information regarding the maturity required to implement certain indicators. It addresses information regarding the constraints on the type of analysis (e.g. part 5) or on the tools required (e.g. part 2), etc. but it does not relate these constrains to a measurement maturity model where the limitations and evolution are clearly described. What is more, it lacks information regarding the constraints of the purpose of some indicators and the scope of these.

- **Defining the measurement program using reusable elements**: all the elements of the measurement program are easily reusable. The measurement program finally consists on three elements:

  1. The indicator definition which contains all the information required for building and analysing the indicators for achieving a specific measurement goal.
  2. The measure definition which includes the definition of the measure and the procedures for collecting the data required for obtaining the result of the measure.
  3. The measurement process, which describes the specific activities and tasks which should be performed in order to analyse the indicators and achieve the measurement goals.

All these elements are easily reusable. GQM for example purpose a GQM plan, and a measurement and analysis plan. These plans are focused on a specific project and the reusable entities are not easily identified.

GQM lightweight method proposes the packing activity at the end of the method as a practice to ease the reuse of the measurement program. However MIS-PyME is focused on the reuse of the measurement program from the first activity of the method. Indicators and measurements are defined for being easily reused, all their related information is included in a reusable object. In addition the measurement processes are integrated into the other processes of the organization (e.g. resources management, specific project administration, project management, etc.). When one of these processes is applied (e.g. specific project administration) their associated measurement processes are also easily applied and therefore the measurement programs defined are easily reused.

Phases	GQM	GQ(I)M	GQM lightweight method	MIS-PyME	ISO/IEC 15939	PSM
Planning,					Accept requirement for measurement: Identify the measurement scope, management commitment and communication commitment	Obtaining Organizational
	Defining GQM team		Introduce measurement program	Initiate the measurement program	Assign resources	Support, Defining Responsibilities, and Providing Resources
	Select improvement areas	Identify business goals and identify what you want to learn			Characterize organization unit and Identify information needs	Identifying and prioritizing project issues
	Select application project and establish project team			Specifying the measurement plan and Introductory session		
Definition:	Define measurement goals, conduct GQM interviews, review or produce software process models.	Identify sub-goals, identify entities and attributes formalize measurement goals	Define measurement goals, goals formalization.	Formalizing measurement goals and checking whether a measurement program is reused		

	Define questions and hypothesis and review it	Identify quantifiable questions and indicators	Defining questions	Defining indicators		
	Produce analysis plan	Identify data elements				
	Define metrics and review	Define measures	Define metrics	Defining measures	Select and specify measures	Selecting and specify project measures
	Produce GQM and measurement plan	Identify the actions needed to implement measures, prepare the plan.	Produce GQM plan, define data collection procedures, define data instruments	Specifying the collection procedures for getting the results of the measure	Define data collection, analysis, and reporting procedures	Integrating measures into the technical and management processes - Define data collection, define indicators, reporting procedures
				Integrating the measurement program	Define criteria for evaluating the information products and the measurement process	
	Review plans		Produce data collection plan	Measurement program verification	Review, approve, and provide resources for measurement tasks	
Integration/ Data collection phase	Hold Trial period: Define collecting procedures, create metrics base, etc. and testing them.		create metrics base	Instrumentation	Acquire and deploy supporting technologies	
				Trial of the measurement program	Integrate procedures	

	Hold kick-off sessions The objective is that all participants agree on the data collection activities and be informed of the procedures, tools and templates			Close of the acceptance task		
				Verifying and adapting the measurement process to be applied		
	Collect and check data collection form, store measurement data in metrics base		Collect and validate data. Store data collected	Collecting the data	Collect data: collect data, store it and verify it	Collect data: collect data, store it, verify it and normalize data
The Analysis and Interpretation phase.	Define analysis sheets and presentation slides		Data analysis	Building the indicators	Analyse data, Interpret results, and develop and verify information products. Communicate results	Analyse indicators
	Prepare feed-back sessions, organize feed-back session, report measurement results.		Data interpretation/ feedback sessions	Analysing and Interpreting the data Communicating the results		Reporting Results and Using Results for decision making
Evaluate measurement				Evaluating the measurement program: evaluating, updating and lessons learned identified.	Evaluate information products and the measurement process, Identify potential improvement and identify lessons learned (measurement experience base)	Evaluating measures and indicadors, the measurement process and updating the experience database.
Packaging			Packing results			

Table G-1. Mapping of GQM, GQ(I)M, ISO/IEC 15939, PSM, GQM lightweight and MIS-PyME

ANNEXES

A. Acronyms
B. References

A. Acronyms

AIQM	Asian Institute of Quality Management
AR	Action-Research
BMP	Balancing Multiple Perspectives
CMMI	Capability Maturity Model Integration
DBMS	Database Management System
EMS	Experience Management System
EXMP	Measurement EXaMPles
FPP	Few People involved in the Process
FSTEP	Few but effective and complete Steps
GB&P	Specific Guidelines to understand the Benefits & Potential for management
GINT	Specific Guidelines to INTegrate measurement in the software processes
GMM	Specific Guidelines to adapt measurement definition to the Measurement Maturity of the company
GPIN	Specific Guidelines to support Basic Process Improvement Needs
GQ(I)M	Goal Question Indicator Metric
GQM	Goal Question Metric
HP	Hewlett-Packard
IBM	International Business Machines
IS	Information Systems
ISBSG	International Software Benchmarking Standards Group
ISO	International Organization for Standardization
KA	Key Area
M&A	Measurement and Analysis
MIS-PyME-3M	MIS-PyME Measurement Maturity Model
MP	Measurement Program
OMG	Object Management Group
P.I.G	process improvement goals
PA	Process Area
PRJ	Project
PROC	Process
PROD	Product
PSM	Practical Software and Systems Measurement
QIP	Improvement Paradigm
RES	Resource
RUSE	Reuse Measurement Models

SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SE	Software Engineering
SEI	Carnegie Mellon Software Engineering Institute
SME	Small and medium-sized enterprise
SME	Software Management Environment
SPC	Software Process Control
SPEM	Software Process Engineering Metamodel
SPI	Software Process Improvement
STL	Sistemas Técnicos de Loterías del Estado
XML	Extensible Markup Language

B. Reference

Abran, A., Buglione, L., and Sellami, A. (2004a). *Software Measurement Body of Knowledge – Initial Validation using Vincenti's Classification of Engineering Knowledge types.* Paper presented at the Software Measurement Conference.1-16

Abran, A., Moore, J. W., Bourque, P., Dupuis, R., and Tripp, L. L. (2004b). *Guide to the Software Engineering Body of Knowledge-2004 Version*. Los Alamitos, California: IEEE Computer Society.

Avison, D., Lan, F., Myers, M., and Nielsen, A. (1999). Action Research. *Communications of the ACM, 42(1)*, 94-97.

Baldassarre, M. T., Boffoli, N., Caivano, D., and Visaggio, G. (2004). *Managing Software Process Improvement (SPI) through Statistical Process Control (SPC).* Paper presented at the Product Focused Software Development and Process Improvement (PROFES). 30-46

Baldassarre, M. T., Boffoli, N., Caivano, D., and Visaggio, G. (2005). *Improving Dynamic Calibration through Statistical Process Control.* Paper presented at the 21st IEEE International Conference on Software Maintenance (ICSM 2005), Budapest, Hungary. 273-282

Baldassarre, M. T., Caivano, D., and Visaggio, G. (2003). *Comprehensibility and Efficiency of Multiview Framework for Measurement Plan Design.* Paper presented at the second International Symposium on Empirical Software Engineering (ISESE'03). 89 - 98

Basili, V. R. (1992). *Software modeling and measurement: The Goal/Question/Metric paradigm. Technical Report, CS-TR-2956*: University of Maryland.

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994a). The experience factory. In J. J. M. (Ed.), *Encylopedia of Software Engineering* (Vol 1, pp. 469-476): John Wiley & Sons.

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994b). Goal Question Metric Paradigm. In J. W. Sons (Ed.), *Encyclopedia ofSoftware Engineering* (Vol. 1, pp. 528-532).

Basili, V. R., and Rombach, D.H. (1988a). The Tame Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions of Software Engineering, 14*(6), 758-773.

Basili, V. R., and Rombach, H. D. (1988b). Support for comprehensive reuse. *IEEE Software Engineering Journal*, 758-773.

Basili, V. R., and Weiss, D. (1984). A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering, 10*(11), 758-773.

Basili, V. R. (2000). *Using Experiments to Build a Body of Knowledge.* Paper presented at the 7th European Workshop on Software Process Technology (EWSPT 2000). Salzburg (Austria). 265-282.

Basili, V. R., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., and Trendowicz, A. (2007). *GQM+ strategies - Aligning business strategies with software measurement.* Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, ESEM., Madrid, Spain. 488-490

Baskerville, R., and Wood-Harper, A. T. (1996). A Critical Perspective on Action Research as a Method for Information Systems Research. *Journal of Information Technology, 3*(11), 235-246.

Baskerville, R. (1997). Distinguishing Action Research From Participative Case Studies. *Journal of Systems and Information Technology, 1*(1), 25-45.

Baskerville, R. (1999). Investigating Information Systems with Action Research. *Communications of the Association for Information Systems, 2(*19*).*

Bassman, M. J., McGarry, F., and Pajerski, R. (1995). *Software measurement guidebook.* Maryland: NASA Goddard Space Flight Center, Software Engineering Laboratory.

Bégnoche, L., Abran, A., and Buglione, L. (2007). *A Measurement Approach Integrating ISO 15939, CMMI and the ISBSG.* Paper presented at the Software Measurement European Forum (SMEF). 111-130.

Benbasat, I., Goldstein, D. K., and Mead, M. (1987). *The Case Research Strategy in Studies of Information Systems*. *MIS Quarterly, 11*(3), 369-386.

Berander, P., and Jönsson, P. (2006). *A goal Question Metric Based Approach for Efficient Measurement Framwork Definition.* Paper presented at the ISESE'06. 316-325.

Berry, M., and Vandenbroek, M. F. (2001). *A Targeted Assessment of the Software Measurement Process.* Paper presented at the 7th International Software Metrics Symposium (METRICS .01). 222 -236.

Berry, M., Jeffery, R., and Aurum, A. (2004). *Assessment of Software Measurement: an Information Quality Study.* Paper presented at the 10th International Symposium on Software Metrics (METRICS'04). 314-325.

Birk, A., Solingen, R. v., and Järvinen, J. (1998). *Business Impact, Benefit, and Cost of Applying GQM in Industry: An Indepth, Long-term Investigation at Schlumberger RPS,.* Paper presented at the 5th International Symposium on Software Metrics (METRICS'98), Bethesda Maryland. 93-96.

Boffoli, N. (2006). *Non-Intrusive Monitoring of Software Quality.* Paper presented at the European Conference on Software Maintenance. 318-322.

Boffoli, N., Bruno, G., Caivano, D., and Mastelloni, G. (2008). *Statistical process control for software: a systematic approach.* Paper presented at the ACM-IEEE international symposium on Empirical software engineering and measurement. 327-329.

Brereton, P., Kitchenham, B., and Budgen, D. (2008). *Using a Protocol Template for Case Study Planning.* Paper presented at the EASE: EBSE.

Briand, L. C., Differding, C. M., and Rombach, H. D. (1996a). Practical Guidelines for Measurement-Based Process Improvement. *Software Process - Improvement and Practice, 2(4)*, 253-280.

Briand, L.C., Morasca, S., and R.Basili, V. R. (1996b). Property-based software engineering measurement. *IEEE Transactions of Software Engineering, 22(1)*, 68-86.

Briand, L. C., Morasca, S., and Basili, V. R. (2002). An Operational Process for Goal-Driven Definition of Meassures. *IEEE Transactions on Software Engineering, 28(12)*, 1106-1125.

Brijckers, A., and Differding, C. (1996). *The Role of Software Process Modeling in Planning Industrial Measurement Programs.* Paper presented at the 3rd International Sysmposium on Software Metrics (METRICS'96).

Buglione, L., and Dekkers, C. (2006). *A Murphological View on Software Measurement: a serious joke or a funny serious thing?* Paper presented at the Smef 2006, Rome (Italy). 315-329.

Caivano, D. (2005). *Continuous Software Process Improvement through Statistical Process Control.* Paper presented at the 9th European Conference on Software Maintenance and Reengineering (CSMR 2005), Manchester, UK. 288-293.

Chein, I., Cook, S. W., and Harding, J. (1948). The Field of Action Research. *The American Psychologist, 3*, 43-50.

Ciolkowski, M., Heidrich, J., Simon, F., and Radicke, M. (2008). *Empirical Results from Using Custom-Made Software Project Control Centers in Industrial Environments.* Paper presented at the Empirical Software Engineering and Measurement. 243-252.

Clark, B. (2002). Eight secrets of software measurement. *IEEE Software, 19(5)*, 12–14.

CMMI. (2006). *CMMI for Development, V1.2: Improving processes for better products*. Pittsburgh: Carnegie Mellon University.

CMMI-Product-Team. (2006). *Capability Maturity Model Integration for Development.CMU/SEI-2006-TR-008*: Software Engineering Institute, Pittsburgh: Carnegie Mellon University.

Comer, P., and Chard, J. (1993). A measurement maturity model. *Software Quality Journal, 2(4)*, 277–289.

Cuadrado-Gallego, J. J., Buglione, L., and Rodríguez, D. (2007). *Suggestions for Improving Measurement Plans: A BMP application in Spain.* Paper presented at the Software Measurement European Forum.

Daskalantonakis, M. K., Yacobellis, R. H., and Basili, V. R. (1990). A Method for Assessing Software Measurement Technology. *Quality Engineering, 3(1)*, 27-40.

Daskalantonakis, M. K. (1992). A Practical View of Software Measurement and Implementation Experiences Within Motorola. *IEEE Transactions on Software Engineering, 18*(11), 998-1010.

Dekkers, C., and Bradley, M. (1999). It is the people who count in measurement. The truth about measurement myths. *STSC CrossTalk*, 12-14.

Dekkers, C., and McQuaid, P. (2002). The dangers of using software metrics to (mis)manage. *IEEE IT Professional*, 24–30.

Dekkers, T. (2005). *Basic measurement implementation: away with the crystal ball.* Paper presented at the Software Measurement European Forum, The Netherlands. 113–122.

DeMarco, T. (1982). *Controlling Software Projects - Management, Measurement & Estimation*. Englewood Cliffs, NJ, USA.

Differding, C. M. (2001). *Adaptive measurement plans for software development.* Kaiserslautern, Stuttgart.

Dittrich, Y., Rönkkö, K., Eriksson, J., Hansson, C., and Lindeberg, O. (2008). Cooperative Method Development. Combining Qualitative Empirical Research with Method. *Technique and Process Improvement - Empirical Software Engineering, 13(3)*, 231-260.

DoD (Ed.). (2000). *PSM: Practical Software and Systems Measurement - A Foundation for Objective Project Management Version 4.0c*: US DoD.

Dumke, R., Abran, A., and Buglione, L. (2006). *Suggestions for Improving Measurement Plans: First results from a BMP application.* Paper presented at the Software Measurement European Forum.

Eickelmann, N. S. (2001). *Empirical Studies to Identify Defect Prevention Opportunities Using Process Simulation Technologies.* Paper presented at the 26th Annual NASA Goddard Software Engineering Workshop (SEW.01). 22-26.

ESI. (2007). *Europe Software Institute. Available on: www.esi.es/en/main/iitmark.html*

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management Review, 14*, 532-550.

Estay, C., and Pastor, J. (2000a). *Improving Action Research in Information Systems with Project Management.* Paper presented at the Americas Conference on Information Systems, Long Beach, California. 1558-1561

Estay, C., and Pastor, J. (2000b). *Towards a project structure for Action-Research in Information Systems.* Paper presented at the 10th Annual Business and Information Technology Conference (BIT'2000), Manchester (United Kingdom).

Estay, C., and Pastor, J. (2001). *Un Modelo de Madurez para Investigación-Acción en Sistemas de Información.* Paper presented at the VI Jornadas de Ingeniería del Software y Bases de Datos (JISBD'2001), Ciudad Real (España). 265-281.

Florac, W. A. y Carleton, A. D. (1999). Measuring the Software Process. Statistical Process Control for Software Process Improvement. Addison Wesley.

French, W. L., and Bell, C. H. (1996). *Organizational Development: Behavioral Science Interventions for Organization Improvement*. London: Prentice Hall.

Geppert, B., and Weiss, D. M. (2003). *Goal-Oriented Assessment of Product-Line Domains.* Paper presented at the 9th International Software Metrics Symposium (METRICS'03).

Goethert, W., and Hayes, W. (2001). *Experiences in Implementing Measurement Programs.* No.Report CMU/SEI-2001-TN-026: Carnegie Mellon University

Goethert, W., and Siviy, J. (2004). *Applications of the Indicator Template for Measurement and Analysis*. No.Report CMU/SEI-2004-TN-024

Goldenson, D. R., Gopal, A., and Mukhopadhyay, T. (1999). *Determinants of Success in Software Measurement Programs: Initial Results.* Paper presented at the 6th IEEE International Symposium on Software Metrics (METRICS'99). 10-21.

Goldenson, D., Rout, T., and Tuffley, A. (2005). *Measuring Performance Results in Small Settings: How do you do it and what matters most?* Paper presented at the First International Research Workshop for Process Improvement in Small Settings.41-44.

Gorschek, T., Garre, P., Larsson, S., and Wohlin, C. (2006). Model for Technology Transfer in Practice. *IEEE Software, 23(6)*, 88-95.

Goodman, P. (2004). *Software Metrics — Best Practices for Successful IT Management*. FBCI, Brookfield, CT, USA: Philip Jan Rothstein.

Gopal, A., Mukhopadhyay, T., and Krishnan, M. S. (2005). The impact of institutional forces on software metrics programs. *IEEE Transactions on Software Engineering, 31*(8), 679–694

Gopal, A., Krishnan, M. S., Mukhopadhyay, T., and Goldenson, D. R. (2002). Measurement Programs in Software Development: Determinants of Success. *IEEE Transactions on Software Engineering, 28*(9), 863-875.

Grady, R. B., and Caswell, D. L. (1987). *Software Metrics: Establishing a Company-Wide Program*. Englewood Cliffs, NJ, USA: Prentice Hall.

Gresse, C., HOISL, B., and Wüst, J. (1995). *A process model for planning GQM-based measurement. Software Technology Transfer Initiative.* No.Report STTI-95-04-E: Dept. of Computer Science, Univ. of Kaiserslautern, Kaiserslautern, Germany.

Gresse, C. (2002a). *Operationalizing the reuse of software measurement planning knowledge.* Kaiserslautern, Germany.

Gresse, C. (2002b). *Planing and Executing GQM-Based Software Measurement.* No.Report LQPS001.01E, UNIVALI, Sao José, Brazil.

Gresse, C., Punter, T., and Anacleto, A. (2003). *Software measurement for small and medium enterprises.* Paper presented at the 7th International Conference on Empirical Assessment in Software Engineering (EASE), Keele, UK.

Hall, T., and Fenton, N. (1997). Implementing Effective Software Metrics Programs. *IEEE software, 14*(2), 55-65.

Heidrich, J., and Münch, J. (2008). *Goal-Oriented Setup and Usage of Custom-Tailored Software Cockpits.* Paper presented at the Product Focused Software Process Improvement (PROFES '08). 4-18.

Hendrick, R., Kstler, D., and Valett, J. (1992). *Software Management Environment (SME) - Concepts and Architecture.* No.Report SEL-89-103: NASA Goddard Space Flight Center Code 551.

IEEE. (2003). IEEE Std 1490™-2003 Adoption of PMI StandardA Guide to the Project Management Body of Knowledge -Description.

IEEE (1990). *IEEE 610.12:1990 standard glossary of software engineering terminology.*: Computer Society IEEE.

ISO/IEC (2002). *ISO/IEC 15939, Software Engineering-Software Measurement Process*.

ISO/IEC. (2007). *ISO/IEC NP TR 29110: Software Engineering -- Lifecycle Profiles for Very Small Enterprises (VSE).*

Iversen, J., and Mathiassen, L. (2003). Cultivation and engineering of a software metrics program. *Information Systems Journal, 13* (1), 3–19.

Iversen, J. H., Mathiassen, L., and Nielsen, P. A. (2004). Managing Risk in Software Process Improvement: An Action Research Approach. *MIS Quarterly, 28*(3), 395-433.

Jedlitschka, A., and Pfahl, D. (2005). Reporting Guidelines for Controlled Experiments in Software Engineering. *ACM/IEEE International Symposium on Empirical Software Engineering*, 95-104.

Jeffery, R., and Berry, M. (1993). *A framework for evaluation and prediction of metrics program success.* Paper presented at the 1st International Software Metrics Symposium, Los Alamitos, CA, USA. 28–39.

Johnson, K., and Kulps, M. (2004). *Measurement within the CMMI.* Paper presented at the Teraquest SEPG 2004; Orlando, Florida.

Jones, C. L. (2003). Implementing a successful measurement program: Tried and true practices and tools. *Cutter IT Journal, 16(11)*, 12-18.

Kasunic, M. (2006). *The State of Software Measurement practice: Results of 2006 Survey*. No. Report. CMU/SEI-2006-TR-009: Software Engineering Institute.

Kautz, K. (1999). Making sense of measurement for small organizations. *IEEE Software, 16*(2), 14–20.

Kenett, R., and Bake, E. (1999). *Software Process Quality : Management and Control*. New York, NY, USA: Computer Aided Engineering.

Kitchenham, B., Pfleeger, S. L., and Fenton, N. (1995). Towards a Framework for Software Measurement Validation. *IEEE Transactions on Software Engineering, 21*(12), 929-944.

Kitchenham, B. (1996). *Software Metrics — Measurement for Software Process Improvement*. Oxford, UK: NCC Blackwell.

Kock, N., and Lau, F. (2001). Information Systems Action Research: Serving Two Demanding Masters. *Information Technology & People (special issue on Action Research in Information Systems), 14*(1), 6-11.

Laporte, C. Y., Alexandre, S., and Renault, A. (2008). Developing International Standards for Very Small Enterprises. *IEEE, 41* (3), 98-101.

Lavazza, L., and Barresi, G. (2005). *Automated Support for Process-aware Definition and Execution of the Measurement Plans.* Paper presented at the 27th international conference on Software engineering, St. Louis, MO, USA. 234-243.

Lau, F. (1997). A Review on the Use of Action Research in Information Systems Studies. In A. S. Lee, J. Liebenau andJ. I. Degross (Eds.), *Information Systems Research: Information Systems and Qualitative Research.* (pp. 31-68). London: Chapman & Hill.

Layman, B., and Rohde, S. (1999). Experiences Implemeting a Software Project Measurement Methodology. *ASQ, 99*(2).

Lee, Y. W., Strong, D. M., Kahn, B. K., and Wang, R. Y. AIMQ: A Methodology for Information Quality Assessment. *Information & Management, 40*(2), 133-146.

Lindvall, M., Donzelli, P., Asgari, S., and Basili, V. (2005). *Towards Reusable Measurement Patterns.* Paper presented at the 11th IEEE International Software Metrics Symposium (METRICS 2005). 21-29.

Lindvall, M., Tesoriero, R., and Costa, P. (2002). *Avoiding Architectural Degeneration: An Evaluation Process for Software Architecture.* Paper presented at the Pr oceedings of the 8th IEEE Symposium on Software Metrics (METRICS'02). 77 - 86.

Luna, I. C. (2008). *Diagnóstico sobre la situación de la calidad del software en la industria española.* Paper presented at the Jornadas Profesionales de Calidad y Testing de Software (EXPO QA), Madrid.

MacDonell, S. G., and Fletcher, T. (1998). *Metric Selection for Effort Assessment in Multimedia Systems Development.* Paper presented at the 5th International Sotware Metrics Symposium, Bethesda, MD, USA. 97-100.

McGarry, F., Burke, S., and Decker, B. (1998). *Measuring the Impacts Individual Process Maturity Attributes Have on Software Products.* Paper presented at the 5th International Symposium on Software Metrics (METRICS'98). 52-61.

McTaggart, R. (1991). Principles of Participatory Action Research. *Adult Education Quarterly, 41*(3).

Mendonça, M. G., and Basili, V. R. (2000). Validation of an Approach for Improving Existing Measurement Frameworks. *IEEE Transactions on Software Engineering*, 494-499.

Metzker, E. (2003). *An Experimental Process Metrics Support Environment and a Cross-Organizational Study on its Acceptance by Practitioners.* Paper presented at the 9th International Software Metrics Symposium (METRICS'03). 135 - 143.

Minkiewicz, A. (2000). *Software measurement? what's in it for me?* Paper presented at the SM /ASM.

Moher, T., and Schneider, G. M. (1981). *Methods for improving controlled experimentation in software engineering.* Paper presented at the 5th Conference on Software Engineering. 224 - 233.

Morasca, S. (2008). *Refining the axiomatic definition of internal software attributes.* Paper presented at the 2nd Empirical Software Engineering and Measurement (ESEM'08). 188-197.

Musa, J. D., Iannino, A., and Okumoto, K. (1987). Software Reliability: Measurement, Prediction, Application. *IEEE Computer Society*.

Niessink, F., and Vliet, H. V. (1998). *Towards Mature Measurement Programs.* Paper presented at the Software Maintenance and Reengineering. 8(11). 82-88.

Niessink, F., and Vliet, H. V. (1999). *Measurements Should Generate Value, Rather Than Data.* Paper presented at the 6th International Software Metrics Symposium (METRICS'99), Boca Raton. 31-39.

Niessink, F., and Vliet, H. V. (2001). Measurement Programs Success factors revised. *Information and Software Technology, 43*, 617-628.

OMG. (2009). *Software Process Engineering Metamodel (SPEM)*, from http://www.omg.org/technology/documents/formal/spem.htm

Oktaba, H. (2005). *MoProSoft: A Software Process Model for Small Enterprises.* Paper presented at 1st International Research Workshop for Process Improvement in Small Settings, Pittsburgh, United States.

Oktaba, H., García, F., Piattini, M., Ruiz, F., Pino, F. J., and Alquicira, C. (2007). Software Process Improvement: The Competisoft Project. *IEEE Computer Society*, 21-28.

Padak, N., and Padak, G. (1994). *Guidelines for Planning Action Research Projects*, from http://archon.educ.kent.edu/Oasis/Pubs/0200-08.html

Park, R. E., Goethert, W. B., and Florac, W. A. (1996). *Goal-Driven Software Measurement-A Guidebook*: Carnegie Mellon University Pittsburgh: Software Engineering Institute.

Pfleeger, S. L. (1993). Lessons learned in building a corporate metrics program. *IEEE Software, 10*(3), 67–74.

Pfleeger, S. L. (1997). Assessing measurement. *IEEE Software, 14*(2), 25-26.

PPfleeger, S. L. (1999). Understanding and Improving Technology Transfer in Software Engineering. *Systems and Software, 47*(2-3), 111-124.

PMI. (2000). *PMBOK: A Guide to the Project Management Body of Knowledge*. Unite States: Project Management Institute Communications.

Ramil, J. F., and Lehman, M. M. (2001). *Defining and Applying Metrics in the Context of Continuing Software Evolution.* Paper presented at the 7th International Software Metrics Symposium (METRICS'01). 199-209.

Richardson, I., and Wangenheim, C. G. v. (2007). Why are Small Software Organizations Different? *IEEE Software, 24*(1), 18-22.

Rifkin, S. (2001). What makes measuring software so hard? *IEEE Software Engineering Journal, 18* (3), 41-45.

Rombach, H. D. (1991). Practical benefits of Goal-Orientated Measurement. In *Software Reliability and Metrics*: Elsevier Applied Science.

Rubin, H. (1987). *Critical success factors for measurement programs.* Paper presented at the Spring Conference of the International Function Point Users Group (IFPUG), Scottsdale, AZ, USA.

Rubin, H. (1991). Measuring Rigor and Putting Measurement into Action. *American Programmer, 4*(9), 9-23.

Russac, J. (2002). Chapter 18: IT Measurement: Practical Advice from the Experts - Cheaper, Better, Faster. In Addison-Wesley (Ed.), (pp. 147–158).

Robson, C. (2002). *Real World Research*: Blackwell.

Runeson, P., and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering. Springer.*

Schneidewind, N. F. (1992). Methodology for Validating Software Metrics. *IEEE Transactions on Software Engineering, 18*(5), 410-422.

Seaman, C. B. (1999). Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering, 25*(4), 557-572.

SEI. (1995). *The Capability Maturity Model: Guidelines for Improving the Software Process.*: Software Engineering Institute.

Selby, R. W., Porter, A. A., Schmidt, D. C., and Berney, J. (1991). Metric-Driven Analysis and Feedbback Systems for Enabling Empirically Guided Software Development. *International Conference on Software Engineering*, 288-298.

Selby, R. W. (2005). *Measurement-Driven Dashboards Enable Leading Indicators for Requirements and Design of Large-Scale Systems.* Paper presented at the 11th IEEE International Symposium on Software Metrics (METRICS 2005). 22-32.

Solingen, R. v., Berghout, E., and Kooiman, E. (1997). *Assessing Feedback of Measurement Data: Relating Schlumberger RPS practice to learning theory.* Paper presented at the 4th International Software Metrics Symposium (METRICS '97). 152-164.

Solingen, R. v., and Berghout, E. (1999). *The Goal/Question/Metric Method - A practical guide for Quality Improvement of Software Development*. London, England: Mc Graw Hill.

Solingen, R. v., and Berghout, E. (2001). *Integration Goal-Orientated Measurement Software Engineering: Industrial Experiences with and Additions to the Goal/Question/Metric Method (GQM).* Paper presented at the 7th International Software Metrics Symposium (METRICS'01). London, England. 246-258.

Solingen, R. v. (2002). The Goal/Question/Metric Approach. *Encyclopedia of Software Engineering, 2*, 578-583.

Sommerville, I., Sawyer, P., and Viller., S. (1999). Managing process inconsistency using viewpoints. *IEEE Transactions on Software Engineering, 25*(6), 784–799.

SWEBOK. (2008). *Chapter 12: Software Measurement*, from http://www.swebok.org/MeasurementKA-Draft-Feb2008.pdf

Tichy, W. F. (1998). Should computer scientists experiment more? *Computer, 31*(5), 32-40.

Team, C. P. (2002). *CMMI for Systems Engineering/Software Engineering, Version 1.1 - Staged Representation (CMU/SEI-2002-TR-002, ADA339224).* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Tesoriero, R., and Zelkowitz, M. V. (1997). *The Web Measurement Environment (WebME): A Tool for Combining and Modeling Distributed Data.* Paper presented at the Annual Software Engineering and Modeling Distributed Data.

Torii, K., Matsumoto, K., Nakakoji, K., Takada, Y., Takada, S., and Shima, K. (1999). Ginger2: An Environment for Computer-Aided Empirical Software Engineering. *IEEE Transactions of Software Engineering, 25*(4), 747-492.

Umarji, M., and Emurian, H. (2005). *Acceptance Issues in Metrics Program Implementation.* Paper presented at the 11th IEEE International Software Metrics Symposium (METRICS'05).

Weyuker, E. J. (1988). Evaluating Software Complexity Measures. *IEEE Transactions of Software Engineering, 14*(9), 1357-1365.

Wieczorek, I. (1997). On the establishment of successful measurement programs in industry. *Software Process: Improvement and Practice, 3*(3), 191–194.

Wiegers, K. E. (1997). Metrics: Ten traps to avoid. *Software Development Magazine, 5*(10).

Wiegers, K. E. (1999). A software metrics primer. *Software Developer Magazine, 5*(7).

Wadsworth, Y. (1998). What is participatory Action Research? *Action Research International, 2.*

Wallace, C., Cook, C., Summet, J., and Burnett, M. (2002). *Human Centric Computing Languages and Environments.* Paper presented at the Symposia on Human Centric Computing Languages and Environments. 63-65.

.Weber, C., and Layman, B. (2002). Measurement Maturity and the CMM: How measurement Practices Evolve as Processes Mature. *Software Quality Parctitioner, 4*(3).

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). Experimentation in Software Engineering - An Introduction. In *International Series in Software Engineering* (Vol. 6, pp. 228): Springer.

Wood-Harper, T. (1985). Research Methods in Information Systems: Using Action Research. In *Research Methods in Information Systems* (pp. 169-191). Amsterdam: North-Holland.

Yin, R. K. (2002). *Case Study Research: Design and Methods* (Vol. 5).

Zannier, C., Melnick, G., and Maurer, F. (2006). *Success of Empirical Studies.* Paper presented at the 28th International Conference on Software Engineering (ICSE'06). 341-350.