

REVIEW

Open Access



Investigating measures for applying statistical process control in software organizations

Daisy Ferreira Brito^{1*} , Monalessa Perini Barcellos^{1*} and Gleison Santos^{2*}

* Correspondence: dfbrito@inf.ufes.br; monalessa@inf.ufes.br; gleison.santos@uniriotec.br

¹Ontology and Conceptual Modeling Research Group (NEMO), Department of Computer Science, Federal University of Espírito Santo, Vitória, ES, Brazil

²Graduate Program on Information Systems - Federal University of the State of Rio de Janeiro, UNIRIO, Rio de Janeiro, RJ, Brazil

Abstract

The growing interest in improving software processes has led organizations to aim for high maturity, where statistical process control (SPC) is required. SPC makes it possible to analyze process behavior, predict process performance in future projects and monitor process performance against established goals. The selection of measures for SPC can be a challenging task. Although the literature suggests measures for SPC, information is fragmented. With an aim towards providing a consolidated set of measures for SPC, as well as processes and goals related to these measures, we investigated the literature through a systematic mapping. Following that, we applied a questionnaire to three professionals from Brazilian organizations to check whether the measures they have used in SPC initiatives could also be found in literature. In this paper we discuss our main findings related to the 47 goals, 15 processes and 84 measures identified considering the systematic mapping and the questionnaire results.

Keywords: Software measurement, Statistical process control, Measure, Systematic mapping

1 Introduction

Software organizations have increased their interest in software process improvement (SPI). There are several standards and maturity models that support SPI implementation. Some of them, such as CMMI (Capability Maturity Model Integration) (CMMI Institute 2010) and MR-MPS-SW (Reference Model for Brazilian Software Process Improvement)¹ (Montoni et al. 2009), guide SPI implementation in levels. At the highest levels (such as CMMI levels 4 and 5 and MR-MPS-SW levels B and A), SPI involves statistical process control (SPC).

SPC was originally proposed in the manufacturing area to support improvement programs. SPC is used to determine if a process is under control from a statistical point of view. The use of SPC in software organizations is more recent and there are still some issues to be explored (Card et al. 2008). Different from manufacturing processes, software processes are human-intensive and creative. Thus, each execution of a software process has unique characteristics that may affect the process behavior (Komuro 2006).

The use of SPC in software organizations has revealed problems that affect the successful implementation of SPC (Takara et al. 2007; Barcellos et al. 2013). Unsuitable measures² and data are one of the main problems, since they postpone SPC practices until proper measures are identified and suitable data are collected (Kitchenham and

Charters 2007; Takara et al. 2007; Barcellos et al. 2013). In the literature, there are several works showing measures that can be used in SPC or that were used in SPC initiatives.³ However, information is often quite widespread and access to it can be difficult, burdensome and sometimes inefficient.

In view of the above, we believe that a comprehensive study providing information about measures for SPC is relevant for academics who want to investigate those measures and for professionals who want a basis to help them to define measures for SPC. Thus, we searched the literature looking for secondary studies providing a set of measures for SPC. Since we did not find any, we decided to investigate the literature to gather up a set of measures that can be useful in SPC initiatives.

To investigate the literature and ensure study comprehensibility and repeatability, as well as to reduce the researchers' influence on the results, we adopted a systematic approach through a systematic mapping. According to Kitchenham and Charters (2007), a systematic mapping provides an overview of a research area and helps identify gaps that can be addressed in future research. Additionally, three Brazilian professionals answered a questionnaire providing information about measures they have used in SPC initiatives.

This paper addresses the systematic mapping, the questionnaire and their main results. It extends further (Brito and Barcellos 2016) to present the main results of the systematic mapping. In the current paper, a more comprehensive background is provided, the results presented in (Brito and Barcellos 2016) are revisited, new information is presented (e.g., venues of the selected publications and new graphs) and publications from which the systematic mapping findings were obtained are informed. Moreover, we present the results of a questionnaire answered by three professionals to identify measures they have used SPC in Brazilian software organizations.

The paper is organized as follows: Section 2 presents the background for the paper, addressing software measurement and SPC; Section 3 concerns the systematic mapping; Section 4 addresses the questionnaire; Section 5 discusses the systematic mapping and questionnaire results; and Section 6 concludes the paper.

2 Software measurement and statistical process control

Software measurement is a process applied by organizations in several contexts. For instance, in project management, measurement helps to develop realistic plans, as well as monitor project progress, identify problems and justify decisions (McGarry et al. 2002). In process improvement initiatives, measurement supports the analysis of process behavior, as well as identifying needs for improvement and predicting if processes will be able to achieve the established goals (Florac and Carleton 1997).

Fenton and Pfleeger (1997) state that measuring software products, processes and projects is crucial for software organizations because measures quantify properties of these entities and allow you to obtain relevant information about the work done and to be done. The main purpose of measurement is to provide quantitative information to support decision making (Fenton and Neil 2000). In this sense, measurement should be applied to several software processes (e.g., project management, quality assurance, requirements engineering, coding, testing, etc.) to provide useful information to well-informed decision making at both project and organization level.

Software measurement is the continuous process of defining, collecting and analyzing data related to software processes and products to understand and control them, as well as supply meaningful information for their improvement (Solingen and Berghout 1999). It is a primary support process for managing projects, and is also a key discipline in evaluating software product quality and software process performance and capability (ISO/IEC 2007).

To perform software measurement, an organization must initially plan it. Based on its goals, the organization must define which entities (processes, products and so on) are to be considered for software measurement and which of their properties (e.g., size, cost, time etc.) are to be measured. The organization must also define which measures are to be used to quantify those properties. For each measure, an operational definition must be specified, indicating, among others, how data is to be collected and analyzed. Once planned, measurement can start. Measurement execution involves collecting data for the defined measures, storing and analyzing them. Data analysis provides information for decision making, supporting the identification of appropriate actions. Finally, the measurement process and its products should be evaluated to identify potential improvements (Barcellos et al. 2010).

Software measurement is an essential process for organizations to achieve maturity in software development. Depending on the organization's maturity level, software measurement is performed in different ways. At the initial levels (such as CMMI levels 2 and 3), measurement basically consists of collecting data from projects and comparing them with their corresponding planned values. At high maturity levels (such as CMMI levels 4 and 5), it is also necessary to carry out SPC to understand process behavior, determine their performance in previous executions and predict their performance in current and future projects, verifying if they are capable of achieving the established goals (Barcellos et al. 2013).

SPC uses a set of statistical techniques to determine if a process is under control, from a statistical point of view. A process is under control if its behavior is stable, i.e., if its variations are within the expected limits, calculated from historical data (Florac and Carleton 1999). The behavior of a process is described by data collected for measures that characterize the process (Barcellos et al. 2013).

A process under control is a stable process and as such, has repeatable behavior. Consequently, it is possible to predict its performance in future executions and thus prepare achievable plans and continuously improve the process. On the other hand, a process that varies beyond the expected limits is an unstable process. The causes of these variations (the so-called special causes) must be investigated and addressed by improvements aiming at stabilizing the process. Once the processes are stable, their levels of variation can be established and sustained, making it possible to predict process results, enabling also the possibility to identify which processes are capable of achieving the established goals and which ones are failing to achieve them. In this case, actions that change the process in order to make it capable must be carried out. Stabilizing critical processes is a practice of high maturity organizations or organizations that aim achieving the highest maturity levels (Florac and Carleton 1999).

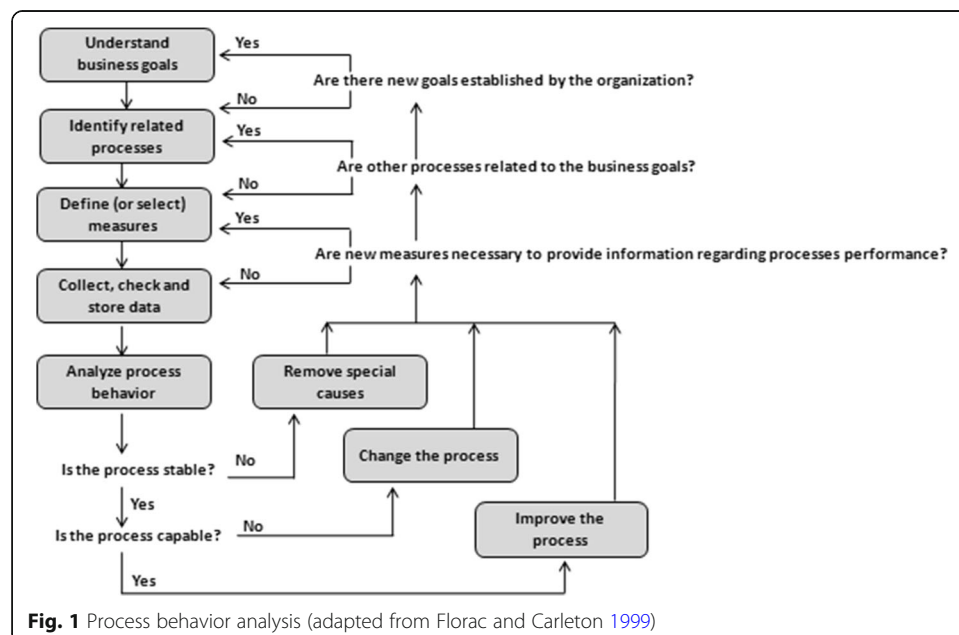
Figure 1 summarizes the process behavior analysis using SPC principles. First, it is necessary to understand the organizational business goals. Next, the processes related to business goals are identified and the measures used to provide quantitative

information about their performance are identified. Data are collected, checked, stored and used to analyze process behavior by means of statistical techniques. If a process is unstable, the special causes should be removed. If it is not capable, it should be changed. Finally, if it is capable, it can be continuously improved.

As shown in Fig. 1, organizations must understand their business goals and thus, identify the processes to be submitted to SPC and the measures to be used. These measures should be able to quantify aspects of process behavior and provide useful information regarding goals achievement. For example, an organization that has the goal *Reduce defects in delivered products* could select the *Inspection* process to be submitted to SPC and use, among others, the measure *inspection effectiveness* (ratio between the number of delivered defects and the number of detected defects) to analyze process behavior and goal achievement.

When applying SPC, data collected for measures are analyzed by using control charts, which enable the representation of process behavior variations and the analysis of process stability and capacity. There are several types of control charts (e.g., X-bar R, X-bar S, XmR) (Florac and Carleton 1999). Based on the data collected, control limits (upper, central and lower) are calculated and the process behavior is analyzed against these limits, considering stability tests, such as the ones defined by Wheeler and Chambers (1992), and capacity analysis methods, such as capacity index (Wheeler and Chambers 1992).

To define the measures, organizations can use approaches such as GQM (Goal Question Metric) (Basili et al. 1994). GQM is a systematic approach for tailoring and integrating goals for software processes, products and quality perspectives of interest, based upon project and organizational specific needs. To put it simply, GQM states that goals provide the basis from which it is possible to identify information needs that can be met by measures. By following this idea, organizations can derive information needs from their goals and define measures to meet the information needs. Although approaches such as GQM are useful, they do not provide measures that can be reused



by organizations. A set of measures already used in SPC initiatives could help organizations define their own measures.

In literature, there are several records of experiences involving the use of SPC in software organizations (e.g., Komuro 2006; Wang et al. 2008; Vijaya and Arumugam 2010 and Tarhan and Demirors 2012). From these experiences, it is possible to obtain knowledge about measures used in SPC and reuse it in other organizations. However, although the literature suggests several measures that can be used in SPC, information is dispersed among different publications and access to it is not trivial. Thus, a consolidated set of measures can be useful for organizations. With this in mind, we carried out the systematic mapping described in the next section.

3 Review

The systematic mapping was performed following the approach defined in (Kitchenham and Charters 2007), which includes three phases:

- (i) *Planning*: In this phase, the topic of interest, study context and object of the analysis are established. The research protocol to be used to perform the research is defined, containing all the necessary information for a researcher to perform the research: research questions, sources to be searched, publication selection criteria, procedures for data storage and analysis and so on. The protocol must be evaluated by experts and tested 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. Once the protocol is approved, it can be used to conduct the research.
- (ii) *Conducting*: In this phase, the research is performed according to the protocol. Publications are selected, and data are extracted, stored and quantitatively and qualitatively analyzed.
- (iii) *Reporting*: In this phase, the research results produced are recorded and made available to potential interested parties.

3.1 Research protocol

The systematic mapping goal was to identify measures that have been used in SPC initiatives for software processes or suggested for it. In order to achieve this goal, we defined seven research questions (RQ). Table 1 presents the research questions and their rationale.

The search string was developed considering three groups of terms that were joined with the operator AND. The first group includes terms related to SPC. The second includes terms related to measures and the third includes terms related to software. Within the groups, we used the OR operator to allow for synonyms. The following search string was used: (“*statistical process control*” OR “*SPC*” OR “*quantitative management*”) AND (“*measurement*” OR “*measure*” OR “*metric*” OR “*indicator*”) AND (“*software*”). To establish this search string, we performed some tests using different terms, logical connectors, and combinations among them. More restrictive strings excluded some important publications identified during the informal literature review that preceded the systematic mapping. These publications were used as control publications, meaning that the search string should be able to retrieve them. We decided to use a comprehensive string that provided better results in terms of number and

Table 1 Systematic Mapping Research Questions

ID	Research Question	Rationale
RQ1	When and in which type of vehicle have the publications been published?	Providing understanding on when and in which type of vehicles (e.g., conference, workshop, journal, etc.) the selected publications have been published.
RQ2	What measures have been applied in SPC initiatives (or suggested for it)?	Identifying measures that have been applied in SPC initiatives or that have been suggested for them and verifying if a same measure has been applied in many different initiatives or suggested by different publications.
RQ3	What goals have led to the use/suggestion of the measures?	Identifying the goals related to the measures and verifying if a same measure is related to different goals in different publications.
RQ4	What processes are the measures related to?	Identifying the processes to which the identified measures are related and verifying if certain processes have been used more often.
RQ5	Which are the measure categories?	Identifying categories of the measures and verifying if any category has been more frequent. For categorizing a measure, it must be considered the categories suggested in (MCGARRY et al.2002), namely: Time, Cost, Effort, Performance and Size.
RQ6	Have the measures been used in SPC initiatives?	Investigating if the identified measures have been used in practical experiences involving SPC.
RQ7	Have the measures been used in the context of SPI standards/models? Which ones?	Investigating if the identified measures have been used in the context of software process improvement programs based on standards or maturity models. Moreover, identifying the standards and models used in the initiatives.

relevance of the selected publications, even though it had selected many publications eliminated in subsequent steps.

Seven digital libraries were used as sources of publications: IEEE Xplore (ieeexplore.ieee.org), ACM Digital Library (dl.acm.org), Springer Link (<http://www.springerlink.com/>), Engineering Village (<http://www.engineeringvillage.com/>), Web of Science (webofscience.com), Science Direct (www.sciencedirect.com), and Scopus (www.scopus.com). These digital libraries were selected based on (Kitchenham and Brereton 2013), which suggests searching IEEE and ACM, which ensure good coverage of important journals and conferences, and at least two general indexing systems such as Scopus, Compendex (Engineering Village) and Web of Science. Besides the sources suggested in (Kitchenham and Brereton 2013), we also searched Springer Link and Science Direct because they have been used in other systematic reviews performed by members of the research group in which this work was carried out.

Selection of the publications was performed in five steps:

- (S1) *Preliminary selection and cataloging*, when the search string was applied in the search mechanisms of the digital libraries. In this step, we limited the search scope to the Computer Science area.
- (S2) *Duplicate Removal*, when publications indexed by more than one digital library were identified and the duplications were removed.
- (S3) *Selection of Relevant Publications – First Filter*, when the title, abstract and keywords of the selected publications were analyzed considering the following inclusion (IC) and exclusion (EC) criteria:
 - (IC1) the publication addresses SPC in software processes and measures used in this context.
 - (EC1) the publication does not have an abstract.

- (EC2) the publication is published as an abstract.
- (EC3) the publication is a secondary study, a tertiary study, a summary or an editorial.
- (S4) *Selection of Relevant Publications – Second Filter*, when the full text of the publications selected in S3 is read with the purpose of identifying the ones that provide useful information considering the following inclusion (IC) and exclusion criteria (EC):
 - (IC2) the publication presents measures for SPC in software processes or presents cases involving SPC in which the measures used are cited.
 - (EC4) the publication is a copy or an older version of an already selected publication.
 - (EC5) the publication is not written in English.
 - (EC6) the publication full text is not available.
- (S5) *Snowballing*, when, as suggested in (Kitchenham and Charters 2007), the references of publications selected in the study have been analyzed looking for the ones able to provide evidences for the study. Therefore, in this step, references of the publications selected in S4 were investigated by applying the first and second filters.

Publication selection was performed by the first author. For each publication, an identifier was defined and the following information was recorded: title, authors, year, reference and source. Publication selection was reviewed by the second author, who performed the publication selection procedure and reviewed the results obtained by the first author in each step. Discordances were discussed and resolved in meetings.

After selecting the publications, data were extracted and recorded. Data extraction and recording consisted of extracting data from the publications for each research question and recording them in a form designed as a spreadsheet. To extract measures, processes and goals, first we extracted those elements exactly as they were named in the publications (e.g., we extracted the measure *schedule variable*, which refers to the ratio between actual duration and estimated duration, from (Wang and Li 2005)). Next, we adjusted the elements' name aiming to make it clearer (e.g., we changed the name of the measure *schedule variable* to *duration estimation accuracy*). Finally, we identified elements with the same meaning and assigned the same name to all of them (e.g., all measures referring to the ratio between actual duration and estimated duration were named *duration estimation accuracy*). In summary, the data extraction procedure consisted of: (i) extracting the elements (goals, processes and measures) as they are named in the publications and recording the relations between them; (ii) adjusting names for clarity; (iii) unifying equivalent elements.

With regard to the relation between goals, processes and measures, we extracted and recorded only the relations that we found in the publications, i.e., we did not create new relations between goals, processes and measures. For example, even if there was a measure found in a publication that could be related to a process found in another, we did not record the relation because it was not defined in the publications analyzed.

Data extraction and recording were performed by the first author. The names used to represent the measures, processes and goals were based on information provided by the publications and on the researchers' interpretation. Aiming towards quality assurance, after data extraction and recording, data validation was performed by the second

and the third authors, who reviewed the extracted data. The review process consisted of: (i) reading the publications and verifying if data were correctly extracted; (ii) verifying the names given by the first author to goals, processes and measures; and (iii) verifying the goals, processes and measures the first author considered equivalent. Divergences were discussed and resolved.

Once data were validated, data interpretation and analysis were carried out. Quantitative data were tabulated and used in graphs and statistical analysis. Qualitative analysis was performed considering the findings, their relation to the research questions and the systematic mapping purpose.

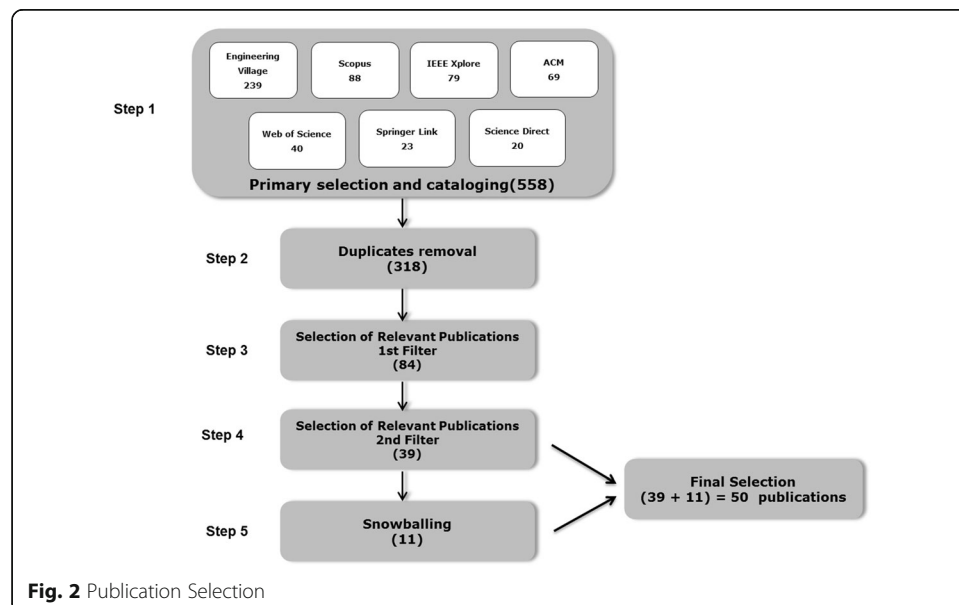
3.2 Results

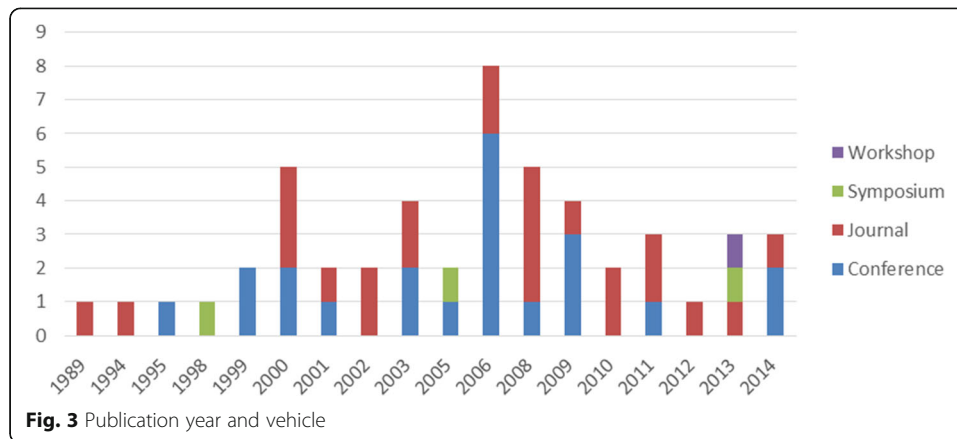
The systematic mapping considered studies published up to April 2016. As a result of S1, 558 publications were obtained (79 from IEEE Xplore, 88 from Scopus, 69 from ACM, 20 from Science Direct, 239 from Engineering Village, 40 from Web of Science and 23 from Springer Link). After S2, 240 duplications were eliminated, resulting in a total of 318 publications. After S3, only 84 studies were selected (a reduction of approximately 73.58%). After S4, we reached 39 studies. After applying the snowballing procedure(S5), 11 publications were added, reaching a total of 50 publications.

Figure 2 illustrates the process followed to select the publications, which resulted in 50 selected publications. The list of selected publications is shown in [Appendix 1](#).

There follows below, for each research question, a data synthesis of the main results obtained.

Publication vehicle and year (RQ1): Publication years range from 1989 to 2014, with occasional gaps, as shown in Fig. 3. With regard to publication vehicles, 26 publications (52%) were published at scientific events and 24 (48%) in journals. Among the publications published at scientific events, 22 were published at conferences, three at symposiums and one at a workshop. Journals usually require more





mature works. The homogeneous distribution of the studies in scientific events and journals can be seen as a sign that the topic has been explored, discussed and matured.

Table 2 presents the journals and scientific events where most of the publications were published. 12 (24%) of the publications were issued by the IEEE Software journal, revealing its predominance. It is followed by the Software Quality Journal, which published three (6%) of the selected publications, and by Software Process Improvement and Practice Journal, which published two (4%) of them. With regard to scientific events, the International Conference on Software Maintenance, the International Conference on Software Engineering and the International Conference on Software Quality published two (4%) of the selected publications. Venues that published only one of the selected publications are not shown in Table 2.

Measures for SPC (RQ2), Supported Goals (RQ3) and Related Processes (RQ4): In 2016, data was extracted and recorded, as described in the research protocol. As a result, a total of 108 measures, 15 processes and 49 goals were identified. These results were published in (Brito and Barcellos 2016). In this paper, we revisited these results and refined them, aiming to obtain a more consolidated set of measures, processes and goals.

We started off the refinement by providing a definition for the processes. This helped us to identify different processes that, in fact, refer to the same process; too large processes that could be decomposed into smaller ones more suitable for SPC; and processes that are subprocesses of others. In (Brito and Barcellos) the following processes were identified: Coding, Customer Release, Design, Fixing, Inspection, Maintenance, Project

Table 2 Publications Venue

Publication Venue	Type	Number of Publications	Percent
IEEE Software	Journal	12	24
Software Quality Journal	Journal	3	6
Software Process Improvement and Practice	Journal	2	4
International Conference on Software Maintenance - ICSM	Conference	2	4
International Conference on Software Engineering - ICSE	Conference	2	4
International Conference on Software Quality - ICSQ	Conference	2	4

Management, Quality Assurance, Recruitment, Requirements Development, Requirements Management, Review, Risk Management, Software Development and Testing.

According to (Fagan 1976), an inspection is a particular type of review that follows a well-defined and rigorous process to evaluate artifacts produced in software projects (Fagan 1976). Thus, Inspection and Review can both refer to the Review process. On analyzing the measures related to these processes in (Brito and Barcellos 2016), we noticed that all the measures could be related to the Review process. Therefore, we decided to eliminate the Inspection process and link the measures related to Inspection in (Brito and Barcellos 2016) to Review.

As for the Risk Management process, which can be considered a subprocess of Project Management (PMI 2012), we noticed that the only measure related to it in (Brito and Barcellos 2016) is a measure related to the Project Management process. Thus, we only kept the latter.

With regard to the Software Development process, it is too large for SPC (Tarhan and Demirors 2008; Barcellos et al. 2013). According to (ISO/IEC 2008), this process has several software-specific lower-level processes. Most of the measures related to the Software Development process in (Brito and Barcellos 2016) are, in fact, related to processes that comprise it. Thus, we broke down the Software Development process into Requirements Development, Requirements Analysis, Design, Coding and Testing.

With regard to the Quality Assurance process, on revisiting the publications analyzed in the study, we realized that they do not refer to the Quality Assurance process as a whole, but only to the Audit process, which can be performed aiming towards quality assurance. Therefore, we exchanged the Quality Assurance process for Audit. Although Audit can be deemed a type of review, we kept the Audit and the Review processes, the former referring exclusively to independent reviews and the latter referring to internal reviews.

Finally, the Customer Release process was eliminated because during measure refinement (explained later), all measures related to this process were excluded.

After these refinements, the resulting set of processes is: Audit, Coding, Design, Fixing, Maintenance, Project Management, Recruitment, Requirements Development, Requirements Management, Review, Requirements Analysis and Testing. Table 3 presents a definition for each of these processes.

With regard to goals, in (Brito and Barcellos 2016), 49 goals were cited. Revisiting these goals, we noticed that some of them had a very similar meaning and could be unified. Thus, we unified the goals *Reduce the number of delivered defects*, *Deliver a near defect-free system* and *Improve defect detection* in *Improve defect detection to reduce the number of delivered defects*. Moreover, some general goals encompass more specific goals, i.e., the last can be seen as sub-goal of the first. Considering that, we refined the set of goals indicating goals that can be sub-goals of others. Table 4 presents the goals and their relations. The table also shows the identifiers of the publications (see Appendix 1) from which the goals were extracted.

In addition to the links presented in Table 4, other relations between goals are possible. Table 4 shows the relations we considered more direct. For example, we represent *Improve software process effectiveness* as a sub-goal of *Improve product quality*, because process quality directly influences product quality (Fuggetta 2000). However, *Improve product quality* could also be a sub-goal of *Minimize rework*.

Table 3 Processes definition

Process	Definition
Audit	Independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria (ISO/IEC 2017).
Coding	Transforms specified designs, behavior, interfaces and implementation constraints into actions that create a software system element implemented as a software product or service (ISO/IEC 2017).
Design	Aims at providing sufficient detailed data and information about the system and its elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture (ISO/IEC 2017).
Fixing	Process in which bugs and defects observed in the software are handled and solved to improve the software quality during its development. Differently from the maintenance process, fixing refers to fixing bugs and defects detected before delivering the software to the client (e.g., defects detected during unit tests) (Ghapanchi and Aurum 2011).
Maintenance	Process responsible for making corrections, changes and improvements to deployed software and elements (ISO/IEC 2017).
Project Management	Involves activities related to both project planning and project monitoring and control. It establishes and maintains plans that define project activities and provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan (CMMI Institute 2010).
Recruitment	Aims to provide the organization with necessary human resources, consistent with business needs (ISO/IEC 2017).
Requirements Development	Responsible for eliciting, analyzing and establishing customer, product, and product component requirements (CMMI Institute 2010).
Requirements Management	Its purpose is to manage requirements and to ensure alignment between them and the project's plans and work products. It includes documenting requirement changes and their rationale and maintaining bidirectional traceability between source requirements, all product and product component requirements, and other specified work products (CMMI Institute 2010).
Review	Aims to evaluate the quality of a process or product. Involves a person or a group of people examining software processes, software components, a system or its associated documentation to discover potential problems (Sommerville 2006).
Requirements Analysis	Aims to provide a rigorous basis of data and information for technical understanding in order to aid decision-making across the life cycle (ISO/IEC 2017).
Testing	Consists in verifying and validating the software by executing it with the purpose of finding defects (Myers 2004).

Some goals are not related to others (G01, G02, G03, G07, G11, G12, G13 and G14). Most of these goals (G03, G07, G11, G12, G13 and G14) address test aspects and could be sub-goals of a general test-related goal. However, none of the goals identified in the study represents such a generalized goal. Thus, we did not relate them as sub-goals of others.

The goals *Reduce effort due to poor quality performance* and *Monitor response time in order not to delay software updates and changes* cited in (Brito and Barcellos 2016) were eliminated because during the measures refinement process (explained next), all measures related to these goals were excluded.

In (Brito and Barcellos 2016), 108 measures were cited. Analyzing the set of measures, we noticed that some of them were not normalized. If measures are not normalized, it is not possible to compare them nor use them to describe process behavior (Barcellos et al. 2013). For instance, the measure *number of defects* is not suitable for SPC, because it is not possible to analyze the behavior of the related process (e.g., Coding) considering the number of defects detected in source codes with different sizes. Thus, we eliminated the following measures: *maintenance time*, *number of action items detected in peer reviews*, *defects delivered*, *development effort*, *number of defects injected in coding*, *number of defects injected in design*, *number of defects injected in*

Table 4 Goals and sub-goals

ID	GOAL	ID	SUB-GOAL
G01	Assess and monitor the maintenance process (P22, P27)	–	–
G02	Control variation in coding and code review processes (P34)	–	–
G03	Estimate and control defects, effort and schedule of testing process (P19)	–	–
G04	Evaluate process quality effectiveness (P35, P50)	G04.1	Evaluate defect-detection effectiveness (P41)
		G04.2	Evaluate inspection effectiveness (P03, P28)
		G04.3	Evaluate peer review effectiveness (P21, P24, P44)
		G04.4	Manage effectiveness of defect removal activities (P16)
G05	Improve product quality (P05, P10, P12, P16, P20, P23, P24 P35, P39, P44, P50)	G05.1	Improve defect detection to reduce the number of delivered defects (P18, P38, P43)
		G05.2	Improve software process effectiveness (P03, P10, P32)
		G05.3	Improve software reliability (P01)
		G05.4	Increase customer satisfaction (by managing defects) (P04)
		G05.5	Manage defect injection distribution in different kinds of activities (P16)
		G05.6	Reduce defects in the products (P13, P32, P36)
		G05.7	Reduce injected defect (P06, P43)
		G05.8	Reduce requirements volatility (P23)
		G05.9	Understand and predict product quality (P40)
		G05.10	Verify quality goals achievement (P26)
G06	Improve review process (P25, P43)	G06.1	Understand and predict inspection process (P40)
G07	Manage system-testing activity (P16)	–	–
G08	Monitor process efficiency (P31)	G08.1	Evaluate coding efficiency (P49)
		G08.2	Evaluate defect-fixing efficiency (P16)
		G08.3	Evaluate design efficiency (P49)
		G08.4	Evaluate testing efficiency (P16, P24, P49, P50)
G09	Reduce operational costs (P04)	G09.1	Improve productivity (P04, P16, P39, P44)
		G09.2	Minimize rework (P11)
		G09.3	Monitor project cost and schedule (P20, P29)
		G09.4	Reduce cost due to poor quality performance (P26)
		G09.5	Improve estimation and planning (P44)
G10	Understand software processes performance (P07, P20)	G10.1	Understand fixing process performance (P07, P17)
		G10.2	Understand review process performance (P07, P17)
		G10.3	Understand project management process performance (P07, P08, P17)
		G10.4	Understand recruitment process performance (P07, P20)
		G10.5	Understand test process performance (P07)

Table 4 Goals and sub-goals (Continued)

ID	GOAL	ID	SUB-GOAL
G11	Understand the effect of reviews as verification activities in test (P09)	–	–
G12	Understand the effect of test design in test development (P07, P09,P17)	–	–
G13	Understand the relationship between productivity and quality assurance activities during test development (P07, P09,P17)	–	–
G14	Verify changes in test process (P26)	–	–
G15	Win the market competition (P15)	G05	Improve product quality (P05, P10, P12, P16, P20, P23, P24 P35, P39, P44, P50)
		G09	Reduce operational costs (P04)

requirements, test development effort, test development internal review effort, test design effort, test design internal review effort, test procedure preparation effort, test procedure preparation internal review effort, number of defects, effort, action items resolution effort, test development peer review effort, defect-fixing effort, amount of time spent responding to problems. However, it is important to notice that if these measures can be normalized they can be useful within the SPC context. For instance, if the measure *maintenance time* is normalized by product size (e.g., number of KSLOC) or by number of solved defects, it can adequately describe the maintenance process behavior and be used in SPC.

After eliminating unnormalized measures, we revisited the publications selected in the study and verified if measures referred to by different names in (Brito and Barcellos 2016) are equivalent. Most of the publications do not provide information about the operational definition of the measures. This makes it hard to understand the measures' meaning, and identify equivalent measures. For instance, some measures refer to *problems*, while others refer to *non-conformances*. Since the publications do not provide a clear operational definition to the measures, it can be difficult to understand if what is referred to as *problem* in a publication is equivalent to what is referred to as *non-conformance* in another. We revisited the publications and analyzed information about the measures in examples, graphs, descriptions, etc. This allowed us to identify equivalent measures. For example, the measure *problem arrival rate* (*problems detected/product size*) is equivalent to *defect density* (*number of detected defects/product size*) and the measure *defect removal rate* (*number of removed defects/effort spent removing defects*) is equivalent to *rework efficiency* (*number of fixed defects/defect fixing effort*).

After refining the measures, we analyzed the relation existing between the resulting set of measures and the processes. We noticed that some measures were related to processes which the measure is not able to characterize. Thus, we removed these relationships and related the measures to the processes they characterize. In this sense, the relationship between *Defect detection efficiency* (*number of defects in tests/effort spent reviewing tests*) and the Testing process was eliminated and the measure was related to the Review process, because the measure refers to the efficiency of reviews that evaluate tests. Additionally, the relationship between *review speed* (*product size/time spent on review*) and Coding was removed, while its relationship to Review was maintained.

We also analyzed the relationships between measures and goals with a view towards identifying any of these relationships where the measure is not able to support the goal. Thus, we removed the relationship between the measures *effort estimation accuracy* (*actual effort/estimated effort*) and *duration estimation accuracy* (*actual duration/estimated duration*) and the goal *Improve product quality* and related these measures to the goal *Improve estimation and planning*.

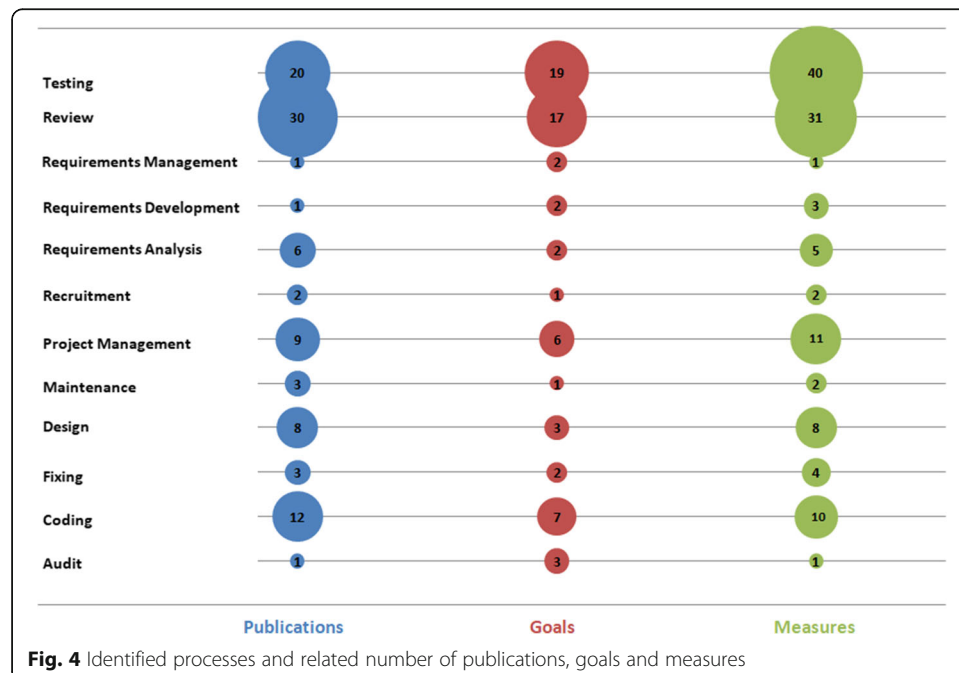
The resulting set of measures, goals and processes is shown in [Appendix 2](#).

Figure 4 shows the identified processes (y-axis), the number of publications citing them and the number of goals and measures related to each process. The circle size refers to the number of elements they represent. For example, the *Testing* process was cited in 20 publications. In these publications, 19 goals and 40 measures related to Testing were reported.

As the figure shows, Review and Testing were the most cited processes (respectively in 30 and 20 publications), followed by Coding (12 publications), Project Management (9 publications), Design (8 publications) and Requirements Analysis (6 publications). Therefore, most of the goals and measures are related to Review or Testing, indicating a predominance of defect-related measures, followed by project management and coding-related measures. Requirements Management, Requirements Development and Audit were the less cited processes (only one publication). Only one measure was reported to Requirements Management and Audit processes.

Measures Category (RQ5): From the 82 measures identified, 32 (39,02%) are related to Quality, 15 (18,29%) to Effort, 20 (24,39%) to Performance, 10 (12,19%) to Time, and 5 (6,09%) to Cost.

Use of Measures in the context of Standards/Maturity Models (RQ6 e RQ7): The majority of the measures identified were applied in practice (79 measures, 96,34%) and most of these (66 measures, 83,54%) were used in SPC initiatives involving standards/



maturity models. All these measures were used in SPI initiatives involving CMMI. Among them, the following measures were also used in initiatives involving ISO 9001 (ISO 2015) (corresponding to 15,15% of the identified measures): defect density, effort estimation accuracy, duration estimation accuracy, percentage of effort saved due to process automation, Review effectiveness, time spent on review preparation per reviewer, effective preparation speed, effective review speed, preparation speed and review speed.

3.3 Discussion

Most of the measures identified are related to defects (39 measures, 47,56%) and consequently, to processes that deal with defects, such as *Testing* and *Review*. Measures related to defects are often used in SPC for two main reasons: (i) processes addressing defect-related measures are directly related to software quality, and are therefore critical to organizations and natural candidates for SPC, since critical processes are the ones indicated to be statistically controlled (Tarhan and Demirors 2008; CMMI Institute 2010; Barcellos et al. 2013); (ii) these processes are performed many times in projects, favoring data collection and obtaining the amount of data required for SPC.

Defect density was the most cited measure, and it was used in 33 publications (66%). In some studies, this measure is applied to quantify different types of defects (e.g., in P15, *code defect density* and *file defect density*).

Review was the most frequently cited process, being used in SPC in 30 publications (60%). *Testing* was the second most cited, being used in SPC in 20 publications (40%), followed by *Coding*, which was used in SPC in 12 publications (24%). The *Project Management* process was the object of analysis in 9 publications. *Project Management* is also a suitable process for SPC, because it is usually a critical process (it addresses items such as Budget and Schedule, among other important aspects) and data can be collected frequently. Other processes, such as *Audit*, were cited in only one publication.

Some publications (P07, P20, P26, P28, P31 and P46) refer to *Software Development* as the process used in SPC. Usually, the software development process as a whole (involving requirements development, requirements analysis, design, coding and testing) is not suggested to be controlled by using SPC, since it is too large and SPC is indicated for smaller processes (Tarhan and Demirors 2008; Barcellos et al. 2013). However, although publications cite software development process, measures are in fact related to phases of this process, which are processes suitable for SPC. For instance, the measure *productivity* (P07, P16, P27 and P30) is collected for each task, activity or phase, producing data which is useful to describe the behavior of the requirement development, requirement analysis, design, coding and testing processes.

Considering that small processes are more suitable for SPC, some measures are related to parts of processes. For instance, the measures *ratio of test procedure preparation review effort* and *test procedure preparation productivity* (in P09) are related to the *Testing* process, more specifically to the *Testing Preparation* subprocess.

With regard to measure category, *quality* measures are the most cited (39,02%). This is a consequence of most measures being related to defects, itself directly related to

quality aspects. *Performance* measures are the second most cited (20 measures, 24,39%), particularly the ones related to productivity, which describe process behavior by means of the effort spent and the work done. There is no measure related to size. Size measures are not suitable for use on their own in SPC because they are not able to describe process performance. They are often used to compose other measures able to provide information about process behavior or to evaluate effects of corrective/improvement actions (for example, after using SPC to analyze the coding process behavior and performing actions to improve this process, one could measure product size to evaluate if the actions had any impact on it).

As for goals, some publications explicitly present the goals that motivated SPC use and measure selection. Others do not mention the goals explicitly, but it is possible to infer them from the text. Some publications, however, do not present the goals and it is not possible to deduce them based on the text (e.g., P02, P14, P30, P33, P37 and P48). SPC should be performed to support the monitoring of goals (Florac and Carleton 1999; CMMI Institute 2010; Barcellos et al. 2013). In this sense, it is important to make clear which goals are to be monitored and which measures are to be used for this.

Among the identified goals, some are general, such as *Win the market competition* (P15) and others very specific, such as *Understand the effect of reviews as verification activities in test* (P09). In line with the most cited measures, most goals are related to quality aspects (e.g., *Reduce defects in the products*, *Improve product quality*, *Improve defect detection to reduce the number of delivered defects*). There are several goals involving the understanding of process performance (e.g., *Understand fixing process performance*, *Understand project management process performance*). We noticed that publications citing these goals report cases in which SPC practices were starting to be used. Therefore, the first result expected from SPC was to know the processes' behavior so that it would be possible to improve them.

With respect to measures use, most measures (96.34%) were used in practical initiatives. Only the measures *test effectiveness*, *review preparation rate* and *review rate*, cited in P43, were not applied in a real situation reported in the selected publications. We did not eliminate these measures because the P43 authors argued that they are suitable for SPC and we agree with them.

SPC can be applied in the context of SPI programs or in isolation. In other words, an organization can apply SPC to some processes, aiming to understand and improve their behavior in a particular context or to achieve a certain goal. On the other hand, an organization can apply SPC in the context of models such as CMMI, aiming at a broader process improvement in a SPI program. From the measures identified, 83,54% were used in practical initiatives involving CMMI or ISO 9001. This shows that in the context of software processes, SPC has been used in SPI programs guided by standards or maturity models, particularly CMMI.

3.4 Threats to validity

Every study presents threats to the validity of its results. Threats should be treated as carefully as possible and should be considered together with the results obtained in the study. Following the classification presented by Petersen et al. (2015), we will discuss the main threats to the mapping study results next.

Descriptive validity is the extent to which observations are described accurately and objectively. To reduce descriptive validity threats, a data collection form was designed in order to support data extraction and recording. The form objectified the data collection procedure and could always be revisited. However, due to the lack of clear information with regard to measures, processes and goals in some publications, the collection form is not enough to treat the threat. While some publications present detailed information that answers the research questions, others address the research questions superficially, which may have impacted the researchers' understanding and contributed towards the extraction of inappropriate data. Moreover, the use of ad-hoc procedures for data extraction and refinement impacts the results. Although some steps were defined (e.g., extract the elements; adjust names aiming for clarity; unify equivalent elements; eliminate non-normalized measures; identify sub-goals, etc.), they can be subjective and dependent on the reviewer decisions. With an aim towards minimizing the threat, data extraction and refinement were performed by the first author and reviewed by the second and third authors. Discordances were discussed and resolved.

Theoretical validity is determined by the researcher's ability to capture what is intended to be captured. In this context, one threat concerns the search string, since useful publications may not contain the chosen terms. This threat was dealt with through several tests performed considering control publications until we got the string that was used. In order not to exclude relevant publications, we decided to use a comprehensive string. Moreover, we also minimized this threat through backward snowballing, when relevant publications not captured by the search string were selected. Another threat is regarding the analysis of abstracts during the application of the first filter in the selection of relevant publications. If not properly performed, relevant papers can be discarded. We minimized this threat by performing the analysis from the point of view of different researchers. Thus, a publication was discarded only if all the researchers agreed that it did not satisfy the inclusion criteria. The researcher bias over data extraction and classification is also a threat to theoretical validity. To minimize this threat, data was extracted and recorded by the first author and reviewed by the second and third authors. Another threat to theoretical validity regards the sample of publications used in the study. It is possible that useful publications have not been available in the sources searched. To minimize this threat, we searched seven digital libraries and, after that, performed backward snowballing, providing good coverage for the study. However, since the study object consisted of articles, we did not analyze other types of publications, such as technical reports, dissertations and theses, which could affect the study results.

Finally, *Interpretive validity* is achieved when the conclusions drawn are reasonable given the data obtained. The main threat in this context is the researcher bias over data interpretation. To minimize this threat, interpretation was performed by the first author and reviewed by the others. Discussions were carried out until a consensus was reached. Another important threat regards the subjectivity of the qualitative interpretation and analysis.

Even though we have treated many of the identified threats, the adopted treatments involved human judgment, therefore the threats cannot be eliminated and must be considered together with the study results.

4 Questionnaire

The systematic mapping provided information about measures used in SPC according to literature records. After the mapping study, we applied a questionnaire to three professionals from Brazilian organizations, aiming to identify processes, goals and measures they have used in SPC.

Our goal was to investigate if goals, processes and measures reported by the professionals were also found in the literature.

The participants were professionals with experience in implementing or appraising SPC practices in Brazilian software organizations. We were able to identify six professionals that fit this profile. One of them reported not having access to data required to answer the questionnaire and chose not to answer it based only on his memory. Three professionals reported that they had worked on the same projects. Consequently, their answers were the same and we decided to consider only one of them. Thus, the results consider the answers provided by three professionals.

Measures used in Statistical Process Control in Software Organizations

This questionnaire aims to identify measures that have been used in statistical process control (SPC) in software organizations, as well as processes and goals related to these measures.

Information provided by the participants will be used only for research purposes. The participants identify will be kept anonymous.

1. Participant Information

Please, provide the information bellow. We ask your name and email for us to be able to contact you in case of doubts.

Name: _____

Email: _____

For how many years have you been worked with SPC? _____

Provide information about the organizations in which you have worked with SPC and the role you played (implementer or appraiser): _____

2. Processes, goals and measures used in SPC

Which processes, goals and measures have been used in SPC?

Inform in the table below (following the example) the processes that have been used in SPC, the related goals that have been considered in SPC and the associated measures used in SPC. Use the Comments column to provide further information. Add to the table how many lines you need.

Example:

Process	Goal	Measure	Comments
Project Monitoring and Control	Monitor project schedule and budget performance	Schedule Performance Index (earned value/planned value)	Indexes of Earned Value Analysis
		Cost Performance Index (earned value/actual cost)	

Provide your answers in the table below:

Process	Goal	Measure	Comments

Fig. 5 Form used for data collection

Concerning the participants' profile, one of them (hereafter identified as participant #1) is a member of a CMMI level 5 organization with 6 years' declared experience with SPC. The second participant (participant #2) is a MR-MPS-SW implementer and appraiser who worked as a consultant in 3 organizations successfully evaluated at CMMI level 5. The last participant (participant #3) is also a MR-MPS-SW implementer and appraiser who worked as a consultant in an organization successfully evaluated at CMMI level 5 and in two organizations successfully evaluated at MR-MPS-SW level A.

Figure 5 shows the form used for data collection. The form was sent by email to the participants after they had accepted to participate in the study and was returned to the researcher by the participants after they had filled in the form.

Table 5 summarizes data obtained from the questionnaires answered by the participants. Similarly to the procedure adopted in the systematic mapping, when it came to consolidating data we unified equivalent measures, goals and processes. In the Category column, Q refers to Quality and P to Performance.

Based on the participants' answers, five measures, five goals and seven related processes were identified.

With regard to measure category, three of the cited measures are related to quality and two are related to performance. *Productivity* was the only measure reported by

Table 5 Identified measures, processes and goals

Measure	Process	Goals	Category	Participant
Defect density (<i>number of defects / product size</i>)	Product Requirements Specification	Monitor process quality	Q	#1
	Design			
	Coding			
	Testing			
Productivity (<i>effort / product size</i>)	Product Requirements Specification	Monitor process performance	P	#1, #2
	Coding	Improve productivity		
	Testing			
	Design			
	Requirements Analysis			
Rework (<i>effort spent on rework/ product size</i>)	Requirements Analysis	Improve productivity		#2
	Design			
	Coding			
	Testing			
Quality (<i>number of injected defects / number of fixed defects</i>)	Requirements Analysis	Improve product quality	Q	#2
	Design			
	Coding			
	Testing			
Defect Index (<i>number of fixed defects/ number of defects detected by the client</i>)	Architecture Verification	Monitor defects index		#3
	Scenarios Validation (requirements)			
	Testing			

more than one participant. Participants reported the same measure for several processes. Therefore, the measures provide different information according to the process they related to. For instance, *density defect*, when related to *Product Requirements Specification* refers to defects in product requirements. On the other hand, when it is related to *Design*, it refers to defects in the software design. Similarly, when *rework* is related to *Analysis*, it refers to rework done when performing analysis, while when it is related to *Coding* it refers to coding rework.

With regard to processes, *Testing* was the most cited, having been pointed out by all participants. This process deals directly with product quality (category of most of the cited measures) and is a critical process. Therefore it is a good candidate for SPC. *Design* and *Coding* processes were reported by two participants. Some of the cited processes are, in fact, subprocesses of other processes mentioned. *Product Requirements Specification* and *Scenarios Validation* are subprocesses of *Requirements Analysis*, and *Architecture Verification* is a subprocess of *Design*. At CMMI and MR-MPS-SW high maturity levels, organizations have to select subprocess to SPC, meaning that the processes to be used in SPC should be part of other processes. However, the subprocesses' granularity is not explicitly established. Thus, what is considered a process in an organization may be a subprocess in another. This can be an explanation for the different granularity levels of the processes identified.

None of the measures reported by the participants is related to *Project Management* or *Review* processes. It was expected that measures related to these processes would be cited, since they are critical processes and allow for frequent data collection, which are characteristics of processes suitable for SPC.

With regard to goals, only two were informed by more than one participant (*Improve productivity* and *Monitor process performance*). As with measures, participants defined goals in a general way and related the same goal to several processes. Thus, when related to a specific process, the goal is "specialized" to it. For example, when *Improve productivity* is related to the *Coding* process, it refers to the *Coding* process performance, and when it is related to the *Requirements Analysis* process, it refers to the *Requirements Analysis* process performance.

The questionnaire results show that few measures have been used and they are mainly related to quality and productivity. In addition, measures and goals have been defined in a general way and related to several processes.

The use of few measures might be explained by the fact that for a measure to be used in SPC, data must be frequently collected and analyzed, which often demands more effort than measuring in a traditional way (i.e., without SPC). Thus, organizations might have decided to use few measures to analyze the behavior of processes submitted to SPC. Moreover, it is worth noticing that all participants have SPC experience within the context of maturity models (CMMI and MR-MPS-SW) and have worked in similar small/medium organizations, which may also have contributed to the little diversity in the identified measures.

4.1 Threats to validity

As discussed in the systematic mapping section, when carrying out a study, it is necessary to consider threats to the validity of its results. In this section we discuss some threats involved in the questionnaire.

At first, we can highlight two threats related to repeatability. The first one refers to the ability to repeat the study's behavior with the same participants. The main threat in this context is related to the communication and sharing of information among the participants. To address this threat, the questionnaire was sent to the participants' personal emails, so that they could answer it individually. Additionally, participants were informed that answers should be based on their own experiences in implementing or appraising SPC in software organizations. The second threat can compromise the ability to repeat the study behavior with different participants. Although we have tried to address this threat by selecting participants with different profiles, the participants' profile is homogeneous and the number of participants is very small. Therefore it is possible that other participants, with different profiles or experience in different organizations, could give different answers. Also, since the selection of processes and measures for SPC is directly related to an organization's goals, organizations with different goals can submit different processes and use different measures, which could also lead to different results.

With regard to the quality of the answers provided by the participants, there was the threat of the participants not providing correct information. To address this threat, we provided examples of information that should be included in the questionnaire, so that the participants could better understand how to answer it. Moreover, in order to avoid answers not reflecting the reality due to personal expectations or concern about being judged for his/her answers, participants were informed that the study did not represent any personal assessment and their identities would be kept in confidence.

In summary, due to the small number of participants and their homogeneous profile, the results found in the questionnaire are preliminary results and cannot be generalized.

5 Consolidated view of the findings

In this section, we present some discussions involving the systematic mapping and questionnaire results, with a view to providing a consolidated view of the results obtained in both studies.

In both studies, measures related to quality and performance were the most cited. Also, there is a predominance of defect-related measures. Three measures (shown in Table 6) were found in both questionnaire and systematic mapping. Considering both of the studies, 84 different measures were identified.

With regard to processes, the systematic mapping identified 12 processes and in the questionnaire, seven processes were cited. *Review* was the most cited process in the systematic mapping, while *Testing* was the most cited in the questionnaire. *Requirements Analysis*, *Design*, *Coding* and *Testing* were identified in both systematic mapping and the questionnaire. The other three processes reported in the questionnaire were not

Table 6 Measures identified in both studies

Systematic Mapping	Questionnaire
Defect density (number of detected defects/ product size)	Defect density (<i>number of defects / product size</i>)
Productivity (<i>effort /product size</i>)	Productivity (<i>effort / product size</i>)
Defect injection rate (by phase) (number of injected defects/number of removed detected defects)	Quality (<i>number of injected defects / number of fixed defects</i>)

Table 7 Requirements for a measure to be used in SPC (Barcellos et al. 2013)

R1. The measure must be aligned to organizational or project goals.
R2. The measure must be able to support decision making.
R3. The measure must be able to support software process improvement.
R4. The measure must be related to a critical process.
R5. The measure must be able to describe the process behavior.
R6. The measure must have appropriate granularity level.
R7. The operational definition of the measure must be correct and satisfactory.
R8. The correlated measures to the measure must be defined.
R9. The measure must be correctly normalized (if applicable).
R10. It must be possible to normalize the measure (if applicable).
R11. The criteria for grouping data for analysis must be defined.
R12. Data collected for the measure must include context information.
R13. Data collected for the measure must be accessible and retrievable.
R14. The measure should not consider aggregate data (or if it does, it should be possible to disaggregate them).
R15. Data collected for the measure must be consistent.
R16. Data collected for the measure must be precise.
R17. The amount of collected data must be enough for applying SPC techniques.

explicitly identified in the literature, but they can be considered part of other processes. The *Product Requirements Specification* and *Scenario Validation* processes reported in the questionnaire can be understood to be part of the *Requirements Development* process found in the literature. The *Architecture Verification* process can be understood as part of the *Design* process. Although these processes are part of processes identified in the literature, we can consider them different processes, since it is possible to submit a process (e.g., Requirements Development) or a subprocess (e.g., Requirements Specification) to SPC. Thus, considering the mapping and questionnaire results, 15 processes were identified. Three of them are subprocesses of others.

As for goals, from the five goals identified in the questionnaire, the goals *Improve product quality* and *Improve productivity* were also identified in the literature. Thus, in total, 47 different goals were identified. In the set of goals identified in the literature, there are general and specific goals. On the other hand, all goals reported in the questionnaire are general and related to several processes.

Some of the goals, processes and measures reported in the questionnaire were not identified in the literature. This can be seen as a sign that there are goals, processes and measures used in practice that are not recorded in the literature. However, it is important to reinforce that the majority of the measures found in the literature were used in some SPC practical application.

The set of measures, processes and goals produced as a result of the studies performed provides knowledge about measures used in SPC initiatives and can be useful for organizations to define measures to SPC. However, to use a measure in SPC, some criteria should also be observed. Barcellos et al. (2013) defined a set of requirements that should be considered when selecting measures to be used in SPC. Table 7 summarizes some of them. Some of the requirements are satisfied by the measures identified in the studies (e.g., R5). Others depend on the organization that will use the measures. For instance, to

meet R1, organizations must select measures aligned to their goals, and to meet R7, when selecting a measure, an organization must establish its operational definition.

6 Conclusion

This paper presented the results of an investigation about measures for SPC, the related process and goals supported by them. To investigate the state of the art, a systematic mapping was performed. After that, a questionnaire was answered by three professionals from Brazilian software organizations. As the main result of the studies, 84 measures, 47 objectives and 15 related processes were identified.

Before performing the systematic mapping, we investigated the literature looking for secondary studies about measures for SPC. We did not find any, and then we decided to perform the studies reported in this paper. Although there is no systematic study investigating measures for SPC, we can cite the work performed by Monteiro and Oliveira (2011), which presents a catalog of measures for process performance analysis. However, although they claim to have carried out a broad literature review, they did not follow a systematic approach. Besides, measure category, measurement goals and information about the use of the measures in practical initiatives were not investigated in their study.

According to Kitchenham et al. (2011), systematic mappings provide an idea of shortcomings in existing evidence, which becomes a basis for future studies. Practical findings, in turn, allow for technique improvement or other proposals (Easterbrook et al. 2008). In this sense, the results obtained in the studies addressed in this paper point to gaps and improvement opportunities in the SPC context for software organizations. The results showed us that SPC has focused on defect-related measures and processes, despite there being many other processes that could be explored and improved by using SPC techniques. Moreover, we noticed a lack of concern with correlated measures that are necessary to support root cause investigation when analyzing process behavior. We also noticed that although measures are cited, their operational definitions are not addressed. Even basic information about the measures (e.g., how often data are collected) is not presented in the publications. Clear and unambiguous operational definitions are crucial in order to get consistent measurements, an important requirement in SPC context (Barcellos et al. 2013).

In this work we have limited ourselves to presenting the literature and questionnaire findings. Therefore, although it would be possible to infer that a certain measure is related to other goals or processes than the ones we found in the studies, we did not do that. Qualitative techniques could have been used to analyze data presented in this paper and identify other relations between goals, processes and measures, as well as relations between different measures (e.g., a measure may need another to provide information about a certain goal). As an ongoing work, we have been analyzing the findings aiming to get new information from them, such as new relations between goals, which processes (besides the ones identified in the study) could be measured by the identified measures, which subprocesses could be identified from the processes considering the related measures, which measures could be used in a combined way to support measurement goals, and so on.

Our purpose in this work was to provide a comprehensive set of measures for SPC relevant for academics who want to investigate this subject and for professionals who want a basis to help them to define measures for SPC. However, we are aware that it may be not functional to look for measures in a large table or even in a catalogue. Thus, considering our understanding resulting from the mapping study and aiming to strengthen the reuse

1 Appendix 1

Table 8 Selected Publications

ID	Title
P01	Kumari, KS, Amulya, B, Prasad, RS (2014) Comparative study of pareto type II with HLD in assessing the software reliability with order statistics approach using SPC. <i>International Conference on Circuits, Power and Computing Technologies (ICCPCT)</i> , p 1630–1636.
P02	Alhassan, MA, Jawawi, DNA (2014) Sequential strategy for software process measurement that uses Statistical Process Control. In: 8th Malaysian Software Engineering Conference (MySEC), p 37–42.
P03	Vashisht, V (2014) Enhancing Software Process Management through Control Charts. <i>Journal of Software Engineering and Applications</i> , Feb, p 87–93.
P04	Fernandez-Corralles, C, Jenkins, M, Villegas, J (2013) Application of Statistical Process Control to Software Defect Metrics: An Industry Experience Report. <i>ACM / IEEE International Symposium on Empirical Software Engineering and Measurement</i> , p 323–33.
P05	Abubakar, AM, Jawawi, DNA (2013) A Study on Code Peer Review Process Monitoring using Statistical Process Control. <i>Software Engineering Postgraduates Workshop (SEPoW)</i> , p 136–141.
P06	Pandian, SSA, Puthiyannayagam, P (2013) Control Charts for Improving the Process Performance of Software Development Life Cycle. <i>Ijrras</i> , vol. 14, no. February, p 248–256.
P07	Tarhan, A, Demirors, O (2012) Apply quantitative management now. <i>IEEE Software</i> , v. 29, n. 3, p 77–85.
P08	Tarhan, A, Demirors, O (2011) Assessment of software process and metrics to support quantitative understanding: Experience from an undefined task management process. <i>Communications in Computer and Information Science (CCIS)</i> , v. 155, p 108–120.
P09	Tarhan, A, Demirors, O (2011) Investigating the effect of variations in the test development process: A case from a safety-critical system. <i>Software Quality Journal</i> , v. 19, n. 4, p 615–642.
P10	Schneidewind, N (2011) What can Software Engineers Learn from Manufacturing to Improve Software Process and Product? <i>Intelligent Information Management</i> , 01, p 98–107. https://doi.org/10.4236/iim.2009.12015
P11	Vijaya, G, Arumugam, S (2010) Monitoring the stability of the processes in defined level software companies using control charts with three sigma limits. <i>WSEAS Trans. Info. Sci. and App.</i> , 7(10), p 1200–1209. Retrieved from http://portal.acm.org/citation.cfm?id=1865374.1865383
P12	Zhang, H, Kim, S (2010) Monitoring software quality evolution for defects. <i>IEEE Software</i> , 27, 58–64. https://doi.org/10.1109/MS.2010.66
P13	Selby, RW (2009) Statistical Process Control for System Development Using Six Sigma Techniques. <i>AIAA SPACE Conference & Exposition</i> , Sept.
P14	Zhao, F, Peng, X, Zhao, W (2009) Software development process monitoring based on nominal transformation. <i>Eighth IEEE/ACIS International Conference on Computer and Information Science</i> , p 983–988.
P15	Zhu, M, Liu, W, Hu, W, Fang, Z (2009) Target Based Software Process Evaluation Model and Application. <i>Second International Conference on Information and Computing Science (ICIC)</i> , v. 1, p 107–110.
P16	Gou, L, Wang, Q, Yuan, J, et al. (2009) Quantitative defects management in iterative development with BiDefect. <i>Software Process: Improvement and Practice - Addressing Management Issues</i> , v. 14, n. 4, p 227–241.
P17	Tarhan, A, Demirors, O (2008) Assessment of Software Process and Metrics to Support Quantitative Understanding. <i>IWSM-Mensura</i> , p 102–113
P18	Weller, EF, Card, D (2008) Applying SPC to Software Development: Where and Why. <i>IEEE Software</i> , v. 25, p 48–50.
P19	Wang, Q, Gou, L, Jiang, N, et al. (2008) Estimating fixing effort and schedule based on defect injection distribution. <i>Software Process Improvement and Practice</i> , v. 11, p 361–371.
P20	Chang, C-P, Chu, C-P (2008) Improvement of causal analysis using multivariate statistical process control. <i>Software Quality Journal</i> , v. 16, n. 3, p 377–409.
P21	Card, DN, Domzalski, K, Davies, G (2008) Making Statistics Part of Decision Making in an Engineering Organization. <i>IEEE Software</i> , v. 25, n. 3, p 37–47.
P22	Baldassarre, MT, Caivano, D, Visaggio, G (2006) Non invasive monitoring of a distributed maintenance process. <i>Conference Record - IEEE Instrumentation and Measurement Technology Conference</i> , no. April, p 1098–1103.
P23	Wang, Q, Jiang, N, Gou, L, et al. (2006) BSR: A statistic-based approach for establishing and refining software process performance baseline. <i>Proceedings of the 28th international conference on Software engineering</i> , p. 585–594.
P24	Komuro, M (2006) Experiences of applying SPC techniques to software development processes. In: 28th international conference on Software engineering - ICSE, p 577.

Table 8 Selected Publications (*Continued*)

ID	Title
P25	Tarhan, A, Demirors, O (2006) Investigating suitability of software process and metrics for statistical process control. <i>Software Process Improvement</i> , vol. 4257, p 88–99.
P26	Zhang, Y, Sheth, D (2006) Mining software repositories for model-driven development. <i>IEEE Software</i> , v. 23, n. 1, p 82–90
P27	Boffoli, N (2006) Non-intrusive monitoring of software quality. <i>Proceedings of the European Conference on Software Maintenance and Reengineering</i> , p 319–322.
P28	Sargut, KU, Demirors, O (2006) Utilization of statistical process control (SPC) in emergent software organizations: Pitfalls and suggestions. <i>Software Quality Journal</i> , vol. 14, no. 2, p 135–15.
P29	Wang, Q, Jiang, N, Gou, L, Che, M, Zhang, R (2006) Practical experiences of cost/schedule measure through earned value management and statistical process control. <i>Software Process Change</i> , p 348–354.
P30	Wang, Q, Li, M (2005) Measuring and improving software process in China. <i>International Symposium on Empirical Software Engineering</i> , p 177–186.
P31	Baldassarre, MT, Boffoli, N, Caivano, D, Visaggio, G (2005) Improving dynamic calibration through statistical process control. <i>21st IEEE International Conference on Software Maintenance (ICSM'05)</i> , p 273–282.
P32	Narayana, V, Swamy, R (2003) Experiences in the inspection process characterization techniques. In: <i>International Conference on Quality Software</i> , n. Jan, p 388–395.
P33	Eickelmann, N, Anant, A (2003) Statistical process control: What You Don't Measure Can Hurt You! <i>IEEE Software</i> , vol. 20, no. 2, p 49–51.
P34	Jacob, AL, Pillai, SK (2003) Statistical process control to improve coding and code review. <i>IEEE Software</i> , v. 20, n. 3, p 50–55.
P35	Sargut, KU; Demirors, O (2003) Utilization of Defect Density Metric for SPC Analysis. In: <i>13th International Conference on Software Quality</i> , no. Cmm.
P36	Bertolino, A, Marchetti, E, Mirandola, R, Lombardi, G and Peciola, E (2002) Experience of applying statistical control techniques to the function test phase of a large telecommunications system. <i>IEEE Software</i> , v. 149, n. 4, p 349–357.
P37	Jalote, P, Saxena, A (2002) Optimum control limits for employing statistical process control in software process. <i>IEEE Transactions on Software Engineering</i> , vol. 28, no. 12, p 1126–1134, 2002.
P38	Florence, A (2001) CMM Level 4 Quantitative Analysis and Defect Prevention. <i>Crosstalk</i> , Feb, p. 20–21.
P39	Mohapatra, S, Mohanty, B (2001) Defect Prevention through Defect Prediction: A Case Study at Infosys. <i>Proceedings. IEEE International Conference on Software Maintenance</i> , p 260–272.
P40	Weller, EF (2000) Practical applications of statistical process control. <i>IEEE Software</i> , v. 17, n. 3, p 48–55.
P41	Florac, WA, Carleton, AD, Barnard, JR (2000) Statistical Process Control: Analyzing a Space Shuttle Onboard Software Process. <i>IEEE Software</i> , vol. 17, no. 4, p 97–106.
P42	Paulk, M (2000) Applying SPC to the personal software process. <i>10th Intl. Conf. Software Quality</i> , p 77–87.
P43	Jalote, P, Dinesh, K, Raghavan, S, Bhashyam, MR, Ramakrishnan, M (2000) Quantitative Quality Management through Defect Prediction and Statistical Process Control. <i>Proceedings of Second World Quality Congress for Software</i> .
P44	Keeni, G (2000) The Evolution of Quality Processes at Tata Consultancy Services. <i>IEEE Software</i> , v. 17, n. 4, p 79–88.
P45	Hong, GY, Xie, M, Shanmugan, P (1999) A statistical method for controlling software defect detection process. <i>Computers & Industrial Engineering</i> , vol. 37, p 137–140.
P46	Lewis, NDC (1999) Assessing the evidence from the use of SPC in monitoring, predicting & improving software quality. <i>24th International conference on computers and industrial engineering</i> , vol. 37, no. 1–2, p 157–160.
P47	Hayes, W (1998) Using a Personal Software Process(SM) to improve performance. <i>Proceedings Fifth International Software Metrics Symposium</i> , p 61–71.
P48	French, VA (1995) Applying software engineering and process improvement to legacy defense system maintenance: An experience report. <i>International Conference on Software Maintenance</i> , p 337–343.
P49	Card, D (1994) Statistical Process Control for Software? <i>IEEE Software</i> , v. 11, n. 3, p 95–97.
P50	Card, DN, Berg, RA (1989) An industrial engineering approach to software development. <i>Journal of Systems and Software</i> , v. 10, n. 3, p 159–168.

of the identified measures, we have been working on a pattern-based approach to support measure selection for SPC initiatives. As a result, we have developed MePPLa, a Measurement Planning Pattern Language (Brito et al. 2017) built on the basis of the findings presented in this paper. MePPLa provides a set of goals, processes and measures (with detailed operational definitions) suitable for SPC and a mechanism to support measures selection according to the goals to be achieved.

7 Endnotes

¹MR-MPS-SW (Montoni et al. 2009) is a Brazilian reference model for software process improvement that, like CMMI-Dev (CMMI Institute 2010), addresses process improvement in levels, ranging from G level (lowest) to A level (highest). In MR-MPS-SW, levels A and B are equivalent to CMMI-Dev levels 5 and 4, respectively.

²In this work, we use the term “measure” in conformance to ISO/IEC 15939 (ISO/IEC 2007), i.e., a variable to which a value is assigned as the result of measurement.

³In this work, SPC initiatives denote cases of SPC use in practice.

8 Appendix 2

Tables 9, 10, 11, 12 and 13 present the resulting set of processes, measures and goals. The publications from which data were extracted are also identified. The formulas presented in the table were extracted from the publications. Thus, they reflect the way the measure is calculated according to the source publication. Measures preceded by ^a were used in initiatives involving standards/maturity models. Measures preceded by ^b were used in initiatives not involving standards/maturity models. In the table, when a measure is related to a process/goal, it means that at least one publication cited the measure related to the process/goal. In the tables goals are referred to the id provided in Table 4. When the goal is “-”, it means that it was not possible to identify goals in the publications that cite the measure.

Table 9 Measures, processes and goals – Category: TIME

Measures	Publications	Processes	Goals
^a Defect age (defect resolution date – defect creation date + 1)	P07,P17	Fixing	G10.1
^a Duration estimation accuracy (actual duration / estimated duration)	P08,P14,P15, P23,P30,P44	Project Management	G09.1;G09.5; G10.3;G15
^a Schedule performance index (budget cost for performed work / budget cost for scheduled work)	P20,P29	Project Management	G09.3
^b Code size estimation accuracy (actual code size/estimated code size)	P15	Project Management	G15
^b File estimation accuracy (actual number of files/estimated number of files)	P15	Project Management	G15
^a Actual procurement time (start date for joining a new member to a project - start date of recruitment process + 1)	P07,P17	Recruitment	G10.4
^a Procurement time variance (actual procurement time - planned procurement time)	P07,P17	Recruitment	G10.4
^a Review duration (review closure date – review opening date)	P07,P17,P25	Review	G06;G10.2
^a Review duration per defect (sum of reviews' duration / number of detected defect)	P07,P17,P25	Review	G06;G10.2
^a Time spent on review preparation per reviewer (sum of time spent on review preparation by each reviewer/ number of reviewers)	P34	Review	G02

Table 10 Measures, processes and goals – Category: PERFORMANCE

Measures	Publications	Processes	Goals
^a Rework efficiency (number of fixed defects / defect fixing effort)	P16,P18,P40	Fixing	G08.2;G05.1; G05.9
^a Inspection performance (size of inspected product/ effort spent on the inspection)	P11,P28	Review	–
^b Inspection productivity (number of detected defects in the inspection/ effort spent on the inspection)	P48	Review	–
^a Productivity (effort/product size)	P03,P07,P16, P19,P20,P30, P31,P27	Maintenance, Requirements Analysis, Design, Coding, Testing	G01;G03; G09.1;G05.2; G08;G10
^b Defect closure date variance (actual defect closure date – planned defect closure date)	P13	Review	G05.6
^a Review performance (effort spent on the review /size of the reviewed product)	P07	Review	G10
^a Defect resolution efficiency (number of solved defects /resolution effort)	P07,P17,P25	Review	G06;G10.2
^a Effective preparation speed (product size/average time spent preparing for review + average time spent preparing for rereview)	P34	Review	G02
^a Effective review speed (product size/ time spent on all reviews of the product)	P34	Review	G02
^a Preparation speed (product size/average time spent preparing for review)	P34	Review	G02
^a Review speed (product size/spent time on review)	P03,P24,P34	Review	G02;G04.3
^a Test defect detection efficiency (number of defects in tests / effort spent reviewing tests)	P07,P17	Review	G13
^a Defect detection efficiency (number of defects detected in the review/detection effort)	P07,P17, P25,P19	Review, Testing	G06;G10.2; G03
^a System test speed (size of the tested product/time spent on system test)	P07	Testing	G10.5
^a Test defect resolution efficiency (number of resolved defects / effort spent resolving the defects)	P07,P17	Testing	G13
^a Test design productivity (number of test cases designed / test cases design effort)	P07,P09,P17	Testing	G12
^a Test execution productivity (number of test cases executed/ effort spent on executing test cases)	P09	Testing	G12
^a Test procedure preparation productivity (number of test cases/effort spent on test procedure preparation)	P09	Testing	–
^a Test script productivity (number of test cases/effort spent developing test script)	P07,P17	Testing	G13
^a Unit test speed (size of the tested product/time spent on unit test)	P07	Testing	G10

Table 11 Measures, processes and goals – Category: EFFORT

Measures	Publications	Processes	Goals
^a Fixing effort percentage (effort spent on defect fixing activities/ project total effort * 100%)	P16	Fixing	G07
^a Defect resolution effort ratio (effort spent on defect resolution/ number of solved defects)	P07	Fixing	G10.1
^a Inspection preparation rate (size of the product to be inspected/time spent to prepare the inspection)	P18,P21, P26,P37, P40	Review	G05.1;G04.3; G06.1
^a Effort estimation accuracy (actual effort / estimated effort)	P07,P15,P44	Project Management	G09.1;G09.5; G10.3;G15
^a Task effort estimation accuracy (actual task effort / estimated task effort)	P17	Project Management	G10.3
^a Task effort variance (estimated task effort - actual - task effort)	P17	Project Management	G10.3
^a Percentage of effort saved due to process automation (effort saved due to process automation/project effort *100)	P44	Project Management, Audit, Testing	G09.1
^a Review effort per defect (effort spent on the review / number of defects detected in the review)	P09	Review	–
^a Total review effort (peer review effort + review effort)	P09	Review	G11
^a Percentage of effort spent on system tests (effort spent at system tests activities/ total effort of project * 100%)	P16	Testing	G07
^a Ratio of test design review effort (test design review effort / test design effort)	P07,P09, P17	Testing	G12
^a Ratio of test development review effort (test development review effort/test development effort)	P09	Testing	G12
^a Ratio of test procedure preparation review effort (effort spent on review of test procedure preparation /effort spent on test procedure preparation)	P09	Testing	–
^a System test effort estimation accuracy (system test actual effort/ system test estimated effort)	P07	Testing	G10.5
^a Test script review effort (effort spent reviewing test scripts / effort spent developing the reviewed test scripts)	P07,P17	Testing	G13

Table 12 Measures, processes and goals – Category: COST

Measures	Publications	Processes	Goals
^b Cost of poor quality in the code (cost of correcting internal failure + cost of correcting external failure)	P26	Coding	G09.4
^b Cost of code quality (cost of appraisal + cost of defect prevention + cost of correcting internal failure + cost of correcting external failure)	P26	Coding	G09.4
^a Cost performance index (budget cost for performed work / actual cost for performed work)	P20,P29	Project Management	G09.3
^b Cost estimation accuracy (actual cost / estimated cost)	P15	Project Management	G15
^b Cost performance accumulated (budget cost for scheduled work / Spent effort)	P20	Project Management	G09.3

Table 13 Measures, processes and goals - Category: QUALITY

Measures	Publications	Processes	Goals
^a Defect injection distribution (by Requirements Development, Design, Coding, and Testing)(defects injected in requirements (or design, coding and testing) / all defects removed in system testing * 100%)	P16	Coding, Design, Requirements Development, Testing	G07
^a Defect removal effectiveness (by Requirements Development, Design, Coding, and Testing)(number of removed defects in requirements (or design, coding and testing) /number of detected defects)	P16,P43	Coding, Design, Requirements Development, Testing	G04.4
^a Defect injection rate (by Requirements Development, Design, Coding, and Testing)(number of injected defects/ number of removed detected defects)	P06,P15,P16, P40,P43,P49	Coding, Design, Requirements Development, Testing, Review	G08.1;G08.3; G05.5;G05.7; G05.9;G15
^b Defects detection rate per Inspection (number of detected defects/number of performed inspections)	P46	Review	–
^b Inspection effectiveness (<i>number of detected defects in inspections/number of delivered defects</i>)	P26,P47	Review	–
^a Inspection rate (inspected product size/ time spent on the inspection)	P18,P21,P26, P32,P37,P40, P41,P42	Review	G04.1;G05.1; G05.2;G04.3; G06.1
^a Percentage of defects caused by faulty logic (number of defects caused by faulty logic /total number of detected defects *100)	P04	Review, Testing	G09.1;G05.4; G09
^a Percentage of defects found in operation (number of defects found in operation/total number of detected defects *100)	P04	Review, Testing	G09.1;G05.4; G09
^a Percentage of high severity defects detected in production (number of high severity defects detected in production /total number of detected defects *100)	P04	Review, Testing	G09.1;G05.4; G09
^a Percentage of high severity defects identified in testing (number of high severity defects identified in testing/total number of detected defects *100)	P04	Review, Testing	G09.1;G05.4; G09
^a Percentage of rejected defects (number of rejected defects/total number of detected defects *100)	P04	Review, Testing	G09.1;G05.4; G09
^a Rework percentage (rework effort/total effort *100)	P03,P07, P11,P28	Requirements Analysis, Design, Coding, and Testing	G09.2;G10
^b Completed task non-conformance density (number of completed tasks/ number of defects of all completed tasks)	P20	Requirements Analysis, Design, Coding, and Testing	G05
^b Expected task non-conformance density (number of tasks expected to be completed/ number of defects of all tasks)	P20	Requirements Analysis, Design, Coding, and Testing	G05
^a Requirements change rate (number of changed requirements /total number of requirements)	P23	Requirements Management	G05.8
^a Defect density (number of detected defects/product size)	P02,P05,P07, P11,P13,P15, P16,P18,P19, P20,P21,P23, P24,P26,P28, P30,P31,P32, P33,P34,P35, P36,P37,P38, P40,P42,P43, P44,P45,P46, P48,P49,P50	Review, Coding, Maintenance, Requirements Analysis, Design, Testing	G02;G05.1; G08.4;G03; G05;G04.3; G04;G05.6; G10;G05.9; G05.10; G04.2;G15
^a Review Efficiency (number of detected defects/ effort spent on the review)	P24	Review	G04.3
^a Review effectiveness (number of defects detected in reviews/total number of defects)	P24,P44	Review	G04.3

Table 13 Measures, processes and goals - Category: QUALITY (*Continued*)

Measures	Publications	Processes	Goals
Review preparation rate (<i>size of the product to be reviewed/time spent to prepare the review</i>)	P43	Review	G06
Review rate (<i>size of the reviewed product/ time spent on the review</i>)	P43	Review	G06
^a Non-conformance density in system test (number of non-conformances in system tests found by verification and validation/number of system tests reviewed by verification and validation)	P07	Testing	G10.5
^a Density of defect in test (number of defects detected in test/ number of test cases reviewed)	P07,P09,P17	Testing	G13
^a Delivered defect density (defects detected after product release / product size)	P16,P43,P50	Testing	G05;G05.1
^a System test effectiveness (number of defects detected in system test / total number of detected defects)	P07	Testing	G10.5
^a System test verification and validation effectiveness (number defects found in system test by review / number of defects found in system test by all sources)	P07	Testing	G10.5
^a Test efficiency (number of defects detected in tests/ number of defects detected in tests + delivered defects)	P16,P50	Testing	G08.4
^a Test non-conformance density (number of non-conformance in test found by verification and validation /number of tests reviewed by verification and validation)	P07	Testing	G10;G10.5
^a Test verification and validation effectiveness (number of defects found in test by review/ total number of defects in test found by all sources)	P07	Testing	G10.5
^a Unit test effectiveness (number of defects detected in unit test / total number of defects)	P07,P26	Testing	G10;G14
^a Unit test verification and validation effectiveness (number of defects in unit tests found by review/number of defects in unit tests found by all sources)	P07	Testing	G10
^b Mean time between failures	P01	Testing	G05.3
Test effectiveness (number of defects detected in tests / total number of detected defects)	P43	Testing	–

Acknowledgments

We acknowledge the financial support of Brazilian Research Funding Agency CNPq (Processes 485368/2013-7 and 461777/2014-2). Authors also thank FAPERJ (projects E-26/210.643/2016, E- 211.174/2016) and UNIRIO (grant PQ-UNIRIO 01/2016 and 01/2017) for the financial support.

Funding

This research is funded by the Brazilian Research Funding Agency CNPq (Processes 485368/2013-7 and 461777/2014-2), FAPERJ (projects E-26/210.643/2016, E- 211.174/2016) and UNIRIO (grant PQ-UNIRIO 01/2016 and 01/2017).

Availability of data and materials

Please contact author for data requests.

Authors' contributions

DB planned the studies and conducted them. MB and GS planned the studies and review publications selection and data extraction. All the authors worked on the paper writing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 26 April 2017 Accepted: 23 August 2018

Published online: 18 September 2018

References

- Barcellos MP, Falbo RA, Rocha AR (2010) Establishing a well-founded conceptualization about software measurement in high maturity levels. In: 7th international conference on the quality of information and communications technology, pp 467–472
- Barcellos MP, Falbo RA, Rocha AR (2013) A strategy for preparing software organizations for statistical process control. *J Braz Comput Soc* 19:445–473
- Basili VR, Rombach HD, Caldiera G (1994) Goal Question Metric Approach. *Encyclopedia of Software Engineering*. Wiley, Hoboken
- Brito DF, Barcellos MP (2016) Measures suitable for SPC: a systematic mapping. XV Brazilian Symposium on software quality. Maceió – AL, Brazil
- Brito DF, Barcellos MP, Santos G (2017) A software measurement pattern language for measurement planning aiming at SPC. XVI Brazilian Symposium on Software Quality, RJ, Brazil
- Card DN, Domzalski K, Davies G (2008) Making statistics part of decision making in an engineering organization. *IEEE Softw* 25(3):37–47
- CMMI Institute (2010) CMMI for Development, Version 1.3. Carnegie Mellon University, Pittsburgh
- Easterbrook S, Singer J, Storey M, Damian D (2008) Selecting empirical methods for software engineering research. In: Shull F, Singer J, Sjøberg DIK (eds) *Guide to Advanced Empirical Software Engineering*. Springer, London, pp 285–311
- Fagan ME (1976) Design and code inspections to reduce errors in program development. *IBM Systems J* 15(3):182–211 (Ch. 22)
- Fenton NE, Neil M (2000) Software metrics: Roadmap. In: *Conf Futur Softw Eng - ICSE'00*, pp 357–370. <https://doi.org/10.1145/336512.336588>
- Fenton NE, Pfeleeger SL (1997) *Software metrics: a rigorous and practical approach*. PWS Publishing Company, Boston
- Florac WA, Carleton AD (1997) *Measuring the software process: statistical process control for software process improvement*. Addison Wesley, Boston
- Florac WA, Carleton AD (1999) *Measuring the software process: statistical process control for software process improvement*. Addison Wesley, Boston
- Fuggetta A (2000) Software process: a roadmap. *Proceedings of the Conference on The Future of Software Engineering*, pp 25–34
- Ghapanchi, AH, Aurum, A (2011) Measuring the effectiveness of the defect-fixing process in open source software projects. *Proceedings of the 44th Hawaii international conference on system sciences*
- ISO 9001 (2015) *Quality management systems — Requirements*
- ISO/IEC (2007) ISO/IEC15939—Systems and Software Engineering—Measurement Process
- ISO/IEC (2008) ISO/IEC12207—Systems and software engineering — Software life cycle processes
- ISO/IEC (2017) ISO/IEC12207—Systems and software engineering — Software life cycle processes
- Kitchenham B, Brereton P (2013) A systematic review of systematic review process research in software engineering. *Inf Softw Technol* 55(12):2049–2075
- Kitchenham B, Charters S (2007) *Guidelines for performing systematic literature reviews in software engineering*. EBSE 2007-001. Keele University and Durham University Joint Report, UK
- Kitchenham B, Budgen D, Brereton OP (2011) Using mapping studies as the basis for further research – A participant-observer case study, *Information and Software Technology*, Volume 53, Issue 6, pp. 638–651, Butterworth-Heinemann Newton, MA
- Komuro M (2006) Experiences of applying SPC techniques to software development processes. In: 28th international conference on software engineering - ICSE, p 577
- McGarry J, Card D, Jones C et al (2002) *Practical software measurement: objective information for decision makers*. Addison Wesley, Boston
- Monteiro LFS, Oliveira KMD (2011) Defining a catalog of indicators to support process performance analysis. *Journal of Software Maintenance and Evolution: Research and Practice*, Volume 23, Issue 6, pp. 395–422, Wiley, New York.
- Montoni M, Rocha AR, Weber KC (2009) MPS.BR: a successful program for software process improvement in Brazil. *Softw Process Improv Pract* 14:289–300
- Myers GJ (2004) *The art of software testing*, 2nd edn. Wiley, Hoboken
- Petersen K, Vakkalanka S, Kuzniarz L (2015) Guidelines for conducting systematic mapping studies in software engineering: an update. *Inf Softw Technol* 64:1–18
- PMI (2012) *A guide to the Project Management body of knowledge*, 5th edn
- Solingen R, Berghout E (1999) *The goal/question/metric method: a practical guide for quality improvement of software development*. McGraw-Hill Publishing Company, New York
- Sommerville I (2006) *Software engineering*, 8th edn. Addison-Wesley, Boston
- Takara A, Bettin AX, Toledo CMT (2007) Problems and pitfalls in a CMMI level 3 to level 4 migration process, *6th International Conference on the Quality of Information and Communications Technology (QUATIC)*, pp 91–99
- Tarhan A, Demirors O (2008) Assessment of Software Process and Metrics to Support Quantitative Understanding. In: Cuadrado-Gallego JJ, Braungarten R., Dumke R.R., Abran A. (eds) *Software Process and Product Measurement*. Mensura 2007, IWSM 2007. Lecture Notes in Computer Science, vol 4895. Springer, Berlin
- Tarhan A, Demirors O (2012) Apply quantitative management now. *IEEE Softw* 29(3):77–85
- Vijaya G, Arumugam S (2010) Monitoring the stability of the processes in defined level software companies using control charts with three sigma limits. *WSEAS Trans Info Sci And App* 7(10):1200–1209 Retrieved from <http://portal.acm.org/citation.cfm?id=1865374.1865383>
- Wang Q, Gou L, Jiang N et al (2008) Estimating fixing effort and schedule based on defect injection distribution. *Softw Process Improv Pract* 11:361–371
- Wang Q, Li M (2005) Measuring and improving software process in China, *International Symposium on Empirical Software Engineering*, pp 177–186
- Wheeler DJ, Chambers DS (1992) *Understanding Statistical Process Control*. 2nd ed. Knoxville,TN - SPC Press