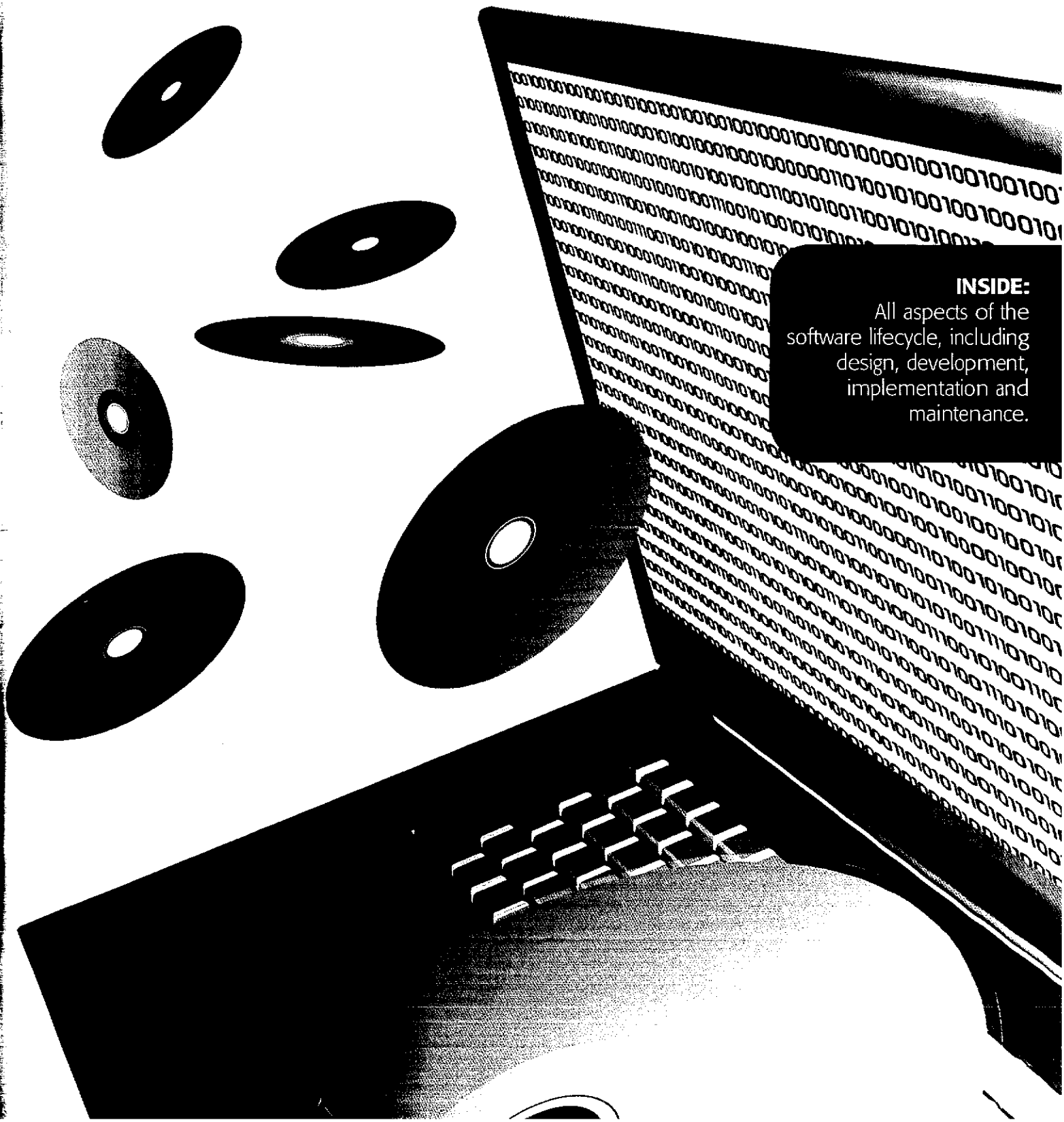
The Knowledge Network

# IET Software

Formerly IEE Proceedings Software



**INSIDE:**
All aspects of the software lifecycle, including design, development, implementation and maintenance.

# Instructions to contributors

## Aims and scope

*IET Software* is published bimonthly by the Institution of Engineering and Technology. All papers are subject to rigorous peer review and the decision to publish is based on the reviewers' comments. All papers should include a comprehensive, up-to-date literature review and the experimental results presented should demonstrate a significant advance over the current state of the art.

*IET Software* publishes papers on all aspects of the software lifecycle. The focus of the journal is on the methods used to develop and maintain software and their practical application. Authors are especially encouraged to submit papers on the following topics: software and systems requirements engineering; formal methods, design methods, practice and experience; software architecture, object orientation, reuse and re-engineering; testing, verification and validation techniques; software dependability and measurement; human systems engineering and human-computer interaction; knowledge engineering; expert and knowledge-based systems, intelligent agents; information systems engineering; application of software engineering in industry and commerce; software engineering technology transfer; management of software development; theoretical aspects of software development; education and training.

## Manuscript submission

All submissions to *IET Software* must be uploaded via our online submission system at:
**http://mc.manuscriptcentral.com/iet-sen**
Hard copy submissions cannot be accepted. Original papers should be submitted as a single file that includes all figures and tables. The following formats are acceptable for original papers: .pdf, .ps, .doc, and .rtf. If the paper is intended for a specific Special Issue, please make this clear when uploading your paper.

## Paper length

Original research papers submitted to *IET Software* should not generally exceed 4000 words, although some exceptions exist (for example, review papers). Please see our length policy document for further details at:
**http://www.ietdl.org/IET-SEN**
Please note that overlong papers will be rejected without review.

## Layout

Papers must be typed in a font size no smaller than 10 pt, and presented in single column format with double line spacing. All pages should be numbered. Authors should not copy the format of the published journal. All accepted papers will be copyedited and typeset according to the IET Research Journals house style.

## Language, spelling and grammar

All papers must be written in UK English. If English is not your first language, you should ask an English-speaking colleague to proofread your paper. Papers that fail to meet basic standards of literacy will be declined immediately.

## Abstract

This should be informative and suitable for direct inclusion in abstracting services as a self-contained article. The abstract must not exceed 250 words and should indicate the general scope and also state the main results obtained, methods used, the value of the work and the conclusions drawn. No figure numbers, table numbers, references or displayed mathematical expressions should be included.

## Figures and figure captions

Figures will be reproduced exactly as supplied with no redrawing or relabelling. It is therefore imperative that the supplied figures are of the highest possible quality. The preferred format is encapsulated postscript (.eps) for line figures and .tif for halftone figures with a minimum resolution of 300 dpi (dots per inch). High-resolution pdf files are also acceptable for figures and a list of figure captions should be provided in a separate section at the end of the manuscript. Each figure should be explicitly referred to in the text in numerical order. Colour should be used only where absolutely necessary for an understanding of the figure. We reserve the right to make a charge for colour printing in certain cases. Colour should not be used for distinguishing data in line diagrams.

## Tables

Tables should be explicitly referred to in the text in numerical order. Large tables should be avoided and split into smaller tables where possible.

## References

IET Research Journals use the Vancouver (numerical) system for references. You should number your references sequentially through the text, and each reference should be individually numbered and enclosed in square brackets (e.g. [1]).

## Copyright

Completed 'Publication Agreement and Assignment of Copyright' and 'Statement of Originality' forms are required for all papers. These should be emailed, posted or faxed to the Editorial Office (address given on the form) as soon as you have uploaded your paper. Further information is available at:
**http://www.ietdl.org/IET-SEN**

## Proof correction

Once your paper has been accepted and typeset, we will upload the page proof of your paper to our website and will send you an email notifying you that it is available for viewing. We ask that you return your corrections within three working days. Note that your paper will be published online in advance of printed publication and it is therefore in your interest to return your corrections to us as soon as possible.

## Complimentary journal copies

Once the paper has been printed, we will send the corresponding author a complimentary copy of the issue in which the paper appears.

# Contents

VOLUME 2 NUMBER 5 OCTOBER 2008

**IET Research Journals Dept.:**
Michael Faraday House, Six Hills Way, Stevenage, Herts, SG1 2AY, United Kingdom
Tel: +44 (0)1438 767667
Fax: +44 (0)1438 767317

For enquiries regarding advertising, commercial reprints, supplements and special issues, please contact Emma Patrick at: epatrick@theiet.org, +44 (0)1438 767414

## Submission Information

Papers should be submitted via our submission site at http://mc.manuscriptcentral.com/iet-sen and should be prepared in accordance with the Author Guide at www.ietdl.org/IET-SEN

## Subscription Information and Pricing

There are a number of ways to subscribe to *IET Software* in 2008, according to your requirements. Some of the most popular options are:

**Institutional subscription: £710/$1420**
**Personal subscription: £200/$400**
**IET member subscription: £75/$150**
**Reciprocal society member subscription: £100/$200**
**Single copy price: £119/$238**

Airmail supplement: £20/$40 per title
Online journals are subject to VAT where applicable

The complete range of subscription options, giving full details on how to purchase this and other IET products and services, as well as information on how to order a sample copy, are available at www.theiet.org/publishing/subscribe-buy

## IET Digital Library

IET Research Journals are available online through the IET Digital Library. Features include: "e-first" article publishing; browsable table of contents for all IET publications; cross-journal searching of bibliographic records (by title, keyword, issue date and author name); linking to Inspec abstracts of cited articles via CrossRef to cited full text or from an article to those that cite it; COUNTER-compliant usage statistics; table of contents alerting service, pay-per-view and document delivery.
For further details, please visit www.ietdl.org

## Copyright and Copying

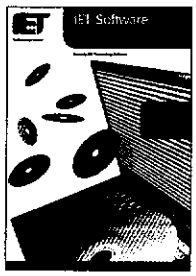


Typeset by: Techset Composition Ltd, Salisbury, United Kingdom

Printed on acid-free paper by: Henry Ling Ltd, Dorchester, United Kingdom

Product code: PJPGL805

Published in IET Software
Received on 7th March 2008
Revised on 20th June 2008
doi: 10.1049/iet-sen:20080026



ISSN 1751-8806

# Implementing a software measurement program in small and medium enterprises: a suitable framework

*M. Díaz-Ley*[1] *F. García*[2] *M. Piattini*[2]

[1]*Gaming Systems Development Department, Sistemas Técnicos de Loterías del Estado (STL), Madrid 28234, Spain*
[2]*Alarcos Research Group, Department of Information Technologies and Systems, Institute of Information Technologies and Systems, Escuela Superior de Informática, University of Castilla-La Mancha, Ciudad Real 13071, Spain*
*E-mail: maria.diaz.ley@gmail.com*

**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 their definition: the measurement program does not provide the required information, the data required are difficult to obtain and so on. The definition of measurement programs is even less encouraging in small and medium enterprises (SMEs), since most of the outstanding measurement program models are not suited to these types of companies. Additionally, the 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. A measurement program definition framework suited to SMEs called MIS-PyME is used to implement a measurement program in the software development and maintenance department of a medium company. The result of this measurement program and the good practices detected from this experience are expounded, along with the benefits obtained using MIS-PyME in contrast to our previous experiences of using the goal question indicator metric.

## 1 Introduction

Process measurement is of critical importance in controlling the development process in terms of performance, process improvement needs and monitoring current improvement initiatives [1]. It can also provide support for the planning, monitoring, control and evaluation of the software process [2]. A well-defined measurement definition approach can also provide support in locating and identifying the causes of problems [2].

In spite of the benefits 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 50–80% of measurement programs do not continue beyond their second year [3] and that two in three metric implementations fail [4], and these results are not encouraging. This tendency towards failure is particularly outstanding in the context of small and medium enterprises (SMEs) [5], the reason being that the factors, which characterise these companies, usually become the cause of the problem.

The features, which should comprise a measurement program model suited to SMEs, depend upon the enterprise's main restrictions. These are limited human resources, limited training and poor software measurement knowledge. An explanation of these features is shown in Table 1.

In many countries, most of the software development and maintenance companies are SMEs (77% in Germany in 2000; 69% in Brazil in 2001; 92% in Mexico in 2002), yet hardly any attention has been paid to this sector with regard to the area of software measurement.

**Table 1** Measurement program model features according to SME restrictions

| SME restrictions | Measurement program model features | Expected benefits |
|---|---|---|
| limited resources | a few people involved in the process (FPEOPL) | ensures 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 organisation which makes organisation-wide process improvement easier |
| | few but effective 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 |
| | specific guidelines to develop measurement goals and understand the benefits and potential for management and other guidelines (GGEN) | ensure users develop the measurement program correctly in order to attain the measurement goal and to learn about measurement analysis, interpretation, decision- making and its benefits prevent users from spending too much time on defining measurement programs (effort saving), and facilitate the measurement programs' reliability and usefulness |
| | specific guidelines to support basic process improvement (GPIN) | assess the measurement goals that should help to achieve the process improvement goals. This prevents users from spending too much time 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 needs of the software process |
| | specific guidelines to integrate measurement in the software processes (GINT) | ensure users learn about the benefits derived from its use are essential for the continuity of the measurement program |
| | specific guidelines to adapt the measurement definition to the measurement maturity of the company (GMM) | advise the user to implement those measurement goals which suit the company's measurement maturity and prevent the user from defining measurement goals which it cannot successfully attain (effort saving). The need to contract measurement experts to develop measurement programs is, thus, avoided increase the opportunities of measurement program success |
| | measurement examples (EXMP) | help users to understand the measurement program and its definition |

In accordance with the previously identified issues, this paper has two goals. The first aims to explain why the existing measurement methodology frameworks for defining and implementing measurement programs are not adapted to SMEs, and to propose and show in practice a measurement methodology framework, called MIS-PyME, which is adapted to them. The second goal aims to underline the measurement success factors, which are suited to SMEs and which were derived from our experience in defining a measurement program in a small software department of a medium company.

The article is organised as follows. Section 2 introduces MIS-PyME. Section 3 sets this work in context by using

the related literature to summarise relevant measurement methodology frameworks and their fitness to SMEs, and to show the known success factors for implementing measurement programs. Section 4 shows a case study in which MIS-PyME and goal question indicator metric (GQ(I)M) are used to define and implement the same measurement program. As a result, certain conclusions are derived regarding which methodology is better suited to SMEs. Some success factors derived from our experience of defining and implementing the measurement program in SMEs are also expounded. Conclusions and further research are outlined in Section 5.

# 2 MIS-PyME framework

MIS-PyME is a methodological framework focused on defining measurement programs based on software indicators in small and medium settings. It is focused on companies or units with the typical characteristics of SMEs with regard to measurement activities, namely:

- The people involved in the measurement program, including the measurement analyst, are from within the company and do not always have 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, and 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.

The MIS-PyME framework is composed of three main modules: the MIS-PyME methodology, the work products that support this methodology (MIS-PyME measurement goals table, MIS-PyME indicator template and MIS-PyME database) and the measurement maturity model (MIS-PyME measurement maturity model). In order to understand the MIS-PyME framework in a global manner, see Fig. 1.

## 2.1 MIS-PyME methodology and work products

The MIS-PyME methodology adapts GQM [6] and GQ(I)M [7, 8] in order to fit it to the SME characteristics, and includes the following work products:

*2.1.1 MIS-PyME measurement goals table:* The MIS-PyME framework proposes a set of the structured measurement goals usually required to achieve process improvement goals. The MIS-PyME measurement goals table is made up of ten process improvement goals and around 150 measurement goals. These goals are characterised and coloured depending on the maturity

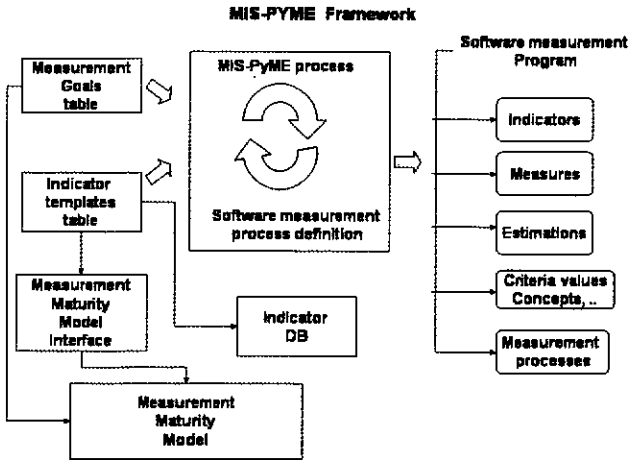



**Figure 1** *MIS-PyME framework*

required to implement them, and these colours are based on the MIS-PyME measurement maturity model (Fig. 2). Table 2 shows an excerpt of some examples of process goals and related measurement goals:

*2.1.2 MIS-PyME indicator templates:* An indicator template is defined for each measurement goal. The indicator template will guide users and help them to define indicators and measures for a specific measurement goal. Among other aspects, an indicator template permits the possibility of implementing the indicator with regard to the measurement maturity of the company, describes the conditions required to successfully implement the indicator with regard to previous indicators required and shows how to integrate this indicator into the software process. The typical questions, which the indicator attempts to answer, are proposed. The template also describes typical outcomes, their related analysis and possible interpretations. Table 3 shows an example of an indicator template and one of its applications, which has been implemented in a company and has been included in the MIS-PyME database.

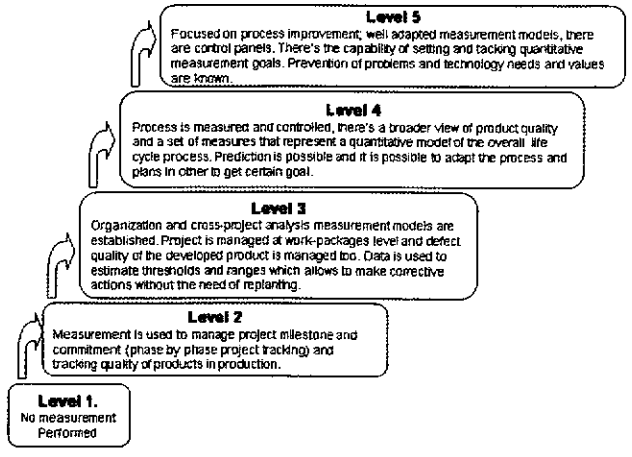



**Figure 2** *Capability levels of MIS-PyME measurement maturity model*

**Table 2** Excerpt of the MIS-PyME measurement goals table

| Process improvement goal | Entity | Measurement goal | Measurement goal |
|---|---|---|---|
| to improve the maintenance service | PROD | [level 4] to monitor the mean time needed to repair and take reactive actions (taking into account the product domain) | [level 4] to monitor the mean time for preventive maintenance |
| | | | [level 4] to monitor the mean time for adaptative maintenance |
| | | | [level 4] to monitor the mean time for corrective maintenance |
| | | | [level 4] to monitor the mean time for urgent corrective maintenance |
| to improve project tracing and oversight | PRJ | [level 2] to monitor the conformity of the project's main activities (or phase-by-phase) in contrast with what is planned in order to understand the progress of the project and take reactive actions if necessary | [level 2] to monitor the requirements that have passed the development process' main activities (analysis, design, construction, acceptance and so on.) |
| | | | [level 2] to monitor the conformity of the duration of the main activities in contrast with the planned activity duration |
| | | | [level 2] to monitor the conformity of real effort in contrast with planned effort per the main activity |
| | | | [level 2] to monitor the number of accepted open incidents during the software development activities in order to understand the remaining work |
| | | | [level 2] to monitor the number of accepted change requests opened during the software development activities in order to understand the remaining work |
| to improve the process | PROC | [level 3] to characterise process compliance within the project | [level 3] to characterise whether the project has performed each of the software process activities defined in the company's development process, and if negative, to understand its consequences |

*2.1.3 MIS-PyME database:* Each MIS-PyME indicator template contains a set of examples of real indicators, which have been defined in a successfully implemented measurement program. An indicator example included in the DB is illustrated in Table 3

Table 4 shows the steps in the MIS-PyME methodology.

## 2.2 MIS-PyME measurement maturity model

The MIS-PyME measurement maturity model has been designed to guide the user in defining the measurement program that is best suited to the maturity of the company. The measurement model defines a set of four themes (software management, quality and development capability, measurement scope, tools support and measurement support for management issues) and five maturity levels. Fig. 3 shows a scheme in which the meaning of each measurement level is represented. Each theme is focused on a measurement aspect upon which the success of the software measurement depends and contains the description of the conditions that are necessary to attain a certain maturity level (Table 5).

An interface with which to integrate the MIS-PyME measurement maturity model into the rest of the MIS-PyME framework has also been specified. This interface consists of three tables which help to define each of the fields of the indicator whose value depends on the measurement maturity level of certain themes.

These fields define the measurement goal of the indicator and are the following: purpose, entity and focus. There are,

Table 3 Example of a MIS-PyME indicator template and its application

| Example of indicator template | Indicator (DB) |
|---|---|
| *goal:* to evaluate the effectiveness of the tests performed in the project in order to improve future projects or the following phases of the project | *goal:* to evaluate the effectiveness of the integration and system tests performed in the project in order to improve future projects |
| *point of view:* top manager, project management, quality control manager | *point of view:* top manager, project management, quality control manager |
| *context:* tested software projects | *context:* tested software projects |
| *questions:* how many failures were found in the phase being evaluated (e.g. integration)? (categorised by its severity/priority) | *questions:* how many failures were found in the integration and system test phases? (categorised by its severity/priority) |
| how many failures were found in the next test phase? (categorised by its severity/priority) Do these failures exceed the threshold? Or do they exceed the threshold in comparison with failures found in the evaluated test phase?..... | how many failures were found in the acceptance test phase?<br>were 70% of the total failures found during the first test phases? Were 80% of total severe failures found in the first test phases? |
| *inputs:* failures found in test phases<br>*algorithms:* ..... | *inputs:* failures found in each test phase categorised by severity |
| *assumptions:* the number of failures – n°f (integration + system tests) > n°f (validation) > n°f (acceptance) | MED-PRJ-FallosAcept<br>MED-PRJ-FallosIntSys<br>..... |
| *recommended maturity:* maturity 4, see measurement maturity interface regarding 'purpose-evaluating', 'focus-process effectiveness', 'entity-process'<br>*integration:* when creating the end of project report, if the goal is to improve other projects. At the revision test phase if the goal is to improve current project<br>*measurement activity information:* .....<br>*analysis/interpretation:* .... | *activity information:*<br>• building the indicator responsible: measurement analyst<br>• analysis frequency: every 6 months....<br>*results location:* close project report of the project<br>*integration:* end of project activity.<br>analysis/interpretation .... |

consequently, three tables in the measurement maturity interface (Fig. 3). Each table poses questions to the user in order to determine whether the measurement maturity of the organisation is sufficient for that field and particular theme. Table 6 shows an outline of the interface table related to the 'purpose' field. For the 'purpose' field MIS-PyME suggests certain values: characterising (understanding), monitoring, evaluating, predicting and optimising. For each value and each theme, the table poses certain questions to the user in order to help him/her to decide whether the organisation is mature enough to define the desired value (e.g. evaluation) for the purpose field.

A more detailed description of MIS-PyME measurement maturity model is given in [9].

# 3 Related works: software measurement program models and success factors

This section expounds the related work found in the literature with regard to the two main focus areas of this paper. The outstanding software measurement methodologies are shown, their adaption to SMEs is questioned, and the factors for successfully implementing measurement programs are summarised.

## 3.1 Measurement methodology frameworks

Among the most representative methods and models for software measurement, the following deserve special attention:

• Goal question metric (GQM) methodology, a strategy that consists of deriving models and measures from measurement goals [6].

• Goal-driven software measurement guidebook [7], this provides an extension of GQM since it explicitly supports the definition of indicators [8]. This extension to GQM is named the goal question indicator metric GQ(I)M.

**Table 4** MIS-PyME methodology

| | |
|---|---|
| *Step 1: Identifying the process improvement goals:* defining the process improvement goals that you wish to carry out aided by software measurement. Identifying the related entities that will help achieve these goals | |
| *role:* top manager and measurement analyst | *input:* needs of the organisation if it is to establish and improve software processes<br>*output:* list of process improvement goals and related entities |
| *Step 2: formalising measurement goals and checking whether a measurement model is reused:* measurement goals are specified. The object of study, the purpose, the environment and the measurement constraints are then defined. The MIS-PyME measurement goal table helps the user to identify the measurement goals that will support the process improvement goals. In this step, we also check whether a measurement goal with the same description already exists in the organisation's measurement process. If in the affirmative, its corresponding indicator template is checked in order to understand whether it fulfils the needs to support the process improvement goal | |
| *role:* top manager and measurement analyst | *input:* MIS-PyME measurement goal table, list of process improvement goals and related entities, current company measurement process document<br>*output:* MIS-PyME measurement goal templates filled in (this is the header of the indicator template) / model reused |
| *Step A:* specifying project plan and brief lecture | a small project plan should be defined. It should contain only: A description of the measurement program goals, the benefits, the people involved and their roles, the calendar and a specification of the acceptance phase (trial analysis and pilot project). A brief lecture to the people involved in the project should be given, addressing the aforementioned issue |
| *role:* top manager | *input:* MIS-PyME measurement goal templates filled in<br>*output:* project plan |
| *Step 3: defining indicators:* the indicators required to implement measurement goals are defined | |
| top manager and measurement analyst | *input:* MIS-PYME indicator templates related to each measurement goal. MIS-PYME database<br>*output:* MIS-PYME indicator templates filled in |
| *Step 4: defining the measures and identifying the actions needed to implement them:* The measures that have to be collected are identified in detail and defined in the checklists. We define which data is to be included/excluded from the measured values, along with how the data will be collected. The ability of the organisation to obtain the measures is analysed, and the way in which they could be collected is established. If it is not possible to collect the desired data, this information may be used to modify the indicator specification | |
| *role:* top manager and measurement analyst | *input:* MIS-PyME indicator templates filled in<br>*output:* measure definition checklists and data collection specifications |
| *Step 5. integrating the measurement program:* the aim of this step is to integrate the measurement activities into previous measurement processes and into other software processes. MIS-PYME provides guidance through which to integrate the indicators and measurement sub-processes into the development, quality and management sub-processes of the company | |
| *role:* measurement analyst | *input:* MIS-PyME indicator templates, measurement definition checklists and collection specifications, current software development, quality and management process documentation<br>*output:* (updated) measurement process and (updated) software development, quality and management process documentation |
| *Step 6: verifying the measurement program:* the measurement process resulting from the process is reviewed and modified if necessary. A lecture is given to the reviewers in order to present the measurement program and its integration into the other software processes, including the changes to the existing measurement process. Once the reviwers have revised the measurement program, they are called to a meeting in which they expound the results of their review | |
| *role:* reviewer | *input:* measurement program specification, (updated) measurement process specification and (updated) software development, quality and management process documentation<br>*output:* verified measurement program specification and verified software processes |

*Continued*

**Table 4** *Continued*

| | |
|---|---|
| *Step 7: instrumentation:* tools which support measurement process are developed or tailored | |
| *role:* measurement analyst | *input:* measurement program<br>*output:* new or tailored measurement tools |
| *Step 8: acceptance of the measurement program:* the measurement program is used in a trial analysis or in a real project (selected as a pilot project). The measurement program is, thus, used as if it were already implemented. Reviewers 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 is given to the people involved in the measurement program, showing them how to use the tools to collect the data, the goals of those data, the measurement process and so on | |
| *role:* reviewer | *input:* measurement program<br>*output:* measurement program updated and the results of the acceptance analysis presented |

• ISO/IEC 15939 [10] identifies the activities and tasks necessary to successfully identify, define, select, apply and improve software measurement in a generic project or the measurement organisation structure. It also provides the common measurement terminology for the software industry.

• Practical software and systems measurement (PSM) [11], whose goal is to provide project and technical managers with the best practices and guidelines in software measurement.

To the best of our knowledge, very little research into this area (measurement models adapted to SMEs) exists in the literature. One related work, which we should outline in this context, is that of the GQM lightweight method defined by Gresse *et al.* [5]. This approach gives guidelines towards how to adapt GQM to SMEs and mentions integrating the reuse of context-specific quality and resource models into the GQM model. This feature is also addressed in Table 1.

Table 7 shows whether the aforementioned measurement methodological frameworks fulfil the characteristics indicated and enumerates the benefits of MIS-PyME with regard to its fitness in SMEs.

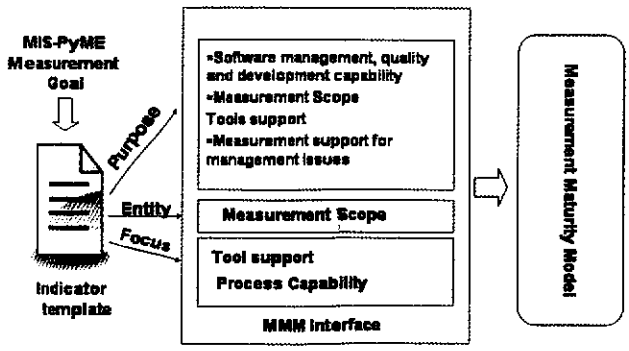



**Figure 3** *MIS-PyME measurement maturity scheme*

## 3.2 Software measurement program success factors

One measurement program success factor, which was identified by Gopal *et al.* [12], is that which causes effects on the use of metrics in decision-making and on the improvement of organisational performance. The success factors selected were divided into two sets: organisational and technical factors.

Daskalantonakis states some good practices obtained from his experience in Motorola [13, 14], and the experience of Hall and Fenton [15] experience identifies the 15 success factors for the implementation of software measurement programs. It is, however, uncertain whether any of these success factors could be applied to SMEs

In fact, none of these studies show especially good practices to follow when the measurement program is implemented in SMEs and the characteristics that are typical of these environments: low measurement maturity level, poor measurement knowledge, measurement not integrated into the company's culture and limited resources and budget.

# 4 Implementing a measurement program in an SME

Sistemas Técnicos de Loterías del Estado (STL) is a company which was created by the Spanish government and which provides the operations and IT development services for the national lottery. 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 that was implemented at that time.

*Measurement program goals:* In June 2006, the director of the development and maintenance department (which is made up of 39 people) became aware of the importance of measurement and seriously wished to deal with this area.

**Table 5** MIS-PyME measurement maturity model

| Themes | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 |
|---|---|---|---|---|---|
| software management, quality and development capability | immature processes. Projects depend on experienced professionals.. | repeat tasks which have been mastered in the past. Project depends on.... | projects characterised and reasonably understood. Project and development system... | measuring of process and process control. Focus on controlling the process... | optimised process. Focus on process improvement. Software process.... |
| measurement scope | carried out occasionally, either with experienced people or not at all | measurement is based on phase-by-phase project tracking,.... | organisation establishes standard processes and measurement models.... | measurement is used in most of the projects and products. Measurement models ... | well-adapted measurement models. Measuring overall process ... |
| tools support | there are no tools to explicitly support... | measurement support tools focused on projects. Some.. | project and product focus measurement tools. There is a measurement ... | measurement support tools focused on projects and products. ... | there are organisational tools which automatically collect.... |
| measurement support for management issues | management is not supported by measures... | basic project management. Milestones and commitment... | the product developed is controlled by means of measures... | it is possible to predict the product, service and other. ... | it is possible to predict and prevent problems... |

He particularly wished to achieve 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 in comparison with the plan (P.I.G 1.1) and to understand and manage any deviations from the plan at the project's closure (P.I.G 1.2).

- *P.I.G 2:* With regard to the process, he also wished to improve the conformance of the process with the test phases.

- *P.I.G 3:* To improve the project conformance. The aim of this goal is to improve the aspects of the project which affect the client the most.

- *P.I.G 4:* To improve the quality of products in production.

By using these process improvement goals as a starting point, the measurement program initiative began. This initiative had two goals:

- *Case study:* This study also aimed to understand whether the MIS-PyME methodology is more suitable than GQ(I)M [7] for defining (not implementing) measurement programs in SMEs.

- *Measurement program experience:* To develop a measurement program that satisfies these goals and which is accepted by the project managers.

## 4.1 Case study: MIS-PyME against GQ(I)M: suitability to SMEs

This section expounds our case study. The case study aims to define the measurement program using GQ(I)M [7] and MIS-PyME and to understand whether MIS-PyME is easier to use and more suiatable for SMEs than its base methodology, GQ(I)M.

The software measurement program development took place in two phases: in the first phase GQ(I)M [7] was used, but the results were not satisfactory. In the second phase, the MIS-PyME framework was successfully applied.

The same people were involved in both phases. The measurement analyst was from within the department, and her usual work consisted of consulting tasks, project coordination and requirement definition and testing. The top manager of the department was the promoter of the initiative. The reviewers were four people who usually play the role of project manager.

In both phases, the measurement program was defined by following each methodology as strictly as possible. In both

**Table 6** MIS-PyME measurement maturity model interface

| Themes | Characterise | Monitor | Evaluate | Predict | Optimise |
|---|---|---|---|---|---|
| proc. capability | has the company defined the attributes which are to be measured, including infor.?.. | (Level 3) have the attributes which are to be measured been included in company processes? ... | (Level 4) is attribute evaluation included in the process? ... | (Level 4) are processes stable enough to be performed rigorously and provide rigorously ... | (Level 5) is it possible to predict attributes in order to prevent problems ...? ... |
| measurement scope | (Level 1) N.A | (Level 2) is an estimating mechanism required to monitor actual against planned?... | (Level 4) does this attribute undergo frequent, rigorous and generalised ...? | (Level 4) has the measurement process been established organisationally?... | (Level 5) does the organisation have a performance control panel of ...? |
| Tool support | (Level 1) N.A | (Level 2) are there any tools which provide the required indicators to show project progress?... | (Level 4) is there an organisation database in which to store historical data? | (Level 4) is there an organisation database in which to store historical data?... | (Level 4) is there a management tool with which to dynamically and automatically?... |
| support for management | what is the intended use of the measurement? Is it meant to ... | (Level 2) is it intended to make reactive decisions based on the results of the measurement.. | (Level 4) does the organisation intend to make improvements as required to achieve its goal? | (Level 4) will prediction be used to improve project planning, avoid future problems, and so on?... | (Level 5) what is the intended use of measurement? Is it meant to be used to make dynamic... |

phases, the measurement program was defined by the measurement analyst and the top manager. The measurement analyst asked the top manager the necessary questions in order to detect goals, sub-goals and measurement goals. Once these goals were defined, the measurement analyst defined the indicators and measures that she required.

The project managers, top manager and measurement analyst evaluated and gave their impressions of the measurement program definition during the review/acceptance phase. The projects, which were used for the verification/acceptance of both measurement programs, were the software development projects that were closed in 2006. Some of these projects developed new software products, but most of them were high-impact maintenance projects.

*Phase 1:* Among the weaknesses and potential improvements, which were detected in Phase 1, are the following:

*Defining indicators according to maturity:* In spite of the time measurement analyst had spent attempting to understand the limitations of the purpose of the indicators, she failed to define some of them. Some examples are as follows:

- The indicator related to P.I.G 2 defined in Phase 1 was not focused on evaluating the conformance of projects with the test phases but on controlling the reliability of the product during the development phases in order to attain the reliability goal once the product was in production. Therefore the final indicator definition was not consistent with its initial aim and our measurement maturity was not sufficient for us to be able to evaluate the reliability progress of the product during the test phases in order to achieve a reliability goal in production. Additionally, this control was not well carried out since the frequency of data collection during the test phases was too low for this control to be effected.

- The indicator related to P.I.G 4 aimed to evaluate the reliability of the product based on a fixed goal that 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 which kind of fixed reliable goal was appropriate for the new products developed.

**Table 7** Measurement program methodologies and their fitness for use in SMEs

| Charac Method.SMEs | MIS-PyME | GQM [6] | GQ(I)M [7] | GQM lightweight | ISO/IEC 15939 [10] | PSM [11] |
|---|---|---|---|---|---|---|
| few people involved in the process (FPEOPL) | Yes | No | No | Yes | Average | Average |
| | in MIS-PyME methodology, the measurement program definition is carried out by two people: the measurement analyst and the top manager who is the promoter and the person most interested in the measurement initiative. The remaining stakeholders are involved in the revision and acceptance phases | 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 even less likely if the people who define the measurement program are from within the company | 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 | 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 organisation will be assigned on a part-time basis | with regard to ISO/IEC 15939 [10], in clause 5.1.2, it indicates the roles required to define a measurement program. However, these are not adapted to SMEs | 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 |
| measurement models (RUSE) | Yes | No | No | Yes | No | No |
| | step 2 of MIS-PyME methodology asks the user to reuse, if possible, a measurement model which has already been defined and implemented in the company. In addition, the methodology guides the user in defining reusable measurement programs. The measurement program includes the indicators, measurement and other elements and processes to put the measurement program into practice, but these elements are defined as self-correlated elements | GQM does not take into account the reuse of measurement programs that have already been defined in the organisation, and it does not define the measurement program for it to be easily resusable | it does not take into account the reuse of measurement models | 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 formalised, and this is called 'Reuse quality and resource models'. The second is performed at the end of the process and consists of packaging the context-specific measurement model for it to be reusable in the future | it does not take into acount the reuse of the measurement models | it does not take into acount the reuse of the measurement models. It does, however, propose it as a principle to be followed in Section 1.5 |

*Continued*

**Table 7** *Continued*

| Charac Method.SMEs | MIS-PyME | GQM [6] | GQ(I)M [7] | GQM lightweight | ISO/IEC 15939 [10] | PSM [11] |
|---|---|---|---|---|---|---|
| a few but effective steps (FSTEP) | Yes | Average | Average | Yes | Yes | Average |
| | MIS-PYME methodology is made up of nine steps which cover the whole definition measurement process to its implementation and are adapted to SMEs. The project plan and training are quite agile. With regard to the definition of the measurement program, only the top manager and the measurement analyst are involved, and the rest of the people involved play their roles in the verification and acceptance tests. The methodology also makes use of MIS-PyME work products that facilitate the definition of the measurement program | there are some activities which could be more agile such as measurement training and planning at the initial stage | GQ(I)M defines the steps with which to define and implement the measurement program. However, some steps related to instrumentation, reviews and acceptance are not included, but these are quite important, especially in an inexpert enviroment, which is usually the case in SMEs | in this method, less effort is assigned to measurement and training when compared with GQM, fewer people are involved in the pilot project, and the measurement program is mainly supported by one or two people who are convinced of its added value | the measurement process consists of four effective composite states. | the measurement process consists of four effective composite states. However, there are many parts that do not easily fit into SMEs: part 6- to implement the measurement process, parts 7.2 and 7.3-to evaluate the measurement process and so on |
| specific guidelines to support basic process improvement needs (GPIN) | Yes | No | No | No | No | Average |
| | the MIS-PyME measurement goals table provides the link between process improvement needs and measurement goals that support those needs | it bases the measurement program on the process improvement goals. However, it does not provide the common measurement goals that help to achieve the common process improvement goals | it does not provide the common measurement goals that help to achieve the common process improvement goals | this methodology does not complete GQM with regard to this kind of support guidelines | it does not provide the common measurement goals that help to achieve the common process improvement goals | 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 organisation-wide needs. Moreover, the model does not directly relate the process improvement goals and measurement needs |

*Continued*

**Table 7** *Continued*

| Charac Method.SMEs | MIS-PyME | GQM [6] | GQ(I)M [7] | GQM lightweight | ISO/IEC 15939 [10] | PSM [11] |
|---|---|---|---|---|---|---|
| specific guidelines to develop measurement goals and understand the benefits and potential for management and other guidelines (GGEN) | Yes | No | Average | No | No | Yes |
| | these guidelines are mostly provided by MIS-PYME indicator templates. Each indicator template provides a section called 'questions,' which suggests the analyses and questions, which can be answered by the indicator, and have been defined to achieve the measurement goal. The 'analysis and interpretation' section gives guidelines with regard to the type of analysis that can be performed on the indicator, along with its possible outcomes, interpretations and possible decisions, which show the potential of the use of measurement | it does not contain much information about possible questions that may be posed to achieve the measurement goal, the input measures, its analysis possibilities and the usefulness of these analyses | it incorporates clear examples to show how to use the methodology, but it does not provide complete guidelines to develop each common measurement goal and their benefits. | this methodology does not complete GQM with regard to this kind of support guidelines | it does not contain much information about possible questions that may be posed to achieve the measurement goal or the input measures. It analyses the possibilities and the usefulness of these analyses | 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 defined. |
| specific guidelines to integrate measurement into the software processes (GINT) | Yes | No | No | No | No | Yes |
| | MIS-PyME methodology dedicates one step to the integration of the measurement into the rest of the development process. Each indicator template also contains a field called 'integration,' which assists the user to integrate it into the measurement, development, quality and management process | it does not contain any information concerning the integration of measurement into the development, quality or management software processes | it does not contain any information concerning the integration of measurement into the development, quality or management software processes | this methodology does not complete GQM with regard to this kind of support guidelines | 5.3.1 indicates that measurement activities should be integrated into the other software processes. However, it does not guide how each common measure should be integrated | Part 2. 5.4 explains how measurement should be integrated into the technical and management processes. In a more practical manner, the measurement templates define how the measure is integrated and in which activities of the technical and management processes it could be used |

*Continued*

**Table 7** *Continued*

| Charac Method.SMEs | MIS-PyME | GQM [6] | GQ(I)M [7] | GQM lightweight | ISO/IEC 15939 [10] | PSM [11] |
|---|---|---|---|---|---|---|
| specific guidelines to adapt the measurement definition to the measurement maturity of the company (GMM) | Yes | No | No | No | No | Average |
| | the MIS-PyME indicator templates provides the user with the necessary information concerning what is required for the successful implementation of certain indicators, taking into account the company's maturity. Purpose, focus and entity are the indicator fields, which contain the maturity restriction information. The restrictions are related to the required level of the measurement maturity model | it does not take the company's maturity into account when the measurement program is being defined | it does not take the company's maturity into account when the measurement program is being defined | this methodology does not complete GQM with regard to this kind of support guidelines | it does not take the company's maturity into account when the measurement program is being defined | PSM addresses information regarding the constraints on the type of analysis (e.g. part 5) or on the tools required (e.g. part 2), and so on, but it does not relate these constraints to a measurement maturity model in which the limitations and evolution are clearly described, and it lacks information with regard to the constraints of purpose of some of the indicators and their scope |
| indicator examples (EXMP) | Yes | Average | Average | No | Average | Yes |
| | the MIS-PyME Database contains a real implemented indicator instance for each indicator template, which is useful for the user in that it facilitates the definition, checks its definition and improves it | it provides measurement program examples, but it does not provide common and useful measures or indicator examples | it provides clear examples but not complete information of common measurement goals, indicators, measures and so on | this methodology does not complete GQM with regard to this kind of support guidelines | it provides some examples of measurement programs and evaluation criteria, but not full common measurement examples | part 3 and 5 are specific chapters dedicated to showing and explaining common measures and indicator examples |

*Integrating the measurement program into the development, management and quality processes:* The measurement program was not intended to be applied to the software development processes existing at that time.

- P.I.G 2 required a definition of the thresholds of the reliability of the product 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 mature enough to do so.

- P.I.G 4 aimed to evaluate the reliability of the products based on a fixed goal that is negotiated with the client before the product is in production. However, this task was not contemplated in the project management and development processes.

*Documenting the measurement program:* 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 we defined the plan for implementing the measurement program in the project plan, and if we defined the measurement program in other formal documents or in the company's software measurement process documentation.

*Phase 2:* The measurement program defined in this phase using MIS-PyME was first more accurate and more understandable:

- The measurement program defined in Phase 2 was successfully integrated into the rest of the software processes without any great changes. It was also integrated into the follow-up project activity, the closing project activity and the Monday meetings.

- The measurement program defined in Phase 2 suited the measurement maturity of the company with one exception, which will be commented upon later.

- The measurement program defined was easy to understand and reuse since it separated the project plan, the elements of which formed the measurement program (measures, indicators and so on), and the measurement process.

However, some problems arose:

*Defining a suited purpose of the indicator:* One of the P.I.G 3 indicators, called 'ind-prj-fiabimp', was modified during the verification phase for it to be more coherent with the company's measurement maturity. Initially, this indicator evaluated the product's reliability based on a reliability goal (a number of failures registered in production once the product was installed). However, thanks to the MIS-PyME measurement maturity module support, we discovered that our measurement maturity was not sufficient to be able to evaluate reliability based on a fixed goal, and we considered that it would be better to evaluate the indicator by using a range of values (good, normal, not good, not acceptable) as a basis.

Other problems which arose were that it was still not possible to build the ind-prj-reqcod, ind-prj-reqverif and ind-prj-reqacept indicators because people were not experienced in using the IRQA tool, which was new, and they had therefore only mastered its basic use. They would have further complicated the process with the other attributes needed to build these indicators. Some training might have been useful with regard to this issue. However, it was decided to implement this part of the P.I.G 1.1 measurement program at a later stage. Another problem was that the density of failures during the project (Ind-prj-denfailures) measured the failures detected during the tests per day. We discovered that it would be better to measure this per week since it was more significant and the failures were not registered in the incident tool at the same instant at which they were detected.

*Results: MIS-PyME against GQ(I)M:* The reasons which could have led to an unsuccessful measurement program definition in Phase 1 when using GQ(I)M and to a successful measurement program when using MIS-PyME were indicated by the measurement analyst as follows:

*Defining measurement goals for process improvement goals:* when using GQ(I)M, it was difficult to specify the indicators that would fulfil the measurement goal and it was therefore easy to cause an error when defining the indicators that support the 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 certain kinds of guidelines such as those proposed by the MIS-PyME: measurement goals table and the guidelines provided in the MIS-PyME indicator templates.

*Adjusting the indicator to the company's measurement maturity:* As shown in the previous section, it was difficult to know which of the indicator's purposes would be possible to be implemented. We failed to define two indicators related to this issue in Phase 1. GQ(I)M does not give advice about the measurement maturity constraints of some indicators with regard to the indicator's purpose, entity and focus. For example, when the indicator's purpose is that of evaluation, it does not give recommendations about what is required if the indicator is to be implemented in a reliable fashion (e.g. understanding the characteristics of the entity, which affects the attribute to be measured in order to define a reliable goal based on these characteristics). MIS-PyME, however, provides the indicator templates that propose

information regarding the required maturity in order to implement each indicator. This information is based on the MIS-PyME measurement maturity model. In Phase 2, although we failed to define the ind-prj-fiabimp indicator, the MIS-PyME measurement maturity model helped us to verify this and to correct this problem.

*Integrating the measurement program into the rest of the software processes:* 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 Phase 1, 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 there is a dedicated task for it.

*Defining the measurement program in a reusable way:* When looking at the plan defined by using GQ(I)M, it was unclear how to document the measurement program and integrate it into other already established measurement processes in an easy and understandable fashion, how to establish the developed measurement program as a measurement process within the company or how to reuse the defined measurement program. 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 that makes its reuse simple. It separates the definition of 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 and guidelines), and the measurement process, taking into account the fact that a measurement process can implement more than one measurement programs or parts of different measurement programs.

*Some missing useful steps:* Another drawback of GQ(I)M is that it does not deal with the instrumentation verification and acceptance phase or who should be involved in these activities. MIS-PyME, however, does.

*Conclusion:* Problems detected 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 at their disposal, with which to define measurement programs, and since the people assigned may be from within the company, it seems that MIS-PyME is more suitable to 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 that enable the (probably) inexpert user to define the measurement program faster and in a more reliable way.

*Threats to validity:* The main aspect with regard to the validity of the study is that MIS-PyME was applied after GQ(I)M. Therefore the measurement analyst had already been trained in defining the measurement program and this might have led to a more competent definition of MIS-PyME. However, both methodologies were followed strictly and although the MIS-PyME application could have been affected by previous experience, MIS-PyME also provides its work products, which may have influenced the measurement analyst. The measurement analyst-based the measurement program on the information provided by the top manager and the work product and not on the measurement program defined by GQ(I)M. If GQ(I)M had been applied after MIS-PyME, the bias would have been greater since GQ(I)M does not provide work support products on which to base the measurement program.

In addition, the measurement analyst maintains that GQ(I)M lacks certain aspects of methodology (roles, verification activities, reuse of measurement programs) and that, thanks to the MIS-PyME support modules, it is more difficult to cause an error than when using GQ(I)M. Although this case study shows a verification of MIS-PyME, a formal validation should be carried out by independent sources and on more than one SME.

## 4.2 Measurement program experience

This section shows the measurement program which was defined in the second phase using MIS-PyME and the implementation phase of the measurement program. The resulting measurement program was made up of 31 indicators, 29 measures, 6 estimations, 9 concept criteria and 3 measurement sub-processes.

The definition of the measurement program took two months and 201 h of effort (Table 8).

**Table 8** Measurement program definition and implementation effort (Phase 2)

| Phase | Effort |
|---|---|
| definition of the measurement program and integration | ~156 h (6 h for the top manager and 150 for the measurement analyst) |
| verification and acceptance | 20 h (the trial test with the pilot project is not included −16 h for project managers and 4 h for the measurement analyst) |
| instrumentation | ~25 h (for measurement analyst) |

**Table 9** Description of indicators of the software measurement program

| Indicator name | Description |
|---|---|
| *P.I.G 1.1:* to improve the project monitoring in contrast with the plan | |
| ind-prj-effortconformace: | monitors the conformance of real effort in comparison with the planned effort |
| ind-prj-prodcod | monitors the project's progress during the coding phase |
| ind-prj-progverif | monitors the project's progress during the verification phase |
| ind-progacept | monitors the project's progress during the acceptance phase |
| *P.I.G 1.2:* to understand and manage deviations from the plan to the project closure | |
| ind-prj-inexaceffort | characterises the exactitude of the effort planned for the project |
| ind-prj-inexacsize | characterises the exactitude of the project's planned size |
| ind-prj-inexaccost | characterises the exactitude of the project's planned cost |
| ind-prj-inexacduration | characterises the exactitude of the project's planned duration |
| ind-prjorg-inexaceffort, ind-prjorg-inexacsize, ind-prjorg-inexaccost, ind-prjorg-inexacduration | these indicators analyse the general results in the development unit by making six-monthly cross-project analyses |
| *P.I.G 2:* to improve the conformance of the process with the test phases | |
| ind-proc-testconformance | evaluates the relation of failures detected in the project's test phases |
| ind-procorg-testconformance: evaluates the relation of failures detected in the department's projects during the test phases | this makes a six-monthly cross-project balance of the above indicator |
| *P.I.G 3:* to improve the project conformance | |
| ind-prj-prjconformance: evaluates conformance with the project | evaluates the conformance with the project developed: this indicator was simply formed from two other indicators, ind-prj-fiabimpl and ind-prj-inexacduration. ind-prj-fiabimpl aimed to evaluate the reliability of the software developed and released under a project. ind-prj-inexacduration aimed to evaluate the deviation in time with regard to what was planned |
| ind-prjorg-prjconformance: | this makes a six-monthly cross-project balance of the above indicator and compares the results with previous periods |
| *P.I.G 4:* to improve the quality of products in production | |
| ind-prodorg-fiabcrit, ind-prodorg-fiabsoport, ind-prodorg-fiabinf | these indicators show the progress of the density of failures in production for each product (number of failures of the product per month). The data required to build this indicator is collected every 6 months. Each indicator includes the products related to its product classification: critical, support, informative products |

The details of the measurement definition for the process improvement goal 1 are shown in Table 9. These indicators were chosen by applying MIS-PyME and its support module: the MIS-PyME measurement goals table. In addition, the indicators were defined by using their corresponding templates that were provided by the MIS-PyME measurement indicator templates.

*Software measurement process:* The measurement program defined in the previous step established the initial measurement process for the development department in STL. Three different sub-processes were identified:

- *The project management measurement sub-process:* The indicators involved in this sub-process are those related to P.I.G 1.1 (to improve the project monitoring in contrast with the plan), P.I.G 1.2 (to understand and manage deviations from the plan to project closure), P.I.G 2 (to improve the conformance of the process with the test phases) and P.I.G 3 (to improve the project's conformance.). The indicators analysed in this process are applied to a specific project and are, consequently, called ind-prj-XXX. This sub-process is carried out for each of the department's projects. In addition, these indicators were integrated in the follow-up activity, and in both the closing project activity of the project management process and its corresponding reports.

- The organisational management measurement sub-process includes the indicators related to P.I.G 1.2 (to understand and manage deviations from the plan to project closure), P.I.G 2 (to improve the conformance of the process with the test phases) and P.I.G 3 (to improve the project conformance.), but is specifically that which concerns all of the department's projects, that is, those which involve a balance of six-monthly cross-projects. These indicators are called ind-prjorg-XXX and were integrated into the usual organisational meetings in the following manner: every 6 months the measurement analyst had to collect the required information with which to build these indicators. This information was then given to the top manager. The top manager, along with the project managers and the quality control manager, analysed the results and suggested improvements which could be made to the management, quality, development and measurement processes during the Monday meetings. These improvements were included in a report produced by the measurement analyst.

- The product management measurement sub-process, which defines the information related to the collection, analysis/interpretation and action plan tasks of the indicators related to P.I.G 4. This measurement process was integrated into the usual organisational meetings in the same manner as the process indicated above.

*Instrumentation:* Tools, which were already in use in the department, were used to collect all the required data. The measurement program's intention was 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 by the company.

- *ActiTime:* A free tool which registers the effort dedicated to each task.

- *Remedy:* An incident management tool. Remedy needed to be tailored for it to be able to automatically launch the queries to the database and to create reports containing the required information.

- *IRQA:* A requirement management tool which was also tailored to allow it to trace the state of each requirement (coding phase, verification phase, acceptance phase and so on).

*Implementation and an example of the results of one indicator from the measurement program:* The measurement program passed the acceptance step, as is indicated in Phase 2 of Section 4.1, and it was successfully implemented. This section shows an example of the results of the ind-procorg-testconformance indicator. The analysis of this indicator is derived from the data collected during the acceptance phase of this measurement program and those collected 6 months later, once the measurement program had been successfully implemented. This indicator was analysed in eight projects (Table 10). The results show that only half of these projects

**Table 10** Example of ind-procorg-testconformance results

| Analysis date | Prj | Relation total | Relation grv | Conform_total | Conform_grv |
|---|---|---|---|---|---|
| 31/12/2006 | PR1 | 8636/1364% | 8333/1667% | yes | yes |
| 31/12/2006 | PR2 | 5000/5000% | 5000/5000% | no | no |
| 31/12/2006 | PR3 | 000/10 000% | 000/10 000% | yes | yes |
| 31/12/2006 | PR4 | 4737/5263% | 4706/5294% | no | no |
| 31/12/2006 | PR5 | 000/000% | 000/000% | yes | yes |
| 01/07/2007 | PR6 | 7273/2727% | 7143/2857% | yes | yes |
| 01/07/2007 | PR7 | 5000/5000% | 6000/4000% | no | no |
| 01/07/2007 | PR8 | 6066/3934% | 6538/3462% | no | no |

achieved their goals: the total relation of failures detected between the integration and acceptance test phases should be at least 70%/30%, and 80%/20% for the severe failures.

### 4.3 General good practices for measurement program implementation in SMEs

This experience led to the production of Table 11, which shows 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 [13–15] have already identified the practices suggested below as being success factors in the implementation of measurement programs. In accordance with these practices, we propose others that make the definition and implementation of measurement programs easier with regard to the restrictions of SMEs.

**Table 11** Success factors for defining and implementing measurement programs in SMEs

| Success factors | Description |
|---|---|
| reusing measurement models and defining measurement programs for the organisation's use | the definition of the measurement program should focus on defining the measurement program in a reusable way and reusing its existing measurement programs |
| measurement for supporting process improvement goals and not business goals | attempt to define measurement programs that focus on supporting software process improvement goals rather than business goals. Low measurement maturity level settings cannot afford measurement programs from any business goal if the aim is to define successful measurement programs in an effortless, accurate and consistent manner |
| 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 a few people | define a draft of the measurement program involving a few people: the main top manager (the main stakeholder) and the measurement analyst. This manner 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 mature enough, 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 (with regard to data collection) data |
| make the measurement analyst aware of any changes in the organisation | the measurement analyst must be at least concerned with the new tools that are to be acquired and could substitute old ones. 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. He should, moreover, be aware of any changes in the quality, management or development management processes |

# 5 Conclusions and further research

This article describes the MIS-PyMe framework and its application in a case study. MIS-PyME provides SMEs with the methodological guide necessary to define measurement programs based on indicators, which support process improvement goals and which are in accordance with their maturity in software measurement. MIS-PyME expects to fulfil the requirements of a software measurement program definition model suited to SMEs, as the most well-known approaches, such as GQM, GQ(I)M, ISO/ IEC 15939 and PSM, were not developed with this objective in mind and require further adaptation to fulfil the restrictions of SMEs.

A case study has been performed in a medium company in order to understand whether MIS-PyME was better adapted to SMEs than its base model GQ(I)M, The experience of implementing the measurement program defined with MIS-PyME and the success factors derived from the experience have also been expounded.

As a result of the case study, some significant conclusions have been reached. First, the MIS-PyME suits SMEs better than GQ(I)M. The measurement program defined by following GQ(I)M was not successfully carried out, mainly because: some indicators did not fit the measurement maturity of the company; another set of indicators was not able to be integrated into the rest of the software processes; and the measurement program was not documented in an understandable fashion. The definition and implementation of the same measurement program in a second phase by applying MIS-PyME provided some significant benefits, including the facts that the managers were happy with the measurement program and that it was successful. As a result, 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.

Finally, the lessons learned from the experience have led us to suggest a set of good practices for the definition and implementation of a measurement program in non-mature SMEs. It is important to note that these practices may not be valid when: the measurement program is not intended to be implemented in an SME; the measurement maturity level is high; the software measurement is integrated into the culture of the organisation; or there are no limitations with regard to the budget and resources for the measurement program, which would, for example, allow the company to contract external consultants/experts in measurement program implantation.

In future work, our main interest will be focused on refining the MIS-PyME framework by defining and implementing measurement programs in other SMEs and on using related experience to incorporate some improvements into the MIS-PyME work products and MIS-PyME methodology where necessary. Among the future issues to be considered, the incorporation of guidelines to help SMEs to apply statistical software techniques [16, 17], according to their measurement maturity level, will be analysed.

# 6 Acknowledgment

We would like to thank the staff of Sistemas Técnicos de Loterías del Estado (STL) for their collaboration. This research has been sponsored by the COMPETISOFT (CYTED, 506AC0287), ESFINGE (Dirección General de Investigación del Ministerio de Educación y Ciencia, TIN2006-15175-C05-05) and INGENIO (Junta de Comunidades de Castilla-La Mancha, PAC08-0154-9262) projects.

# 7 References

[1] LAVAZZA L., BARRESI G.: 'Automated support for process-aware definition and execution of the measurement plans'. Proc. 27th Int. Conf. Software Engineering, St. Louis, MO, USA, 2005, pp. 234–243

[2] BRIAND L.C., MORASCA S., BASILI V.R.: 'An operational process for goal-driven definition of meassures', *IEEE Trans. Softw. Eng.*, 2002, **28**, (12), pp. 1106–1125

[3] RUBIN H.: 'Measuring rigor and putting measurement into action', *Am. Program.*, 1991, **4**, (9), pp. 9–23

[4] PFLEEGER S.L.: 'Understanding and improving technology transfer in software engineering', *Syst. Softw.*, 1999, **47**, (2, 3), pp. 111–124

[5] GRESSE C., PUNTER T., ANACLETO A.: 'Software measurement for small and medium enterprises'. 7th Int. Conf. Empirical Assessment in Software Engineering (EASE), Keele, UK, 2003

[6] SOLINGEN R.V., BERGHOUT E.: 'The goal/question/metric method – a practical guide for quality improvement of software development' (Mc Graw Hill, London, England, 1999)

[7] PARK R.E., GOETHERT W.B., FLORAC W.A.: 'Goal-driven software measurement – a guidebook' (Software Engineering Institute, Carnegie Mellon university, Pittsburgh, 1996)

[8] GOETHERT W., SIVIY J.: 'Applications of the indicator template for measurement and analysis'. Software Engineering Measurement and Analysis Initiative, CMU/SEI, September 2004

[9] DIAZ-LEY M., GARCÍA F., PIATTINI M.: 'MIS-PyME software measurement maturity model – supporting the definition of software measurement programs'. Product focused software development and process improvement (PROFES'08), 5089/2008, Frascati, Monteporzio Catone, Rome, Italy, 2008

[10] ISO/IEC 15939: 'Software engineering – software measurement process', 2002

[11] PSM : 'Practical software and systems measurement – a foundation for objective project management Version 4.0c'. Deptartment of Defense and US Army, November 2000

[12] GOPAL A., KRISHNAN M.S., MUKHOPADHYAY T., GOLDENSON D.R.: 'Measurement programs in software development: determinants of success', *IEEE Trans. Softw. Eng.*, 2002, **28**, (9), pp. 863–875

[13] DASKALANTONAKIS M.K.: 'A practical view of software measurement and implementation experiences within Motorola', *IEEE Trans. Softw. Eng.*, 1992, **18**, (11), pp. 998–1010

[14] DASKALANTONAKIS M.K., YACOBELLIS R.H., BASILI V.R.: 'A method for assessing software measurement technology', *Qual. Eng.*, 1990, **3**, (1), pp. 27–40

[15] HALL T., FENTON N.: 'Implementing effective software metrics programs', *IEEE Softw.*, 1997, **14**, (2), pp. 55–65

[16] CAIVANO D.: 'Continuous software process improvement through statistical process control'. 9th European Conf. Software Maintenance and Reengineering (CSMR 2005), Manchester, UK, 21–23 March 2005, pp. 288–293

[17] BALDASSARRE M.T., BOFFOLI N., CAIVANO D., VISAGGIO G.: 'Improving dynamic calibration through statistical process control'. 21st IEEE Int. Conf. Software Maintenance (ICSM 2005), Budapest, Hungary, 25–30 September 2005, pp. 273–282