IV SIMPOSIO INTERNACIONAL DE SISTEMAS DE INFORMACIÓN E INGENIERÍA DE SOFTWARE EN LA SOCIEDAD DEL CONOCIMIENTO

SISOFT 2006

Entidades Organizadoras

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS (COLOMBIA)

UNIVERSIDAD PONTIFICIA DE SALAMANCA (ESPAÑA)

FUNDACIÓN PABLO VI (ESPAÑA)

LIBRO DE ACTAS

VOLUMEN I

Agosto 23 al 25 de 2006
Cartagena de Indias - Colombia

Editado por :

Luis Joyanes Aguilar
Universidad Pontifica de Salamanca

www.sisoftw.com

ISBN: 84-690-0258-9

ENTIDADES COORGANIZADORAS:

Colciencias(Colombia)

Universidad de Cartagena (Colombia)

Asociación Colombiana de Informatica(Colombia)

InstitutoTecnologico de las Americas(Repu. Dominicana)

Pontificia Universidad del Peru(Peru)

ENTIDADES COLABORADORAS:

Rama Estudiantil IEEE-UD (Colombia)

Universidad de Oviedo (España)

Centro de Tecnologia Avanzada(Mexico)

Universidad Catolica de Colombia(Colombia)

Universidad Privana Antenor de Orrego(Peru)

Centro de Supercomputación de Barcelona(España)

Tompkins Cortand Community College(EEUU)

Tecnologico de Monterrey(Mexico)

Universidad Autonoma de Yucatan (Mexico)

Universidad de las Palmas de Gran Canaria(España)

Universidad Apec (Rep Dominicana)

Escuela Superior Politecnica del litoral(Ecuador)

Universidad Autonoma de Occidente(Colombia)

Universidad Catolica de Montevideo (Uruguay)

Universidad Nacional de Nordestes(Argentina)

ENTIDADES PATROCINADORAS

Avianca (Colombia)

Banco Occidente (Colombia)

Famisanar (Colombia)

Universia

Trotamundos (Colombia)

Empresa de Telefonos de Bogota(Colombia)

Microsoft

Fundación de Educación Superior San José(Colombia)

IV Simposio Internacional de Sistemas de Información e Ingeniería de Software en la Sociedad del Conocimiento

SISOFT2006

VOLUMEN I

Cartagena de Indias, Colombia
23, 24 y 25 de Agosto de 2005

www.sisoftw.com

Editor
Luís Joyanes Aguilar

UNIVERSIDAD PONTIFICIA DE SALAMANCA
(*CAMPUS* MADRID)

FUNDACIÓN PABLO VI

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS

PONTIFICIA UNIVERSIDAD CATÓLICA DE PERÚ

IV SIMPOSIO INTERNACIONAL DE SISTEMAS DE INFORMACIÓN E INGENIERIA DE SOFTWARE EN LA SOCIEDAD DEL CONOCIMIENTO

ISBN: 84-690-0258-9

Editor:
Luís Joyanes Aguilar (FI/UPSAM, España)

Adjuntos al Editor:
Víctor Hugo Medina García (FI/UDFJC, Colombia)
Javier Parra Fuente (FI/UPSAM, España)
Oscar Sanjuán Martinez (FI/UPSAM, España)
Daniel Zapico Palacio (FI/UPSAM, España)
Juan Manuel Lombardo Enríquez (FI/UPSAM, España)

Edita:
Universidad Pontifica de Salamanca campus de Madrid (España)
Fundación Paulo VI, Madrid (España)
Universidad Distrital Francisco José de Caldas, Bogotá (Colombia)

IMPRESO EN BOGOTÁ (COLOMBIA)

DIRECCIÓN DEL SIMPOSIO

- **Dr. Luis Joyanes Aguilar** (Presidente) (UPSAM, España)
- **Dr. Víctor Hugo Medina García** (Director Ejecutivo) (U. Distrital F.J.C., Colombia)

COMITÉ DE PROGRAMA PERMANENTE (STEERING COMMITEE)

- **Dr. Luis Joyanes Aguilar** (Presidente) (UPSAM, España)
- **Dr. Victor Hugo Medina García** (Director Ejecutivo) (UDFJC, Colombia)
- **Msc. José Armando Tabares** (ITLA, Rep. Dominicana)
- **Dr. Juán Manuel Cueva** (U. Oviedo, España)
- **Dr. Maynard Kong** (PUCP, Perú)
- **Dr. Javier Parra Fuente** (UPSAM, España)

COMITÉ DE HONOR

- **Excmo. y Sr. Magnífico D. Marceliano Arranz Rodrigo** (Rector UPSA -España)
- **Excmo. y Rvdo. Sr. D. Fernando Sebastián Aguilar** (Presidente Fundación Pablo VI - España)
- **Excmo. y Rvdo. Sr. D. Ángel Berna Quintana** (Director General Fundación Pablo VI - España)
- **Excmo. Sr. Magnífico D. Ricardo García Duarte** (Rector UDFJC - Colombia)
- **Excmo. Sr. Magnífico D. Rafael Peña Suesca** (Decano Facultad de Ingeniería UDFJC - Colombia)
- **Excmo. Sr. Magnífico D. Germán Sierra Anaya** (Rector Universidad de Cartagena - Colombia)
- **Don Luis Eduardo Garzón** (Alcalde Mayor de Bogotá - Colombia)

COMITÉ DE ORGANIZACIÓN (EUROPEO)

- **Dr. Juán Manuel Lombardo** (Coordinador General) (UPSAM, España)
- **Dr. Javier Parra** (Org. Técnica) (UPSAM, España)
- **Dr. Oscar Sanjuán Martinez** (Org. Técnica) (UPSAM, España)
- **DEA. Daniel Zapico Palacio** (Org. Técnica) (UPSAM, España)
- **Lic. Teresa Díez** (UPSAM, España)
- **Raquel Ureña** (UPSAM, España)

COMITÉ DE ORGANIZACIÓN (COLOMBIANO)

- **Dr. Víctor Hugo Medina García** (Coordinador) (UDFJC)
- **Dr. Jairo Torres Acosta** (UDFJC)
- **Dr. Nelson Pérez Castillo** (UDFJC)
- **Ing. Msc. Anselmo Vega Vega** (UDFJC)
- **Ing. Msc. Edmundo Vega Osorio** (UDFJC)
- **Ing. Giovany Tarazona Bermúdez** (UDFJC)
- **Ing. Beatriz Jaramillo** (UDFJC)
- **Ing. Alexis Ortiz Morales** (UDFJC)
- **Ing. Carlos Andrés Torres** (UDFJC)
- **Ing. Carlos Eduardo Vargas** (UDFJC)
- **Ing. Jesús Guzmán** (UDFJC)
- **Ing. Diana Ahumada** (UDFJC)

COMITÉ DE CO-ORGANIZACIÓN (COLOMBIANO)

- **Ing. Msc. Alfonso Pérez Gama** (ACCIO)
- **Ing. Msc. Pedro Guardela** (U. Cartagena)
- **Ing. Msc. Julio César Rodríguez Ribón** (U. Cartagena)
- **Ing. Msc. David Antonio Franco Borré** (U. Cartagena)
- **Ing. Msc. Miguel Ángel García Bolaños** (U. Cartagena)
- **Ing. Msc. Isaac Zúñiga Silgado** (U. Tecnológica de Bolívar)
- **Ing. Msc. Juán Antonio Contreras** (Corporación Universitaria Rafael Núñez)

COMITÉ DE ORGANIZACIÓN (DOMINICANO)

- **Ing. José Armando Tabares** (Director Ejecutivo, ITLA)
- **Lic. Quinta Ana Pérez** (Secretaría técnica, ITLA)

COMITÉ DE ORGANIZACIÓN (PERUANO)

- **José Antonio Pow-Sang** (PUCP)
- **Ing. Abraham Dávila** (PUCP)

COMITÉ INTERNACIONAL DE APOYO

- **MSc. Oscar Mendoza Macías** (ESPOL, Ecuador)
- **MSc. Jorge Huayahuaca** (TC3, EE.UU.)
- **MSc. Ernesto Ocampo** (U. Católica de Uruguay)
- **MSc. Juán Estanis** (Argentina)
- **Dr. Arturo Fernández** (TEC de Monterrey, México)
- **Ing. Jorge Torres** (TEC de Monterrey, México)
- **MSc. Inmaculada Madero** (UNAPEC, República Dominicana)
- **MSc. Miguel Cid** (INTEC, República Dominicana)
- **MSc. Luz Mayela Ramirez** (U. Católica de Colombia)
- **MSc. Jesús Cardona** (U.A. de Cali, Colombia)
- **MSc. Elmer González (**UPAO, Perú)
- **Ing. José Fernando Baquero Herrero** (CIATEQ, México)

COMITÉ DE PROGRAMA

- **Dr. Luís Joyanes Aguilar** (Presidente) (UPSAM, España)
- **Dr. Álvaro Suárez** (ULPGC, España)
- **Dr. Angel Egido** (U. Catholique d´Angers, Francia)
- **Dr. Ángel Losada** (UPSA, España)
- **Dr. Arturo Ribagorda** (UC3M, España)
- **Dr. Darío Álvarez (**UNIOVI, España)
- **Dr. David Olivieri** (UV, España)
- **MSc. Esteban Bolondi** (U. Católica de Colombia)
- **Dr. Francisco Rubio** (ULPGC, España)
- **Dr. Gustavo Rossi** (U de la Plata, Argentina)
- **Dr. Javier Bustamante** (UCM, España)
- **Dr. Javier Segov**ia (UPM, España)
- **Dr. Javier Nó** (UPSA, España)
- **Dr. Jean Pierre Boutinet** (U. Catholique d´Angers,Francia)
- **Dr. Jesús de la Cruz Escotto,** (USAC, Guatemala)
- **Dr. José R. García-Bermejo** (USA, España)
- **Dr. José Ramón Pin** (IESE- Madrid, España)
- **Dr. Juan Manuel Corchado** (USA, España)
- **Dr. L. Alfonso Ureña López** (UJAEN, España)
- **Dr. Luis Rodríguez Baena** (UPSAM, España)
- **Dr. Manuel Maceiras** (CISTEC, España)
- **Dra. Matilde Fernández** (UPSAM, España)

CONTENIDO

PRÓLOGO

Una breve reseña histórica de SISOFT

En Agosto de 2001, las universidades Pontificia de Salamanca (*campus* Madrid, España) y la Distrital Francisco José de Caldas (Bogotá, Colombia) junto con la Fundación Pablo VI (Madrid, España) organizaron el primer SISOFT en la ciudad de Bogotá. En el mes de Agosto de 2006 se cumplen cinco años, 3 ediciones realizadas y la 4ª edición que ahora celebramos. Bogotá, Lima, Santo Domingo y Cartagena de Indias son las cuatro ciudades donde se ha organizado y celebrado el Simposio de Ingeniería de Software y Sistemas de Información. En estos cinco años, la Ingeniería de Software, los Sistemas de Información y la ya madura Sociedad de la Información y la naciente Sociedad del Conocimiento, han evolucionado considerablemente en paralelo con las grandes innovaciones tecnológicas. SISOFT ha tratado de adaptarse a todos estos grandes cambios y prueba de ello son los dos volúmenes publicados del actual Libro de Actas.

Cuando en el año 2000, las universidades fundadoras decidieron poner en marcha un congreso internacional que pudiera servir de puente entre Europa y Latinoamérica, y en particular entre España y Colombia como países organizadores, en áreas de futuro en el mundo del software, se pensó que dada la dificultad y el gran reto que suponía su organización, deberíamos de organizarlo con espíritu de continuidad y con una periodicidad bianual. Así y tras el éxito del primer simposio, en Agosto del 2001, la Pontificia Universidad Católica de Perú en Lima, asumió el nuevo reto de organización del II Simposio que se celebró en Agosto de 2003, ya con una gran expansión en las zonas geográficas de Latinoamérica y Europa; en 2005, el Instituto Tecnológico de las Américas (ITLA) de Santo Domingo en la República Dominicana, asumió la organización de la tercera edición, donde ya su carácter internacional quedó totalmente asentado, con la incorporación de ponentes y conferenciantes de numerosos países americanos y europeos. Esta circunstancia llevó al Comité Fundador ha pensar en el cambio de periodicidad y de este forma se ha pasado del carácter bianual al carácter anual, y la celebración en 2006 en la histórica, universitaria y hermosa ciudad de Cartagena de Indias, en el Caribe colombiano. El generoso apoyo de las universidades locales hizo que las actuales cinco entidades internacionales, que forman el Comité Organizador apostaran decididamente por una nueva organización que permitiera mostrar durante sus jornadas, los actuales cambios producidos en la media década pasada en las áreas citadas y en los temas fundamentales del simposio.

La Universidad Distrital Francisco José de Caldas con la Universidad Pontificia de Salamanca han asumido de nuevo el gran reto que suponía la periodicidad anual con un programa del Simposio que contemple los ámbitos científicos, académicos, profesionales, económicos junto con el estudio de las innovaciones tecnológicas que ya han impactado en la primera mitad de esta década, y sin duda ya están afectando y afectarán, al menos, a la segunda mitad, también, de esta década.

El Comité Fundacional desea agradecer el esfuerzo científico e investigador de la Universidad Distrital, las universidades locales de Cartagena, que han apoyado el Simposio, y en general de las numerosas universidades colombianas, restantes latinoamericanas y europeas que lo apoyan de una u otra forma. De igual manera desea agradecer al resto de instituciones, organizaciones y empresas que de una u otra forma colaboran en el éxito del Simposio.

La cuarta edición de SISOFT presenta las innovaciones tecnológicas y el estado actual de la investigación en las áreas de *Ingeniería de Informática* e *Ingeniería de Sistemas*, y en particular en Ingeniería de Software y en Sistemas de Información Las comunicaciones aceptadas, junto con las conferencias magistrales, plenarias, talleres, seminarios, mesas redondas (paneles), mostrarán durante los tres días del Simposio, los temas más relevantes y candentes presentados por investigadores, docentes, profesionales, estudiantes de doctorado y maestría, etc,. y una muestra del estado del arte en los temas centrales del mismo.

El contenido

Los dos volúmenes publicados recogen las casi ochenta comunicaciones aceptadas –de las más de cien presentadas- y que serán defendidas durante las sesiones del Simposio. Los artículos que se presentan en ambos volúmenes y las conferencias plenarias y magistrales que se impartirán tratarán sobre temas tan variados e innovadores como los siguientes:

- Nuevas metodologías de Ingeniería de Software y Sistemas de Información
- Arquitecturas *hardware*
- Supercomputadores
- Sistemas de Información Geográfica (GIS)
- Metodologías de Gestión de Proyectos como ITIL, PMP,.
- Gestión de calidad y métricas de software
- Gestión de la Información y el Conocimiento en Ciencias de la Salud, Bioinformática y Biotecnología
- Interacción Persona-Máquina y Usabilidad
- Ingeniería Web (Plataformas, Aplicaciones y Servicios Web)
- Estado actual de la industria del Software (software libre, software bajo demanda, videojuegos, …)
- Estado del arte en Ingeniería de Software
- Web 2.0 (tecnologías, aplicaciones, Ajax, RSS,…)
- Redes Sociales Virtuales
- Agentes y Multiagentes
- Responsabilidad Social Corporativa: en sus enfoques empresarial y tecnológico
- Ingeniería multimedia y Realidad Virtual
- Gestión del Conocimiento e Inteligencia de Negocios
- Estado actual de la Sociedad de la Información y el Conocimiento
- Educación virtual, a distancia: *e-learning*
- Innovaciones tecnológicas en Tecnologías de la Información y las Comunicaciones (TIC)
- Almacenamiento virtual y en red
- Software avanzado de Microsoft (C#, …)
- Innovaciones y mejoras de lenguajes de programación (Java, C++, C#, JavaScript, XML, Ajax,…)
- …

Solo resta comentar que las conferencias, seminarios, talleres, mesas redondas, y conclusiones más sobresalientes se publicarán en un tercer volumen que verán la luz a la terminación del Simposio.

Los diferentes comités de SISOFT 2006 agradecen de nuevo a todas las instituciones, empresas, conferenciantes, ponentes, congresistas, … su participación y quedan a su disposición durante todo el evento y los convocan para SISOFT 2007, pendiente de confirmación del país organizador, sede y fecha, pero que con toda seguridad seguirá la tradición científica e investigadora de las cuatro primeras ediciones

En Cartagena de Indias, agosto de 2006

Fdo: Dr. Luis Joyanes Aguilar

Presidente de SISOFT

Defining, Performing and Maintaining Software Measurement Programs: State of the Art

María Díaz
Sistemas Técnicos de Loterías del Estado
Gaming Systems Development Department
Madrid, Spain, 28234
Maria.diaz@stl.es

Félix García, Mario Piattini
University of Castilla-La Mancha, Information Systems and Technologies Department,
Ciudad Real, Spain, 13071
{Felix.Garcia, Mario.Piattini}@uclm.es

ABSTRACT

This paper analyses different approaches found in literature about how to carry out a measurement program. Hence techniques about planning and defining measurement processes, collecting, analysing, performing feedback sessions and packing the measurement program results are presented in this paper. Finally measurement improvement models are also exposed. These studies have been organized following and extending the software measurement classification proposed by Alain Abran et al.[1] for the Software Engineering Body of Knowledge (SWEBOK). Specifically we have focussed on the measurement process sub-classification. The aim is to present the techniques existing in literature regarding this measurement program area and to set up the initial basis in order to define a methodology framework which supports Small and Medium Enterprises (SMEs) in performing measurement programs according to their specific needs, maturity and limitations.

Keywords: Measurement program, measurement process, state of art, SWEBOK, SMEs.

RESUMEN

Este artículo analiza las diferentes aproximaciones de la literatura sobre cómo llevar a cabo un programa de medición. Por lo tanto, en este artículo se muestran las técnicas de planificación, definición, recolección, análisis, realimentación y almacenamiento de los resultados de la medición del software, además de modelos de mejora de medición. Para la clasificación de los temas de interés de la bibliografía en relación a la medición del software este estudio basa y extiende la clasificación de medición del software propuesta por Alain Abran et al.[1] para Software Engineering Body of Knowledge (SWEBOK). Específicamente este trabajo está enfocado en la sub-clasificación del proceso de medición. El objetivo es presentar las técnicas existentes en la literatura sobre el área de los procesos de medición y establecer la base inicial para definir un marco metodológico que de soporte a las PYMES en la implementación de programas de medición del software de manera que se adecue a sus necesidades específicas, su madurez y sus limitaciones.

Palabras claves: Programa de medición, proceso de medición, estado del arte, SWEBOK, PYME.

1. INTRODUCTION

Most of software organizations are conscious of the importance of software measurement and they have started working on this line. However most organizations do not achieve the initial expected results. There are several reasons why this situation is quite common. First, software development projects are restricted by tight schedule goals. Hence allocating considerable extra effort to provide measurement data to back up decision-making is not desired. As a result, there is a very high risk to get no data at all, or to get incomplete or inaccurate data. Secondly, measurement must be popular with all measurement stakeholders and must be integrated in the culture of the organization to succeed. Another reason is that some of stakeholders judge measurement program as being too exhaustive compared to their benefits. Finally, many organizations and projects possess insufficient or poorly organized data collection and analysis mechanisms which result in limited, inaccurate, or out of time feedback to managers and developers.

All these inconveniencies are especially outstanding in small and medium enterprises (SMEs) where generally there are not enough resources to promote serious measurement program initiatives, training is more difficult and managers do not see the need of a measurement framework as they think that they can control projects more easily without using measures. In this context, usually there is not a measurement culture and then there is not historical data which is necessary when defining and performing a measurement process. Finally, the existing methods and frameworks that support measurement programs such as Goal Question Metric (GQM)[2], Goal-Driven Software Measurement[3], GQ(I)M [3, 4] , PSM[5] and ISO/IEC 15939 [6] do not fully satisfy the needs of medium and small companies. Sometime they are difficult to adjust to the organization and to implement with scarce budget. SMEs require a better adaptation of these models, better focused on its small dimensions in departments, resources, projects, objectives etc. This kind of companies require measurement programs which are easy to understand and implement as they usually cannot spend quite a high amount of resources in carrying out these programs.

The target of this work is to provide an overview about the models and techniques existing in literature about measurement programs. Namely we focus on the business and process assessment, plan definition, performance and evaluation of measurement processes. This work will serve as the basis of our research whose goal is to tailor existing measurement programs models to the specific characteristics of SMEs.

This paper is organised as follows: In section 2 the classification, which provides us with the basis to organize the studies exposed in this paper, is shown. In section 3 we briefly describe a set of representative models and standards regarding the measurement process. In sections 4, 5, 6 and 7 the approaches found in literature regarding the main phases of the software measurement process will be shown. Finally, in section 8 we present the conclusions and we outline the lack of practices to support the special characteristics of SMEs regarding measurement programs which will be the basis of the future research presented.

2. SOFTWARE MEASUREMENT CLASSIFICATION

The classification of the literature about software measurement programs presented in this paper is based on the software measurement classification defined by Alain Abran et al. [1]. They proposed a new Software Measurement KA (Key Area) for SWEBOK (Software Engineering Body of Knowledge) in order to consider Software Measurement as a key area itself in the world of Software Engineering. In addition, we have proposed other sub-classifications which most of them were adapted from the Software Engineering Management KA and Process Measurement KA of the SWEBOK in order to provide a clearer overview of the current state of art. These are the following (see Figure 1):

- “Process and Business Assessment”: We have defined this new category under the Measurement Process area. We missed this sub-classification, as before defining a measurement program, we have to know what the company needs.
- “Perform Measurement Process”, we have sub-classified this section of the Alain Abran et al.’s classification as the following: “Collecting data”, ”Analysis”, “Feedback” and “Packing”.
- “Evaluate Measurement”, we have sub-classified this section of the Alain Abran et al.’s classification as follows: “Evaluate measurement products”, “Evaluate measurement process” and “Measurement Process Improvement
- Finally we propose another sub-classification, which is not covered in this paper. This one has been included with the “Tools and Techniques” classification and is called “Experience Factory”. The experiences of implementing

measurement programs in organizations, such as the methodologies used, the challenges to implement them and their results, should be addressed in this category.

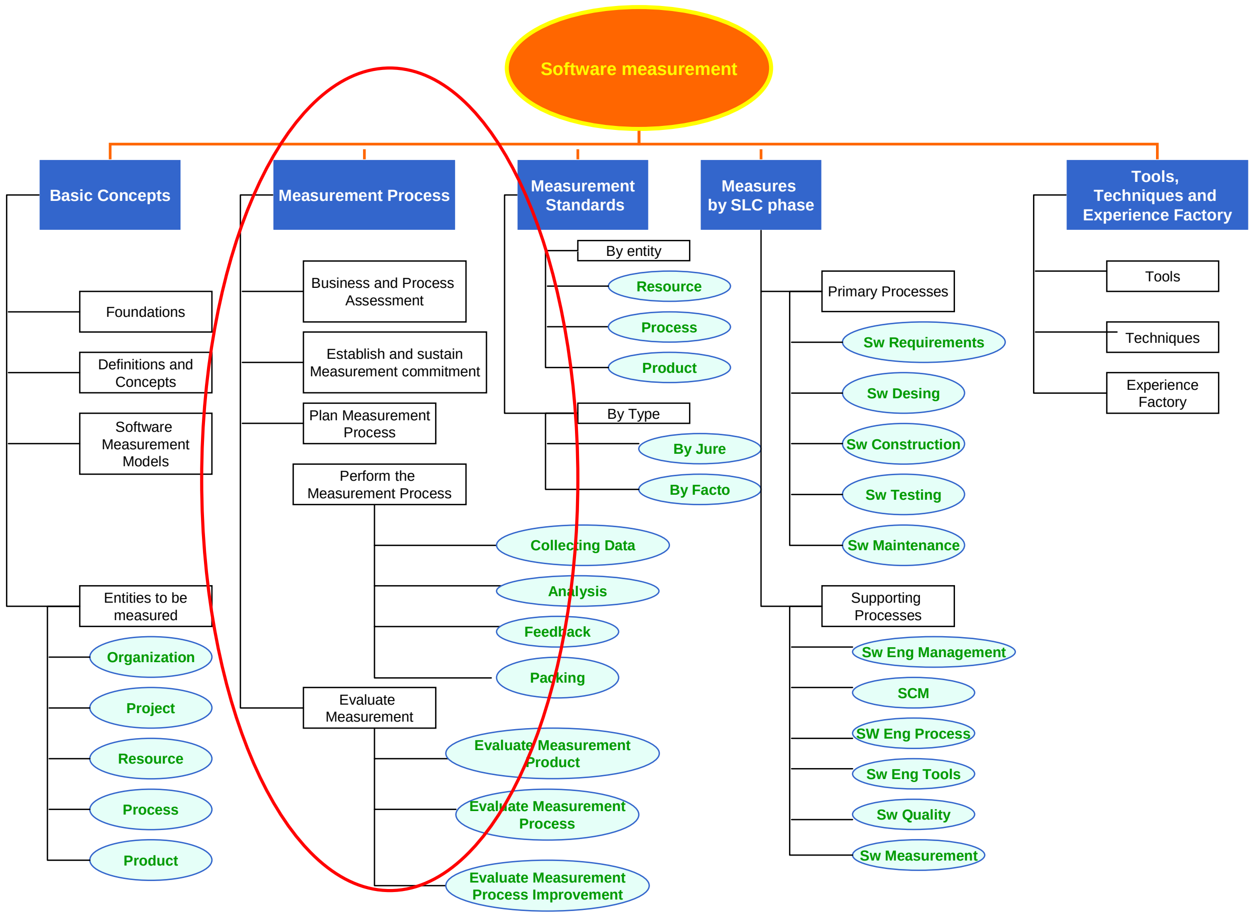

Figure 1: Software measurement classification adapted from [1].

3. BASIC CONCEPTS – MEASUREMENT MODELS

We start with Goal Question Metric (GQM) methodology whose aim is to define the metrics that should be collected depending on the objectives. The GQM process is described as goals generate questions, and questions generate metrics. GQM method was first defined by Basili and Weiss in 1984 and it was extended by Rombach in 1990. GQM model defines guidelines about defining the projects plan for the measurement program, defining the context and objectives, defining the measurement plan, planning the analysis, collecting data, interpreting results and detecting improvements.

The SEI in 1996 [3]published the Goal-Driven Software Measurement guidebook where the measurement practices are explained by means of examples. The guidebook is based on ten organized steps included in tree main groups and it provides an extension of GQM by means of useful templates and checklist in order to define goals, indicators, measures, data collected, indicator's representation, analysis sheets, etc. This extension to GQM is named Goal Question Indicator Metric GQ(I)M which identifies and defines the software metrics that support the organizations business, process improvement and project targets. GQ(I)M sets the link between the objective and the collected data. Another difference between GQ(I)M and GQM is that GQ(I)M explicitly defines and supports the indicators. The key instrument of this methodology is the indicator template which is used to precisely define the indicator's information as follows: "who", "what", "where", "when", "why" and "how". It also covers the required measures needed to build the indicator and the required elements which help the interpretation of the indicator [4].

PSM [5] is a framework created by the Department of Defense in 1994 and its 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 planning, measurement relation, measurement evaluation and establishing and maintaining the compromise.

ISO/IEC 15939[6] identifies the activities and the required tasks in order to successfully identify, define, select, apply and improve the software measurement under a generic project or the measurement organization structure. It also provides the common measurement glossary 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 administrate the processes and to objectively demonstrate the product quality.

The IEEE Std 1061-1998 is a methodology for software quality measures which defines the system software quality by means of a list of the required software quality attributes for the system. The software measures purpose is to evaluate the quality requirements along the project life cycle.

4. PROCESS AND BUSINESS ASSESSMENT

Sometimes the target of a measurement program such as organization's needs comes from the results of a software process assessment. There are some well known frameworks which help to perform the software process assessment. Some of them, called benchmarking, consist of assessing the maturity of software processes in organizations. On the other hand there are analytical techniques which also support the software process assessment.

The benchmarking frameworks consist of comparing the current organization practices to the organizations best practices. Some of these frameworks are the SEI SW-CMM model, its successor CMMI, SPICE Software Process Improvement and Capability d'Etermination which has turned into the standard ISO-15504, ISO 90003:2004, or Bootstrap.

However some companies are not culturally or economically prepared to perform these extensive assessments. Thus some software process assessment models suitable for small companies have appeared: Johnson and Brodman proposed a model called LOGOS Tailored CMM which tailors CMM, P-CMM (the People CMM), PSP (Personal Software Process), TSP (Team Software Process) to small companies. Sanders developed the SPIRE Handbook whose aim is to help small software development units to achieve business benefits from their investment in software process improvement (SPI) using the structure of the ISO TR 15504 (SPICE) as a framework for practical advice. Additionally Cater-Steel [7]published an assessment framework for small enterprises. The framework was based on SPICE and it was sponsored by the Software Engineering Australia (Qld) (The framework, called RAPID, was prepared to perform the assessment in one day thus rather than using the 40 processes defined in ISO 15504, eight key processes were selected.

The MoProSoft [8] Software Process Model defines the best practices of software administration and engineering and it is focused on SMEs. It conforms to ISO 90003:2004 and CMM V1.1. MoProSoft was developed to be set as a Software Process State Mexican Norm in order to support SMEs in their software process improvement and to make them more competitive in the national and international market. COMPETISOFT is a project led by universities, companies and government agencies of Latin American countries whose aim is to develop a software process model for SMEs and also to set up this model as a Software Process Norm for Latin American countries. The software process model proposed in COMPETISOFT project is based on the MoProSoft Software Process Model. We are directly cooperating in the COMPETISOFT project and then our research is included in the framework of this project.

Finally we present the analytical techniques which can be exemplified by the Quality Improvement Paradigm (QIP). Some of these techniques that also support software process assessment are the following: process simulations as explained in [9] or other kinds of experiments as shown in [10]. Process definition review also allows detecting the process deficiencies and potential process improvements. Some examples are exposed in [11] and [12], and an observational study is exposed in [13]. Other techniques involve the tracking of the detected faults cause as it is shown in [14] and [15]. In addition, there is the technique which consists of performing statistical analysis to control the process evolution [16] [17]. Finally other process assessment techniques are the Personal Software Process [18]where the assessment is performed according to a "bottom-up" approach, as improvements are derived from data interpretations of individual’s development practices.

5. PLAN MEASUREMENT PROCESS

GQM covers the plan measurement process phase defining the practices as the following[2]: Firstly the planning phase, where the project’s plan for the measurement program is defined. Secondly the definition phase, where the measurement program is defined using a template that helps to specify the initial goal thus the study object, purpose, quality focus, point of view and context are specified. Afterwards, questions, metrics and hypothesis are defined. The metrics derived in the GQM plan have to be integrated in a measurement plan thus for each metric, the measurement plan answers the questions how, by whom, and when the data are to be collected. Finally, an analysis plan must be also defined based on the GQM plan where some analyses of possible results are simulated.

PSM [5] also specifies the practices of the measurement plan phase. Hence, the measures that provide the projects with the information view required are defined. Therefore, detecting the needs of the beneficiaries who are in charge of the decision making, link the information needs with the entities that could be measured, selecting and specifying the base measures on the organizational projects and processes are the main tasks defined in this phase.

ISO15939[6]identifies the following steps when the measurement process is planned: Characterize the organizational unit, Identify information needs, Select measures, Define data collection, analysis, and reporting procedures, Define criteria for evaluating the information products, Review, approve, and provide resources for measurement tasks and Acquire and Deploy Supporting Technologies.

In literature we can find diverse proposals regarding the measurement process definition. These approaches can complement the measurement process planning section of the models and standards presented previously and tackle the following representative topics:

Specifying the measurement program from the high level business perspective: Niessink and Vliet [19] proposed a general model which highlights that the importance of the measurement program is the return value which it provides to the organizations. Unlike the GQM model [2], the objective of the measurement program is not a measurement target but a high level business target. Niessink and Vliet’s model generally specifies the measurement program from the high level business perspective. Therefore the model defines, in four main steps, how the measurement program is implemented to give support to a certain company’s problem. In addition, the success factors highlighted in the model are those which return value to the organization and are external to the measurement program.

This idea of planning of the measurement process from a high level business perspective has been also underlined by the GQ(I)M model and also by Offen and Jeffery who proposed a framework that extends the Quality Improvement Paradigm/GQM named M3P (Model Measure Manage Paradigm). The aim was setting well defined links between the results of the metrics and the business context and development.

Adding formality to measurement process plan definition: The studies above, show the perspective from which the measurement process should be planned but they do not show how to actually perform a measurement program. Regarding this issue:

Briand et al. [20] proposal aims to properly define measurement processes, especially prediction systems by means of defining an approach of six organized steps based on GQM but adding formality to it. The aim is to add theoretical basis when defining measures as usually it is not only impossible to create an accurate prediction system but also to create a system that accurately characterize the object of study. The main additions of this approach are the following:

- The attributes derived from the GQM hypothesis should be formalized by specifying its properties which constrain and guide the search for new measures for the attribute and accept existing measures that adequate to that attribute. In addition abstractions are identified thus the entity to measure may be mapped into one or more abstractions (mathematical representation of the entity such a graph) so it becomes analyzable and its relevant context attributes become quantifiable. The metrics selected are validated as they must comply against the attribute’s generic and context-dependent properties.
- Hypothesis are refined and statistically checked in order to verify the credibility of the hypothesis and to create the predicted model. Hence one needs to validate a statistical relationship between independent and dependent measures.

We continue showing complementary techniques regarding how to define a measurement process plan. We then show a technique that allows to easily and more accurately define the measurement process and to add formality to the measurement process plan. It consists of the idea that modelling the entity to be measured and integrating it in the measurement process benefits the measurement plan definition.

Brijckers et al. [21] indicated that this idea helps to precisely identify the object to be measured, which is especially important when GQM interviews are performed. It also helps identifying the appropriate person who will collect each data required and when it should be collected. It also supports the identification of the data which could be reused for different goals. Finally data required to define a measurement program will be directly derived from the software process model.

Lavazza and Barresi [22] also supported this idea and they go further by implementing a tool which integrates the GQM plan and the model of the entity to be measured. They claim that the benefit of this idea, besides what was indicated by Brijckers et al.[21], was the formality that it provides to the measurement process. Hence if the entity to be measured is not modelled, then it is difficult to assure that the collected data match the plan as this knowledge relies on the people who performed the measurement plan. In addition it helps to identify which data are impossible to collect. Repeatability of the measurement process is also more feasible. Finally there are other advantages such as the user can define some metrics and questions which where not taken into account previously since there are some elements in the model that are natural candidate metrics. Additionally using this approach the GQM plan will be packed together with the process model being measured making it more precise.

Simplifying the measurement process plan complexity: Regarding this topic, Baldassarre et al. [23] define a framework called 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 especially regarding the interpretation of the GQM measurement plan. They underline 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 also will reduce the overall complexity of the interpretation. This framework is useful to evaluate the comprehensibility of the measurement process, as if it is difficult to understand only the person who has developed the measurement process will be able to understand the process and interpret results; and the ease of use it which is another measurement program success factor. Both factors make the measurement process more repeatable.

Measurement process plan definition pattern: Lindvall et al. [24] defined a pattern to represent the measurement plan definition which also complements GQM. The main advantage of the proposal is that measurement patterns provide the benefit of avoiding starting a measurement program from scratch and allowing identifying a high level pattern which can be applied in a different context. In addition it allows the measurement definition to easily pack and transfer within and between organizations. Measurement patterns are built by making a tree where GQM hypothesis and questions from higher-level goals lead to lower level goals. Therefore, other related hypothesis and questions can also lead to more specific goals until it gets to the lowest level, where goals are basic goals (characterization goals) and they lead directly to metrics. The result is a tree where the GQM definition plan is detailed.

Measurement program social impact: Sometimes it could be worthy to analyse whether the measurement program will be used and well adapted in the organization. Regarding this, the Social Impact model defined by Umarji and Emurian [25] predicts the success and thus the usage of a measurement program by means of measuring the intention to use it. The intention on the other hand depends on the Ease of Use, Usefulness, Control and Attitude to use the measurement program.

The measurement process planning techniques and approaches presented in this section complement the models and standards shown. This set of techniques and approaches adds elements of improvement to the standards and models as it gives to the measurement process plan definition the business perspective, it also adds formality in the definition of the basis of the measurement process (measures and hypothesis) which enhance the chance of success and provides techniques which make the measurement process easier to define, use, reuse and understand. Finally, it also addresses a model to predict the social impact of the measurement program in the organization.

6. PERFORM MEASUREMENT PROCESS

In this section a brief introduction regarding collecting data, analysis, feedback and packing will be exposed and also some approaches from literature that adds value to this area will be indicated.

Collecting data

ISO/IEC 15939[6] 5.3.2 specifies that data shall be collected and stored including any context information necessary to verify, understand or evaluate the data.

GQM method [2] indicates the steps required for collecting data which are classified in two main subgroups: Training and data collection starting, and the construction of a Measurement Support System (MSS). The steps defined for the first group are: "Hold Trial" phase which is a preliminary step before starting collecting the actual data. In this step the collection procedures, tools and templates are defined and tested. The second phase is "kick off" where all the measurement program participants must be present. The target of this phase is that all participants agree on the data collection activities and to inform about the procedure, tools and templates. Data collection step aims at colleting data, feeling in the templates and deliver them to the GQM team who will check the data validation, and pack the templates for future use. The second group aims at defining the set of statistics tools, spread calculation sheets, data base systems and presentation sheets.

PSM [5] part 3 defines the data collection phase in three groups: Collect Data, Verify Data and Normalized Data. At the Collect Data phase, the frequency of the collection and how to report the data to the measurement analyst team is treated. In the Data Verify section a check point list, which helps to verify that data match the measurement specifications, its accuracy etc., is defined. In the last section, Normalized Data, the measures are converted into a common unit for example, translating effort from hours to months.

Analysis

In this phase the measurement results are represented in graphs or other formats which help to make decisions. The initial conclusions should be defined and checked to assure that they are accurate and meaningful and that they result in reasonable actions. This part is supported in ISO/IEC 15939[6] 5.3.3.

PSM [5] part 4 classifies the analysis types in three categories: estimation, feasibility analysis and performance. In addition it presents a relational model where issues areas and entities which have to be taken into account for the analysis are related and then its dependencies are shown. Finally it details the data, models and tools required to perform each of the analysis mentioned.

As an example of a performance analysis experience it is underlined the work of Ramil et al. [17] who performed sequential statistical tests using CUSUM Change Point detection algorithm in order to control the evolution in productivity of the VME kernel.

Feedback

In this phase information products must be documented and communicated to users and stakeholders ISO/IEC 15939[6] 5.4.1.

GQM method [2] includes feedback in the interpretation phase and four phases regarding interpretation are defined: Preparing the feedback sessions, performing the feedback sessions, reporting measurement analysis results, and an analysis of the benefits and costs of the measurement program is also suggested to be performed.

PSM [5] part 4 suggested that the reporting activity should be integrated in the day-to-day management and technical processes and it should include: Overall evaluation of the projects, identifications of specific problems, risks, and lack of information; recommendations and potential new issues.

Van Solingen et al.[26] proposed that two additions had to be included in the GQM model. The first addition refers to establishing conditions which are necessary to facilitate learning in software measurement programs. Therefore it is required that the measurement program explicitly establishes conditions that are necessary to facilitate learning in software measurement programs and the second one refers to explicitly identify whether the measurement program was actually worthwhile. They proposed a model of the learning process between software development team and GQM team based on learning theories and specified a list of learning enablers in order to stimulate the group and the learning sessions. On the other hand, in order to address the costs of a measurement program it is proposed to follow the Birk et al. method in which the GQM steps are used for addressing costs. Thus for each stage defined in the GQM model, the activities related to that stage are identified and afterwards the cost of each activity is calculated. Finally they propose the following points to address benefits: goal attainment, understanding of software product and process, improved communication, attention for software quality and corporate identity.

Packing

This phase refers to keeping the measurement program plan, data collected, analysis sheets, cost-benefits analysis and conclusion and action plans of the measurement in a feasible way. Notice that none of the measurement process models shown above treats packing as an independent stage. In this phase it is important to storage data in a reusable way thus allowing the ease of reusing the information for future projects and allowing the information transference. In addition a framework or data mining tool to easily access to the information stored should also be defined. Koennecker et al. [27] shows the experience of implementing an Experience Factory in an Australian organization. In addition Lindvall et al. [24]also addresses the idea of ease of storage and transference when defining the GQM pattern.

7. MEASUREMENT PROCESS - EVALUATE MEASUREMENT

In this section we will present the main issues regarding the measurement evaluation of the products resulting from the measurement program (metrics, indicators, results, etc.) and the measurement process and the selection of the suitable measurement improvements to be applied.

Evaluate information product

It refers to evaluating information products, measures, indicators and analysis results, against specified evaluation criteria and to determine strengths and weaknesses of the information products. PSM [5] part 7 shows the evaluation criteria which is adapted from the evaluation criteria specified in ISO/IEC 15939[6]5.4.1 and are the following: measurement product use, confidence in measurement results, measurement results fitness for purpose, understandability of measurement results, satisfaction of the assumptions of an indicator model, measurement accuracy and measurement reliability.

Evaluate measurement process

Evaluating of the measurement process refers to evaluating the measurement process by means of quantitative evaluation of the measurement program performance (timeliness, efficiency, defect containment, customer satisfaction), conformance of the measurement process against the measurement plan (external audit to check the compliance of the process to a description of its intended use) and evaluation of the process to an external benchmark of process maturity. PSM [5] and ISO/IEC 15939[6] 5.4.1 cover these steps. In addition there are other analytical techniques to evaluate the measurement process.

As a measurement process evaluation framework we can underline the SEI SW-CMM model which presents a common area called “Measurement and Analysis” whose aim is to encourage organizations to use measures in order to improve its processes. It defines three types of measures for each maturity level: repeatable, defined and managed and optimizing. CMMI provides worthy enhancements regarding software measurement. It defines a new Process Area called “Measurement and Analysis” whose scope is wider than it was in CMM. The Measurement and Analysis process area supports all process areas by providing practices. These guides projects and organizations in aligning their measurement needs and objectives to results that can be used to make decisions and to take appropriate corrective actions. This idea matches with the *Goal-Question-Metric* model and the standard ISO 15939 [6].

SPICE and the ISO-15504 standard include the measurement process which covers all the processes that establish and support the achievement of the organizational business goals. The process treats the definition of measures, the data administration (including the historical data) and the use of metrics in the organization. The aim is to implement and use measures in order to support an effective administration and to objectively demonstrate the quality of the products.

ISO 90003 establishes the need of implementing the process with the aim of controlling the product, the process capability and the client satisfaction. The administration practices use the measures as the essential entry to plan, control and administrate the project and also to control the quality. All of it is focussed on the process improvement.

We now present an analytical technique aimed at assessing and deriving improvements from certain processes (e.g. Key Areas of CMMI) and the measurement processes related to them. This techniques was proposed by Berry and Vandenbroek [28] and it is a bottom-up approach. Therefore the measurements and the reference processes improvements 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 that they assign to measures; and between the performance of the key processes and the importance they assign to key processes. In addition this framework addresses the use of other models such as the: Performance model, Best Practices model, Meta-Measurement model, Social Impact model, Visualisation and Improvement model.

Berry et al. [29] also studied that this targeted measurement key process assessment identified better what has to be changed than the generic Information Quality model based on AIQM. However it required more effort to acquire and deploy.

Another improvement technique could be observing whether the measurement process matches the well known success factors for a measurement program. Gopal et al. [30] show and demonstrate most of the known measurement program success factors.

Measurement process improvement

The measurement process and measurement product deficiencies found in the previous tasks are candidate improvements and should be gathered in the measurement experience data base (Update Experience Base). Some of these candidate improvements will be implemented in the current project's process measurement process To select those candidate improvement to implement in short term it is recommended to perform a costs and benefits analysis [5] part 7 and [6] 5.4.2.

In addition the simulation technique such as the studies of Eickelmann, [9]could be used as a Visualization model in order to analyse beforehand, for example, the benefit of the improvements.

8. CONCLUSIONS AND FURTHER RESEARCH

In this paper we have presented the state of the art regarding software measurement programs. We have organized the approaches and studies in literature based on software measurement classification proposed in [1] but adding an extension where "Process and Business Assessment" sub-classification was added under measurement process, "Collecting Data", "Analysis", "Feedback" and "Packing" were added under "Perform Measument Process"; "Evaluate Measurement Products", "Evaluate Measurement Process" and "Measurement Process Improvement" were added under "Evaluate Measurement" and finally "Experience Factory" classification was added to "Tools and Techniques" classification. Hence, the most representative models and standards and some approaches which complement them regarding the measurement process and business assessment, measurement process plan definition, measurement process performance and the measurement evaluation have been tackled in this paper.

All the knowledge presented in this paper as the state of the art of the measurement process will serve us to base our future research whose target is to set up a framework in order to support SMEs in performing their measurement programs. Especially we will focus on the definition of software indicators. The methodology should take into account the limitations of the SMEs and their maturity. All this work will be developed under the COMPETISOFT project which is based, as we indicated above, in the MoProSoft Software Process Model. The aim of this project is to establish a software process model focused on SMEs for the Latin American countries.

Regarding our research, the relevant point found in literature is that none of the frameworks that support measurement programs, Goal Question Metric (GQM)[2], Goal-Driven Software Measurement [3], GQ(I)M[3, 4], PSM[5] and ISO/IEC 15939 [6], give guidelines about the definition of the indicators and measures according to the maturity of SMEs. Some relevant aspects to highlight are the following:

- What kind of indicator and analysis could we define depending on the maturity of a small and medium company? What are the constraints of each type of indicator? For example, some methods, such as the proposed in [14] which support the definition of prediction measurement processes, require historical data in order to statistically demonstrate the hypothesis defined. But we can not expect that amount of data in a not mature small company.

- What are the basic indicators required in SMEs depending on the maturity of the company?
- More support is required regarding the measurement training in SMEs in order to be focused on its needs, domain and to be effective.

From the issues identified previously the main objective of our ongoing research is to provide SMEs with a methodology framework based on Goal-Driven Software Measurement [3] and PSM[5]which support the definition of software indicators that would depend on SMEs' limitations and their maturity.

REFERENCES

1. Abran, A., Buglione, L., and Sellami, A. Software Measurement Body of Knowledge – Initial Validation using Vincenti's Classification of Engineering Knowledge types. *Software Measurement Conference*. (2004), pp. 1-16.
2. Solingen, R.v. and Berghout, E. *The Goal Quesiton Metric Method - A practical guide for Quality Improvement of Software Development*. Mc Graw Hill. 1999.
3. Park, R.E., Goethert, W.B., and Florac, W.A. *Driven Software Measurement-A Guidebook*. Carnegie Mellon University Pittsburgh: Software Engineering Institute. 1996.
4. Goethert, W. and Siviy, J. Applications of the Indicator Template for Measurement and Analysis. *Software Engineering Measurement and Analysis Initiative.*September 2004
5. *PSM: Practical Software and Systems Measurement - A Foundation for Objective Project Management Version 4.0c*. Deptartment of Defense and US Army. November, 2000.
6. ISO/IEC 15939, Software Engineering-Software Measurement Process. 2002
7. Cater-Steel, A.P. Process Improvement in Four Small Software Companies. *Proceedings of the 13th Australian Software Engineering Conference (ASWEC'01)*. (2001), pp. 262-272.
8. *MoProSoft v. 1.1.* http://www.software.net.mx 2004.
9. Eickelmann, N.S. Empirical Studies to Identify Defect Prevention Opportunities Using Process Simulation Technologies. *Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop (SEW.01)*. (2001), pp. 22-26.
10. McGarry, F., et al. *Software Process Improvement in the NASA Software Engineering Laboratory*. Institute, S.E. 1994.
11. Bandinelli, S., et al. Modeling and Improving an Industrial Software Process. *IEEE Transactions on Software Engineering*. 21 (1995), pp. 440-454.
12. Kellner, M.I., et al. Process Guides: Effective Guidance for Process Participants. *5th International Conference on the Software Process*. (1998), pp. 11-25.
13. Agresti, W. *The Role of Design and Analysis in Process Improvement*. in Elements of Software Process Assessment and Improvement. Emam, K.E. and N. Madhavji IEEE CS Press. 199.
14. Collofello, J. and Gosalia, B. An Application of Causal Analysis to the Software Production Process. *Software Practice and Experience*. 23 (10), (1993), pp. 1095-1105.
15. Nakajo, T. and Kume, H. A Case History Analysis of Software Error Cause-Effect Relationship. *IEEE Transactions on Software Engineering*.17, 8, (1991)
16. Florac, W. and Carleton, A. *Measuring the Software Process: Statistical Process Control for Software Process Improvement*. Addison Wesley. 1999.
17. Juan F. Ramil and Lehman, M.M. Defining and Applying Metrics in the Context of Continuing Software Evolution. *Proceedings of the Seventh International Software Metrics Symposium (METRICS'01)*. (2001), pp. 199-209.
18. Humphrey, W. *A Discipline for Software Engineering*. Addison Wesley. 1995.

19. Niessink, F. and Vliet, H.v. Measurements Should Generate Value, Rather Than Data. *Proceedings of the Sixth International Software Metrics Symposium (METRICS'99)*. (1999), pp. 31-39.

20. Briand, L.C., Morasca, S., and Basili, V.R. An Operational Process for Goal-Driven Definition of Meassures. *IEEE Transactions on Software Engineering*. 28, (December 2002), pp. 1106-1125.

21. Brijckers, A. and Differding, C. The Role of Software Process Modeling in Planning Industrial Measurement Programs. *Proceedings of theThird International Sysmposium on Software Metrics (METRICS'96)*. (1996), pp. 31-40.

22. Lavazza, L. and Barresi, G. Automated Support for Process-aware Definition and Execution of the Measurement Plans. *Proceedings of the 27th international conference on Software engineering*. (2005), pp. 234-243.

23. Baldassarre, M.T., Caivano, D., and Visaggio, G. Comprehensibility and Efficiency of Multiview Framework for Measurement Plan Design. *Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03)*. (2003)

24. Lindvall, M., et al. Towards Reusable Measurement Patterns. *Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS 2005)*. (2005), pp. 21-29.

25. Umarji, M. and Emurian, H. Acceptance Issues in Metrics Program Implementation. *11th IEEE International Software Metrics Symposium (METRICS'05)*. (2005), pp. 20-30.

26. Rini van Solingen and Berghout, E. Integration Goal-Orientated Measurement Software Engineering: Industrial Experiences with and Additions to the Goal/Question/Metric Method (GQM). *Proceedings of the Seventh International Software Metrics Symposium (METRICS'01)*. (2001)

27. Koennecker, A., Jeffery, R., and Low, G. Implementing an Experience Factory based on existing Organisational Knowledge. *IEEE* (2000)

28. Berry, M. and Vandenbroek, M.F. A Targeted Assessment of the Software Measurement Process. *Proceedings of the Seventh International Software Metrics Symposium (METRICS .01)*. (2001), pp. 222 -236.

29. Berry, M., Jeffery, R., and Aurum, A. Assessment of Software Measurement: an Information Quality Study. *Proceedings of the 10th International Symposium on Software Metrics (METRICS'04)*. (September, 2004), pp. 314-325.

30. Gopal, A., et al. Measurement Programs in Software Development: Determinants of Success. *IEEE Transactions on Software Engineering*. 28(9) (2002), pp. 863-875.