# Software Measurement Architectures: A Mapping Study

Ciro Xavier Maretto[1], Monalessa Perini Barcellos[1]

[1] Ontology and Conceptual Modeling Research Group (NEMO), Department of Computer
Science, Federal University of Espirito Santo, Vitória, Brazil
{ciro.maretto, monalessa}@ufes.br

**Abstract.** During the execution of software projects, it is necessary to collect, store and analyze data to support project and organizational decisions. Software measurement is a fundamental practice for project management and process improvement. It is present in the main models and standards that address software process improvement, such as ISO/IEC 12207, CMMI and MR MPS.BR. In order to effectively perform software measurement, it is necessary an infrastructure to support data collection, storage and analysis. This article presents a study that investigated measurement architectures described in the literature. As a result, eight architectures were found. Their main characteristics were analyzed and are presented in this paper.

**Keywords:** Systematic Mapping Study, Software Measurement, Measurement Architecture, Measurement Repository.

## 1 Introduction

Software Measurement is used by organizations in many ways. For instance, in the context of project management, measurement helps develop realistic plans, monitor progress, identify issues and justify decisions [1]. Throughout projects, data are collected for the measures and should be stored in a measurement repository in order to be used in project management and process improvement [2]. In maturity models that organize the software processes in maturity levels, such as CMMI (*Capability Maturity Model Integration)* [3] and MR MPS.BR (Reference Model for Process Improvement of Brazilian Software) [4], measurement is located at initial levels (CMMI level 2 and MR MPS.BR level F) and evolves as the maturity level increases. At high maturity levels (CMMI levels 4 and 5 and MR MPS.BR levels A and B) statistic process control (SPC) must be carried out and it requires extra attention in some measurement aspects, such as data storage.

It is not easy to implement and maintain a measurement repository capable of attending the needs according to the organization maturity level. Usually, organizations start recording measurement data in spreadsheets or in some systems with little or no integration among them [5]. At initial maturity levels, spreadsheets seem to be enough, but as the organization's maturity level increase, the problems of using spreadsheets become more expressive. Most times, to achieve high maturity, organizations need to discard data stored in spreadsheets, develop a measurement repository by using appropriate technologies (e. g., database management systems), and restart the collection and storage of project data. Thus, a good practice is to define an infrastructure which support software measurement and can be used from the beginning of a measurement program until the high maturity levels (or that can be extended to that) [2].

This infrastructure is made of components and can be defined by means of an architecture. According to Zachman [6], an architecture can be understood as a logical structure in which the components are organized and integrated. In the software measurement context, architecture should consider aspects related to the data collection, storage and analysis. In a measurement architecture, one of the main components is the measurement repository. According to Bernstein [7], a repository can be defined as a database sharing information about engineering artifacts. In a measurement architecture, the measurement repository stores measurement data (not limited to the collected data to the measures) and acts as a data provider to analysis.

Aiming to identify proposals to software measurement architecture, we carried out an investigation into the literature. According to Kitchenham [8], a systematic mapping (also known as exploratory study) makes an extensive study in a topic of a specific theme and aims to identify available evidence about that topic. In this sense, we carried out a systematic mapping. For each identified architecture we analyzed its characteristics and verified if the proposal provide support to the SPC.

In this paper, we present the main results of the study. After this introduction, in section 2, we briefly present software measurement and statistical process control; in section 3, the methodology used is described; in section 4 the research protocol is presented; in section 5 the main obtained results are shown; in section 6 some considerations about the results are performed; and finally, in section 7 some final considerations are made.

## 2 Software Measurement and Statistical Process Control

Software measurement is a primary support process for managing projects. It is also a key discipline in evaluating the quality of software products and the performance and capability of organizational software processes. The software measurement process includes the following activities: planning the measurement process, execution of the measurement process, and measurement evaluation [9].

For performing software measurement, initially, an organization must plan it. Based on its goals, the organization has to define which entities (processes, products and so on) to consider for software measurement and which of their properties (size, cost, time, etc.) are to be measured. The organization has also to define which measures are to be used to quantify those elements. For each measure, an operational definition should be specified, indicating, among others, how the measure must be collected and analyzed. Once planned, measurement can start. Measurement execution involves collecting data for the defined measures, according to their operational definitions. Once data are collected, they should be analyzed. The data analysis provides information to the decision making, supporting the identification of appropriate actions. Finally, the measurement process and its products should be evaluated in order to identify potential improvements [10].

Depending on the organization’s maturity level, software measurement is performed in different ways. At initial maturity levels, such as the levels 2 and 3 of

CMMI, the focus is on developing and sustaining a measurement capability that is used to support project management information needs. At maturity levels, such as CMMI levels 4 and 5, measurement is performed for the purpose of statistical process control (SPC), in order to understand the process behavior and to support software process improvement efforts [11]. SPC uses a set of statistical techniques to determine if a process is under control, considering the statistical point of view. A process is under control if its behavior is stable, i.e., if their variations are within the expected limits, calculated from historical data. The behavior of a process is described by data collected for performance measures defined to this process [12].

A process under control is a stable process and, as such, has repeatable behavior. So, it is possible to predict its performance in future executions and, thus, to prepare achievable plans and to improve the process continuously. On the other hand, a process that varies beyond the expected limits is an unstable process and the causes of these variations (said special causes) must be investigated and addressed by improvement actions in order to stabilize the process. Once the processes are stable, their levels of variation can be established and sustained, being possible to predict their results. Thus, it is also possible to identify the processes that are capable of achieving the established goals and the processes that are failing in meeting the goals. In this case, actions to change the process in order to make it capable should be carried out [12].

Statistical process control requires some changes in the traditional measurement, specially related to operational definition of measures, data collection frequency, measurement granularity, data homogeneity and data grouping to analysis [2, 13].

## 3 Metodology

In order to perform the systematic mapping, we used the process proposed in [14], which was defined based on [8]. It consists of the following three activities:

*i) Develop Research Protocol:* In this step the researcher prospects the topic of interest, defines the context to be considered in the study, and describes the object of analysis. Next, he/she defines the research protocol that will be used as a guideline to perform the research. The protocol must contain all the necessary information for a researcher to perform the research (research questions, source selection criteria, publication selection criteria, procedures for storing and analyzing the results, and so on). The protocol must be tested in order to verify its feasibility, i.e., if the results obtained are satisfactory and if the protocol execution is viable in terms of time and effort. The test results allow for improving the protocol when necessary. If the protocol is viable, an expert must evaluate it and once approved, the protocol can be used to conduct the research.

*ii) Perform Research:* In this step the researcher performs the research according to the research protocol. Publications are selected, and data are extracted, stored, and quantitatively and qualitatively analyzed.

*iii) Provide Results:* In this step the research results produced during the execution of the systematic review process should be packaged and published in a conference, journal, technical report or other publication vehicle.

## 4 Research Protocol

The research protocol used in the study contains the following information: objective, research questions, sources selection criteria, publications selection criteria, data storage and data analysis procedures, and protocol test procedure.

*A. Objective*

Analyzing the literature in the context of software measurement architectures, with the main purpose of identifying and analyzing:

*(i)* Proposals for software measurement architectures;
*(ii)* The proposals characteristics;
*(iii)* If the proposals are capable of supporting the statistical process control.

*B. Research Questions*

Q1. Which proposals for software measurement architecture are reported in the literature?
Q2. What are the proposals characteristics?
Q3. Which proposals include support to statistical process control?

In Q3, support to statistical process control consists in support: data collection, storage, representation (by means of control charts), and process behavior analysis.

*C. Sources*

The publications sources must be digital libraries and:

*(i)* Have a search mechanism that allows the use of logical expressions and search in different parts of the publications;
*(ii)* Be available in the CAPES (Coordination for the Improvement of Higher Education Personnel) Journals Portal[1];
*(iii)* Include publications in the Physical Science area, in particular Computer Science.

*D. Procedure for Publications Selection*

The object of analysis are papers published in conferences and journals. Publications selection must be done in three steps:

*1st step – Preliminary selection and cataloging*: the preliminary selection must be done by applying the following criteria using the digital library search mechanism:

*Scope:* title, abstract and keywords.
*Language:* English.
*Search String:* ("measurement framework" OR "measurement database" OR "measurement repository" OR "measurement architecture" OR "metrics repository" OR "metrics database") AND "software".
*Period:* from 1990.
*Area:* Computer Science.

For establishing the search string, we performed some tests using different terms, logical connectors, and combinations between them, aiming to obtain a search string able to return relevant publications to the study and a viable quantity to be analyzed.

[1] CAPES Journals Portal (www.periodicos.capes.gov.br/) is sponsored by Brazilian government and offers access to the publications of many international and national sources, covering all knowledge areas.

During the informal literature review that preceded the study, we found some relevant publications addressing measurement repositories. In fact, although these publications use the term measurement repository, in the context of the study they address measurement architecture. Thereby, we decided to include in the search string terms related to repositories.

Also during the informal review we identified two relevant publications ([15] and [16]) that we used as control publications to evaluate the search strings (the string must be able to return the control publications). The tests to obtain the search string were carried out using the digital libraries Scopus (*www.scopus.com*) e IEEE (*ieeexplore.ieee.org*). Scopus was selected because during preliminary tests it returned the largest number of publications. IEEE, in turn, was selected because the control publication [16] was only available in IEEE.

Considering the tests results we decided to select a comprehensive string and to restrict the publications selection in the later steps, since more restrictive strings excluded one or both the control publications. The selected string returned many publications that deal with measurement repositories not related to software measurement, but to scientific experiments from other computer areas. However, when we tried to restrict the publications by using the term "software measurement" instead of "software", the search results were very restricted and one of the control publications was not returned. So, even being a comprehensive string, the string selected was the one which provided better results in terms of number and relevance of selected publications.

We decided to apply the search string to the title, abstract and keywords, because some tests applying the string to the full text resulted in a large number of publications, being many of them useless. On the other hand, when restricting the string only to the title, useful publications were eliminated.

*2nd Step – Selection of Relevant Publications – 1st filter*: selecting publications by applying a search string does not ensure that all selected publications are relevant, because such selection is restricted to syntactic aspects. Thus, the abstract of the publications selected in the 1st step must be analyzed. Publications that do not satisfy one of (or both) the following criteria must be eliminated:

SC1: The publication addresses collection, storage, analysis or recovering of measurement data.

SC2: The publication addresses some kind of software measurement architecture or measurement repository.

We refer explicitly to measurement repositories in SC2 (and in SC3 presented forward), because, as it was said before, we noticed that some publications address measurement repository proposals that represent an architecture, according to the architecture concept used in the study (see Introduction).

In order to avoid premature exclusions of publications, in case of doubt, the publication should not be eliminated. Besides, publications without an abstract should not be eliminated.

*3rd Step - Selection of Relevant Publications – 2nd filter*: the selection of the publications in the 2nd step considers only the abstract. Consequently, it is possible that some selected publications do not contain relevant information. Therefore, the full text of the publications selected in the 2nd step must be read. Publications that do not satisfy one of (or both) the following criteria must be eliminated:

SC3: The publication describes software measurement architectures or measurement repositories.

SC4: The full text is accessible.

*E. Data Storage Procedure*

Each publication selected in the 1st step must be catalogued with the following data: title, author(s), year, reference data, source (digital library), and a summary. Each catalogued publication must be examined and submitted to the next two steps. The publications eliminated on the 2nd step must be identified as "E2: SC[number of the criteria not satisfied]". Similarly, publications eliminated on the 3rd step must be identified as "E3: SC[number of the criteria not satisfied]".

*F. Data Extraction and Analysis Procedure*

For each publication selected on the 3rd step, the following information must be extracted:

*(i)* Proposal identification. The identification is the proposal name as cited in the publication. If the proposal has no name, it must be identified as "Proposal XYZ", where XYZ are the initial letters of the proposal authors;

*(ii)* A brief description of the proposal;

*(iii)* Proposal characteristics, organized according to the following categories: Technology, Architecture, Collection, Storage, and Analysis;

*(iv)* Indication if the proposal supports statistical process control.

Regarding (iv), it must be recorded "Yes" to proposals whose publications make explicit the support to SPC. It must be recorded "Probably Applicable" to proposals that do not make explicit the support to SPC, but apparently they are able to support it. It must be recorded "No" to proposals that do not mention support to SPC and it is not possible to conclude that they support it.

After the data is extracted from publications, a quantitative and qualitative analysis must be done with the main purpose of discussing the findings related to the research questions.

*G. Test Protocol Procedure*

The research protocol must be tested using a reduced number of sources in order to verify if it is viable. The protocol is considered viable if the procedures are performed as described, if it is possible to answer the research questions and if the time and effort necessary are viable. During the protocol tests, some points need special attention:

*(i)* Number of publications selected on the 1st step: a large number of selected publications may mean that the string should be refined, because it is probably considering a larger domain than the target domain. It can be confirmed if many publications are eliminated in the subsequent steps. On the other hand, small number of selected publications may mean that many useful publications may be prematurely removed, that is, the search string is probably too restrictive.

*(ii)* Number of publications selected by the 2nd step: a large number of publications selected by the 2nd step related to the number of publications selected in the 1st

step might mean that either 2nd step criteria are too close to the search string and must be reviewed or 2nd step is unnecessary.

*(iii)* Number of publications selected for the 3rd step: a small number of publications selected in the 3rd step related to the number of publications selected in the 2nd step suggests that the criteria from the previous stage should be refined, because they probably are too wide in relation to the target domain. It is also important to consider that, in fact, only a small number of publications provide useful information for the research. Thus, once the criteria are aligned to the research objective and to the target domain, even if the number of selected publications is small, the criteria can be considered appropriated.

## 5 The Results

The protocol presented in the previous section was evaluated by an expert. Then, it was tested using the digital library IEEE. The protocol was considered viable and it was executed one more time using the digital library Scopus. In this section some results obtained from these two executions, carried out in November and December of 2011, are presented. Publications selected in both digital libraries were counted only once. In total, 148 publications were selected in the 1st step, 22 in 2nd the step and 12 in 3rd step.

It is possible to notice the significant decrease in the number of publications in the 2nd step. In fact, this result was expected, since we decided to use a comprehensive search string, as argued in section 4.

It is worth mentioning that the focus of the study is on measurement architectures and, for this reason, publications which described lessons learned and case studies that mention the use of measurement architecture (not describing the architecture) were excluded during the selection criteria application.

Analyzing the publications per year, from 148 publications selected by the search string (1st step), 25 (17%) are dating from 1990 to 2000 and 123 (83%) are dating from 2001 to 2011. From 12 publications selected in 3rd step, a quarter is dating from 2009 on. Besides, even we limited publications from 1990 on, the oldest publications are from 1999 and 2000.

From the publications selected in 3rd step, 8 proposals were identified. Table 1 shows a brief description of the proposals and their respective publications.

We analyzed the characteristics of each proposal. Due to space limitation it is not possible to present the characteristics in details. A summary is presented on Table 2. It is worth saying that the publications describe their proposals with different levels of detail and with different foci. Consequently, information regarding the characteristics has also heterogeneous detail levels. For instance, some proposals describe in details characteristics of the adopted architecture, while others just mention the general model in which the architecture is based on, and others nothing said about their architecture. In Table 2, when information regarding a category is not shown, it means that it was not possible to obtain information about it by reading the publications.

**Table 1.** Proposals found.

| Proposal | Description | Ref |
|---|---|---|
| P01 - Generic Measurement Framework Based on MDA | Software measurement framework to support the software measurement entities through metamodels and transformations. For example, given a model of an ER (Entities and Relationships) diagram, measures such as quantity of tables and relationships can be automatically calculated using the framework. For this, framework uses a domain model and a measurement model, which says which entities will be measured and what methods will be used. These models go through transformation processing QVT (Query View Transformation), which generates the measurements. | [15, 17, 18, 19] |
| P02 - WebEv (Web for the Evaluation) | System that uses a measurement framework based on GQM (Goal Question Metric) [20] to business process evaluation and gives support to data collection, storage and analysis. It was defined in terms of measures, mechanisms for data collection and guides to use the data collected. | [21, 22] |
| P03 - NSDIR (National Software Data and Information Repository) | It consists of an organizational benchmarking repository to software projects from the U.S Air Force. It was operational from 1994 to 1998. Although its use has ended up in 1998, the industry and academy efforts continued through CEBASE (Center for Empirically-Based Software Engineering). | [23] |
| P04 - MRS (Measurement Repository System) | It is a measurement repository used by a group of telecommunication companies. One of the main purposes was the supply and products evaluation through reporting generation which compiled data from all participating companies. As a big concern the repository has the safety and privacy of the information. | [24] |
| P05 - MMR Tool | Proposal of a generic and flexible measurement repository for data collection, storage, analysis, and publication. It was projected to give support to all CMMI levels and it was applied in Ericson Research Canada. | [16] |
| P06 - SPDW+ (Software Development Process Performance Data Warehousing) | It presents the data warehousing architecture SPDW+ as a repository solution centralized in measurements, automatic collection and analysis mechanisms. The SPDW+ is an improvement of SPDW that was operational for 3 years in HP Brazil. It was developed aiming the support of process improvements in mature organizations. | [25] |
| P07 - A Universal Metrics Repository | It proposes a structure to a flexible measurement repository, able to adapt itself to different lifetime models, methodologies, and software developments process. The proposal uses transformational view concepts of software development, which considers that the software development process is a series of artifacts transformation. | [26] |
| P08 – Proposal PAU | It presents a generic framework that incorporates database, a formal set of software tests and evaluation measures, as well as an advanced set of analytical techniques for information and knowledge extraction. The approach proposes using this framework and its techniques to extract detailed information and knowledge from the software measurement repositories. | [27] |

**Table 2.** Overview of the general characteristics of the identified proposals.

| Proposal | Features | | | | | |
|---|---|---|---|---|---|---|
| | **Technology** | **Architecture** | **Collection** | **Storage** | **Analysis** | **SPC Support** |
| P01 | Use of DSL(Domain-Specific Language) and tools based on Eclipse platform | Based on MDA (Model Driven Architecture) | Automatic (through models transformation) | XML file | | No |
| P02 | Use of Java (Java JDBC and Java Servlet API) | | Semi-automatic (via web form) | Database | Quantitative analysis resources | Probably Applicable |
| P03 | Use of Sun Solaris Unix, Oracle and client in Visual Basic with ODBC (Open Database Connectivity) | Client-Server (central repository which stores data collected by client software) | Manual and Semi-automatic (through physical or electronics forms) | Database | Analysis tools in a benchmark style | No |
| P04 | | Client-Server (central repository which stores data collected by client software) | Semi-automatic (through electronic form) | Database | Generation of quarterly reports | No |
| P05 | Use of Technologies and Microsoft tools (SQL 2000 Server, Analysis Services Enterprise Edition, Internet Information Server, Intranet Share Portal Server, ASP) | Based on data warehouse environment | Semi-automatic. Intend to use ETL (Extraction, Transformation and Loading) to collect voluminous and periodic data. | Data warehouse. The database model is generic for data flexibility | SQL (Structured Query Language) and OLAP (On-line Analytical Processing) cubes. Data is presented via web portal.<br>It is possible to export data to statistics tools. | Yes |

**Table 2.** Overview of the general characteristics of the identified proposals (cont.).

| Proposal | Features | | | | | |
|---|---|---|---|---|---|---|
| | Technology | Architecture | Collection | Storage | Analysis | SPC Support |
| P06 | Use of Microsoft technologies and tools. (SQL Server 2005, BI Studio, Visual Studio 2005, SQL Server Integration Services and IIS 6.0) | Oriented to services (SOA – Service Oriented Architecture) and based on data warehouse environment with four components | Semi-automatic and automatic, by using ETL. | Data warehouse | Use of BI tools (Business Intelligence) with web interface, including OLAP and dashboard. | Yes |
| P07 | Use of MySQL (only the repository is implemented) | | | Database. The database model is generic for data flexibility. | | No |
| P08 | | | Semi-automatic | Database | Use of statistical techniques and others, such as: multiresolution analysis, classification trees, neural networks and influence diagrams. | No |

## 6 Discussions

In this section we present additional information and some considerations about the results presented in the previous section. In general, the proposals identified are very different. Unfortunately, based on information from the publications, many times it is not possible to compare the proposals in a substantial way. Regarding the proposals characteristics, some considerations are presented below:

*Technology*

The technologies used in the proposals are diverse, varying from free software to proprietary technologies. This can be a reflex of the variety of technological solutions available in the market.

*Architecture*

All the proposals, except the Generic Measurement Framework based on MDA [15, 17, 18, 19], include in their architecture a central repository to store and retrieve data, using a client-server architecture. The proposals MRS [24] and NSDIR [23] have specific client programs for communication with the server. WebEv [21, 22], MMR Tool [16], and SPDW+ [25], in turn, use web resources. The proposals SPDW+ [25] e MMR Tool [16] have architectures based on data warehouse environment, including a component for data collection (ETL), a component to storage (data warehouse) and a component for analysis with analytical capabilities (OLAP). The SPDW+ [25] includes a fourth component responsible for the data integration. It acts as a temporary repository for standardization of the collected data.

The Generic Measurement Framework based on MDA [15, 17, 18, 19] is a conceptual architecture and it is an adaptation of MDA. It is divided in levels, ranging from MOF (Meta-Object Facility) to measurement data, also including a measurement meta-model based on a software measurement ontology.

*Collection*

In the Table 2 it is possible to notice three types of collection: manual, semi-automatic and automatic. Manual collection refers to the use of physical forms in order for people to record data collected for the measures. Semi-automatic collection refers to the use of computational support (for instance, electronic forms and information systems) to record data collected for the measures. In the semi-automatic collection, although there is computational support, data are supplied by people. Automatic collection refers to the use of computational tools and mechanisms which obtain data for the measures without human intervention.

Most of the proposals use semi-automatic collection. The publications which describe the proposals MMR Tool [16] e MRS [24] mention the intention of using automatic collection mechanisms, but these mechanisms are not presented on the publications. Only two proposals implemented the automatic collection: Generic Measurement Based on MDA, [15, 17, 18, 19], by means of models transformation; and SPDW+ [25], with a ETL component. It is important to emphasize that these proposals deal with very specific types of measures (for instance, quantity of tables and relationship in a certain data model and number of errors in a portion of source code), which are more favorable for the automatic collection. Therefore, proposals that deal with measures whose automatic collection is more difficult or not possible adopt semi-automatic collection. This can be seen as a sign of the difficulty and, in some cases impossibility, of adopting automatic collection. Only one proposal (NSDIR [23]) uses manual collection and the data collected in physical forms are after recorded in electronic forms.

*Storage*

The proposals use three different solutions to data storage: relational database (WebEv [21, 22]), XML (eXtensible Markup Language), files (Generic Measurement Framework Based on MDA [15, 17, 18, 19]), and  solutions based on database. Although most of the proposals adopt solutions based on databases, we noticed that each proposal support the storage of different measurement data. We believe that this occurs mainly because the repository structure (the database "model") is defined based on the specification of which entities and elements are to be measured and what information needs are expected to be satisfied by the measurement data.

We also noticed that some proposals provide flexibility regarding which measurement data can be stored. For instance, MMR Tool [16] uses a measurement domain meta-level structure as a data model, with the purpose of allowing adaptation to different measurement contexts. On the other hand, the Universal Metrics Repository [26] is itself a flexible database that aims to store any data from any measures related to different entities.

Finally, we observed that the proposals which include support to statistic process control (SPDW+ [25] e MMR Tool [16]) adopt solutions based on data warehouse.

*Analysis*

Most of the proposals include mechanisms for data analysis and presentation. Some proposals, such as SPDW+ [25] and PAU [27], have more complex mechanisms and tools. The analysis can be purely qualitative, as in WebEv [21, 22], or have a benchmark type, as in NSDIR [23] e MRS [24], in which general data of products and projects can be analyzed to support  identification of best practices. The proposals that support statistic processes control (SPDW+ [25] e MMR Tool [16]) adopt more sophisticated mechanisms to data analysis (both of them use OLAP tools).

*Support to SPC*

Most proposals do not provide support to statistic process control. For instance, the proposal NSDIR [23] includes a repository which stores general data regarding products and projects with the main purpose of using them as benchmarking. Data concerning the process definition or its executions are not stored, what does not allow for carrying out SPC.

Only two proposals (SPDW+ [25] and MMRTool [16]) include support to SPC. Both of them were developed in the context of large companies aiming at the high maturity levels. These two proposals use Microsoft technologies and solutions based on data warehouse environment.

## 7 Final Considerations

This paper presented the results of a systematic mapping about software measurement architectures. Altogether, 148 selected publications from the digital libraries IEEE and Scopus were analyzed and 8 software measurement architectures proposals were found. The proposals have some similarities (for instance, the use of solutions based on database for data storage in most of the proposals), but they also present many differences (for example, the technologies adopted).

Once the purpose of a systematic mapping is to present evidences from the literature about a specific topic, it was not purpose of the study to compare the proposals and determine which one is the best (or worst) of them. The main objective was to identify in the literature proposals for software measurement architecture and analyze them regarding their characteristics and support to SPC.

Nowadays, the results of this study are being used in the definition of a software measurement architecture for organizations aiming to achieve the high maturity.

As limitations of the study we highlight the use of only two digital libraries as sources of publications and the unavailability of the full text of some publications. Concerning the use of only two sources, although it is a limitation, initial tests showed that the selected publications from some other libraries were similar than the selected publications in the digital libraries used until this moment. Concerning publications whose full text was not available, we contacted the authors and some of them made their publications available. However, four publications were eliminated due to the unavailability of the full text.

## Acknowledges

This research is funded by the Brazilian Research Funding Agencies FAPES (Process Number 52272362/11) and CNPq (Process Number 483383/2010-4).

## References

1. Mcgarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., Hall, F.: Practical Software Measurement: Objective Information for Decision Makers. Addison Wesley, Boston, USA (2002).
2. Barcellos, M.: A Strategy for Software Measurement and Software Measurement Repository Evaluation for Statistical Process Control in High Maturity Organizations, Computation and Systems Engineering Program, Doctorate Thesis, Federal University of Rio de Janeiro, COPPE/UFRJ, Rio de Janeiro, Brazil, 2009. (in Portuguese only)
3. CMMI Product Team: CMMI for Development, Version 1.3., http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm, (2010).
4. SOFTEX: MPS.BR: Melhoria de Processo do Software Brasileiro - Guia Geral, http://www.softex.br/mpsbr, (2011).
5. Dumke, R., Ebert, C.: Software Measurement: Establish – Extract – Evaluate – Execute. Springer-Verlag (2010).
6. Zachman, J.: A framework for information systems architecture. IBM Systems Journal. 276–292 (1987).
7. Bernstein, P.A.: Repositories and Object-Oriented Databases, Proceedings of BTW Conference, 34–46. (1997).
8. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University. (2007).
9. ISO/IEC 15939, Systems and Software Engineering – Measurement Process, (2007).
10. Barcellos, M., Falbo, R. A., Rocha, A. R.: Establishing a Well-Founded Conceptualization about Software Measurement in High Maturity Levels. In: Proceedings of the 7th International Conference on the Quality of Information and Communications Technology. 467–472. (2010).

11. Barcellos, M.P., Falbo, R.A., Rocha, A.R.A.: A Well-Founded Software Process Behavior Ontology to Support Business Goals Monitoring in High Maturity Software Organizations. IEEE 5th Joint VORTE-MOST Workshop. pp. 253–262. Proceedings of the IEEE International EDOC Enterprise Computing Conference Workshops, Vitória - ES (2010).
12. Florac, W.A., Carleton, A.D.: Measuring the Software Process: Statistical Process Control for Software Process Improvement. Addison Wesley, Boston, USA (1999).
13. Tarhan, A., Demirors, O.: Apply Quantitative Management Now. IEEE Software. 29, 77–85 (2012).
14. Montoni, M., Investigation Regarding Critical Sucess Factors in Software Process Improvement Initiatives, Computation and Systems Engineering Program, Doctorate Thesis, Federal University of Rio de Janeiro, COPPE/UFRJ, Rio de Janeiro, Brazil, 2010. (in Portuguese only)
15. Mora, B., García, F., Ruiz, F., Piattini, M., Boronat, A., Gómez, A., Carsí, J.Á., Ramos, I.: Software generic measurement framework based on MDA. IEEE Latin America Transactions. 9, 130–137 (2011).
16. Palza, E., Fuhrman, C., Abran, A., Ouest, N., Québec, H.C.M.: Establishing a Generic and Multidimensional Measurement Repository in CMMI context. Proceedings of the 28th Annual NASA Goddard Software Engineering Workshop (2003).
17. Mora, B., García, F., Ruiz, F., Piattini, M., Boronat, A., Gómez, A., Carsí, J.Á., Ramos, I.: Software generic measurement framework based on MDA. IEEE Latin America Transactions. 8, 605–613 (2010).
18. Mora, B., García, F., Ruiz, F., Piattini, M., Boronat, A., Gómez, A., Carsí, J.Á., Ramos, I.: JISBD2007-08 : Software generic measurement framework based on MDA. IEEE Latin America Transactions. 6, 363–370 (2008).
19. Mora, B., Garcia, F., Ruiz, F., Piattini, M.: Model-Driven Software Measurement Framework: A Case Study. 2009 Ninth International Conference on Quality Software. 239–248 (2009).
20. Basili, V.R., Caldeira, G., Rombach, H.D.: The goal question metric approach, Encyclopedia of Software Engineering, Wiley. (1994).
21. Aversano, L., Bodhuin, T., Canfora, G., Tortorella, M.: WebEv - a Collaborative Environment for Supporting Measurement Frameworks. Proceedings of the 37th Annual Hawaii International Conference.1–10 (2004).
22. Aversano, L., Bodhuin, T., Canfora, G., Tortorella, M.: A Framework for Measuring Business Processes based on GQM Proceedings of the 37th Annual Hawaii International Conference. 1–10 (2004).
23. Goth, G.: focus NSDIR : A Legacy Early benchmarking effort was ahead of its time. IEEE Software. 18, 53–56 (2001).
24. Bastani, F.B., Ntafos, S., Harris, D.E., Morrow, R.R., Paul, R.: A high-assurance measurement repository system. Proceedings. Fifth IEEE International Symposium on High Assurance Systems Engineering (HASE 2000). 265–272 (2000).
25. Silveira, P.S., Becker, K., Ruiz, D.D.: SPDW+: a seamless approach for capturing quality metrics in software development environments. Software Quality Journal. 18, 227–268 (2010).
26. Harrison, W.: A flexible method for maintaining software metrics data: a universal metrics repository. Journal of Systems and Software. 72, 225–234 (2004).
27. Paul, R.A., Kunii, T.L., Shinagawa, Y., Khan, M.F.: Software Metrics Knowledge and Databases for Project Management. IEEE Transactions on Knowledge and Data Engineering, 11, 255–264 (1999).