Measures Suitable for SPC: A Systematic Mapping

Daisy Ferreira Brito, Monalessa Perini Barcellos

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

{dfbrito, monalessa}@inf.ufes.br

Abstract. *The growing interest of organizations in improving their software processes has lead them to aim at achieving the high maturity, where statistical process control (SPC) is demanded. Through SPC is possible to know processes behavior, predict their performance in future projects and monitor them in order to meet the stablished goals. One of the challenges to perform SPC is the selection of measures suitable for it. Although the literature suggests measures to be used in SPC, information is dispersed. Aiming to provide a consolidated set of measures useful for SPC, as well as the related processes and goals supported by the measures, we conducted a mapping study. This paper presents the study and discusses the main findings.*

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) [SEI 2010] and MR-MPS-SW (Reference Model for Brazilian Software Process Improvement) [Softex 2016], propose a SPI implementation in levels. At the highest levels (CMMI levels 4 and 5 and MR-MPS-SW levels B and A) SPC is required.

SPC was originally developed in the manufacturing area, aiming to support improvement programs. It is used to determine if a process is under control, considering the statistical point of view. In the context of software organizations, the use of SPC is more recent and there are still some issues to be explored [Card *et al.* 2008].

Real cases of SPC implementation in software organizations have revealed many problems that affect the successful implementation of SPC [Barcellos *et al.* 2013; Takara *et al.* 2007]. The unsuitability of the defined measures and collected data is one of the main problems, since it postpones SPC practices until proper measures are identified and suitable data are collected [Barcellos *et al.* 2013; Kitchenham *et al.* 2007; Takara *et al.* 2007]. In the literature there are several works presenting measures that can be applied to SPC or reporting the use of measures in SPC initiatives. However, information is spread, making the access difficult and sometimes inefficient.

Considering that, we investigated the literature aiming to identify a set of measures that can be applied in SPC initiatives. In order to ensure study comprehensibility and repetitivity, as well as to reduce the researchers influence on the results, we adopted a systematic approach through a systematic mapping. According to Kitchenham *et al.* (2007), a systematic mapping provides an overview of a research area and helps identify gaps that can be addressed in future research. As a result of the study, we obtained a consolidated set of measures, the processes related to them and the goals that led to the use of those measures.

In this paper we present the study and its main results. It is organized as follows: Section 2 talks briefly about SPC, Section 3 concerns the research protocol used in the study, Section 4 presents the main obtained results, Section 5 regards discussions about the results, and Section 6 addresses final considerations.

2. Statistical Process Control

Software Measurement (SM) is an essential process for organizations to achieve maturity in software development. Shortly, SM consists on defining measures to provide usefull information for decision making and goals monitoring, collecting data for the identified measures and analyzing them to obtain necessary information [ISO/IEC 2007].

Depending on the organization's maturity level, SM is performed in different ways. At initial levels, traditional measurement consists basically in collecting data from projects and comparing them with their corresponding planned values. At high maturity levels, besides traditional measurement, it is necessary to carry out SPC in order to know the processes behavior, determine their performance in previous executions, and predict their performance in current and future projects, verifying if they are able to achieve the established goals. SPC uses a set of statistical techniques to determine if a process is under control, considering the statistical point of view. A process is under control if its behavior is stable, i.e., if 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 performance measures defined for this process [Barcellos *et al.* 2013].

A process under control is a stable process and, as such, has repeatable behavior. So, it is possible to predict its performance in future executions and, thus, to prepare achievable plans and to improve the process continuously. On the other hand, a process that varies beyond the expected limits is an unstable process and the causes of these variations (said special causes) must be investigated and addressed by improvement actions, in order to stabilize the process. Once the processes are stable, their levels of variation can be established and sustained, being possible to predict their results. Thus, it is also possible to identify the processes that are capable of achieving the established goals and the processes that are failing in meeting the goals. In this case, actions to change the process in order to make it capable should be carried out. Stabilizing their critical processes is a characteristic of high maturity organizations or organizations that are looking forward to achieve the highest maturity levels [Florac and Carleton 1999].

3. Research Protocol

The systematic mapping was performed following the approach defined in [Kitchenham and Charters 2007], which includes: *planning*, when the research protocol is defined; *conducting*, when the protocol is executed and data are extracted, analyzed and recorded; and *reporting*, when the results are recorded and made available to potential interested parties. In this section we present the main parts of the research protocol.

The **goal** of the study is to identify measures that have been used in SPC initiatives for software processes or suggested for it. For achieving the study goal, we defined seven **research questions** (RQ):

RQ1. When and in which type of vehicle have the publications been published?

RQ2. What measures have been applied in SPC initiatives (or suggested for it)?

RQ3. What measurement goals have lead to the use/suggestion of the measures?

RQ4. What processes are the measures related to?

RQ5. Which are the measures categories?

RQ6. Have the measures been used in SPI initiatives?

RQ7. Have the measures been used/suggested in the context of SPI standards/models? Which ones?

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 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")*. For establishing 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 relevance of the selected publications, even thought it had selected many publications that had to be 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 (*www.springerlink.com*), Engineering Village (*www.engineeringvillage.com*), Web of Science (*webofscience.com*), Science Direct *(www.sciencedirect.com),* and Scopus (*www.scopus.com*).

Publications selection was performed in five steps: (S1) *Preliminary selection and cataloging*, when the search string was applied in the search mechanisms of the digital libraries (we limited the search scope to the Computer Science area). (S2) *Duplicates 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; and (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 was 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 suitable for SPC in software process or presents applications of SPC in which measures used are cited; (EC4) the publication is a copy or an older version of an already considered publication; (EC5) the publication is not written in English; and (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 were analyzed looking for the ones able to present evidence to the study. Thus, in this step, references of the publications selected in S4 were investigated by applying the first and second filters.

4. Results

The systematic mapping considered studies published until 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, achieving 318 publications. After S3, only 84 studies were selected (a reduction of approximately 73,58%). After S4, we achieved 39 studies. Applying snowballing (S5), 11 publications were added, reaching a total of 50 publications. Following, for each research question, we present a data synthesis of the main obtained results.

Publications vehicle and year (RQ1): Publications year range from 1989 to 2014, with some gaps, as shown in Figure 1. Regarding publication vehicle, 26 publications (52%) were published in scientific events and 24 (48%) in journals. Usually journals require more mature works. The homogeneous distribution of the studies in scientific events and journals can be understood as a sign that the topic has been explored, discussed and matured.

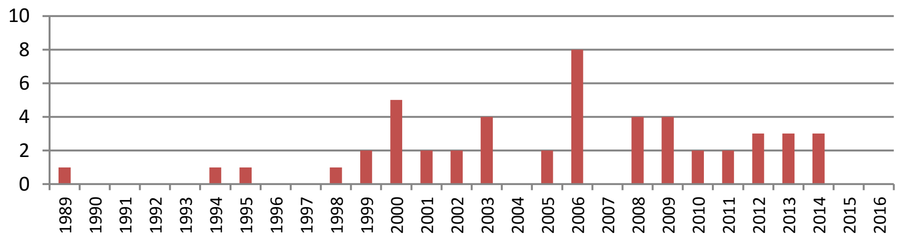

Figure 1 – Publications year.

Measures for SPC (RQ2), Supported Measurement Goals (RQ3) and Related Processes (RQ4): in total, 108 measures were identified, 15 related processes and 49 supported goals. To obtain these results, we followed this procedure: first, we extracted measures, processes and goals from publications exactely how they were recorded (e.g., the measure *delivered error rate,* which refers to the number of errors per thousand lines of code delivered, is cited in [Card and Berg 1989]). Then, we adjusted the names aiming to make the meaning as clear as possible (e.g., we named the cited mesure *scaped error rate*). Finaly, we idenfied the findings with the same meaning and used the same name to represent them (e.g., the cited measure and all others with the same meaning were referred to as *scaped defect density*). Table 1 presents the set of identified measures, the processes and goals related to them. Measures preceded by * were used in initiatives involving standards/maturity models. Measures preceded by ° 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 that measure related to that process/goal. Due to space limitation, Table 1 does not show the publications from which data were extracted. This information can be found in *https://www.dropbox.com/s/bwpq9ck7vn7hzyq/Mapping_Results.pdf?dl=0*. Also due to space restrition, in Table 1 measurement goals are referred by numbers, considering the list of measurement goals provided bellow. When the measurement goal is 0, it means that it was not possible to identify the goal in the publications that cite the measure.

Measurement goals: 1: Assess and monitor the maintenance process; 2: Minimize rework; 3: Control variation in the coding and code review processes; 4: Deliver a near defect-free system; 5: Evaluate defect-fixing efficiency; 6: Evaluate testing efficiency; 7: Improve productivity; 8: Estimate and control defects, effort and schedule of testing process; 9: Evaluate coding efficiency; 10: Evaluate design efficiency; 11: Evaluate defect-detection effectiveness; 12: Improve defect detection; 13: Improve estimation and planning; 14: Improve product quality; 15: Improve software reliability; 16: Improve software process effectiveness; 17: Increase customer satisfaction; 18: Evaluate peer review effectiveness; 19: Evaluate process quality; 20: Monitor process performance; 21: Monitor response time in order not to delay software updates and changes; 22: Improve review process; 23: Reduce costs spent on poorquality performance; 24: Reduce defects in the products; 25: Reduce operating costs; 26: Reduce the amount of effort spent on poorquality performance; 27: Reduce changing in requirements; 28: Reduce the number of escaped defects; 29: Monitor projects cost and schedule; 30: Improve schedule performance; 31: Manage the distribution of defect injection in different kind of activities; 32: Manage system-testing and fixing activities; 33: Manage the effectiveness of defect removal activities; 34: Reduce injected defect; 35: Understand fixing process performance; 36: Understand review process performance; 37: Understand project management process performance; 38: Understand recruitment process performance; 39: Understand software processes performance; 40: Understand test process performance; 41: Understand the effect of reviews as verification activities in test; 42: Understand the effect of test design in test development; 43: Understand the relationship between productivity and quality assurance activities during test development; 44: Understand the testing process performance; 45: Understand and predict product and development processes quality; 46: Verify changes in the test process; 47: Verify if the development meets the quality goals; 48: Evaluate inspection effectiveness; 49: Win the market competition.

Table 1. Measures for SPC, process and goals related to them.

Measures	Processes	Goals
*Actual procurement time *(start date for joining of new project member - start date of recruitment process + 1)*	Recruitment	38
*Procurement time variance *(actual procurement time - planned procurement time)*		
*Requirements change rate *(changed requirements /total requirements)*	Requirements Management	27
°maintenance time	Maintenance	0
°Amount of time spent responding to trouble reports		21
°Cost of poor quality *(cost of correcting internal failure + cost of correcting external failure)*	Coding	23
°Cost of quality *(cost of appraisal + cost of defect prevention + cost of correcting internal failure + cost of correcting external failure)*		
*Defect Injection Distribution *(defects injected in requirements (or design, coding and testing) / all defects removed in system testing * 100%)*	Coding, Design, Requirements Development, Testing	32
*Defect Removal Effectiveness *(number of removed defects in requirements (or design, coding and testing) /number of detected defects)*		33

Table 1. Measures for SPC, process and goals related to them (cont).

Meaures	Processes	Goals
*Cost performance index *(budget cost for work performed / actual cost for work performed)*	Project Management	29
*Schedule performance index *(budget cost for work performed/ budget cost for work scheduled)*	Project Management	29
°Cost Performance Index Acumulated *(budget cost for work scheduled/Total effort used by all actions)*	Project Management	29
*Task effort estimation accuracy *(task estimated effort / task actual effort)*	Project Management	37
*Task effort variance *(task estimated effort - task actual effort)*	Project Management	37
°Code estimation accuracy *(actual code size/estimated code size)*	Project Management	49
°Cost estimation accuracy *(actual cost / estimated cost)*	Project Management	49
°File estimation accuracy *(actual number of files/estimated number of files)*	Project Management	49
*Effort estimation accuracy *(estimated effort / actual effort)*	Project Management	7; 13; 14; 37; 49
*Duration estimation accuracy *(actual duration / estimated duration)*	Project Management	7; 13; 14; 30; 37; 49
*Early defect detection rate *(number of detected defects in reviews/number of all detected defects)*	Review	18
*Review Efficiency *(number of detected defects/ effort spent at the review)*	Review	18
°Action item closure date variance *(actual closure dates - planned closure dates)*	Review	24
*Review effort per action item *(total review effort/number of action items detected in peer review)*	Review	0
*Review effectiveness *(number of defects detected in peer reviews/total number of defects)*	Review	18
*Total review effort *(test development peer review effort + test development internal review effort)*	Review	41
*Non-conformance average review open duration *(sum of review open durations/number of non-coformances)*	Review	22; 36
*Non-conformance detection efficiency *(number of non-conformances/detection effort)*	Review	22; 36
*Non-conformance resolution efficiency *(number of solved non-conformances /resolution effort)*	Review	22; 36
*Review open duration *(closure date – opening date)*	Review	22; 36
Review preparation rate *(size of the product to be reviewed/time spent to prepare the review)*	Review	22
Review rate *(size of the reviewed product/ time spent at review)*	Review	22
Average time spent prepararing for review(sum of time spent by each person at review preparation/ number of reviewers)*	Review	3
*Effective preparation speed *(product size/average time spent prepararing for review + average time spent prepararing for rereview)*	Review	3
*Effective review speed *(product size/spent time at all reviews to the product)*	Review	3
*Preparation speed *(product size/average time spent prepararing for review)*	Review	3
*Review performance *(review effort/size of reviewed product)*	Review	39
*Number of action items detected in peer review	Review	41; 43

Table 1. Measures for SPC, process and goals related to them (cont).

Meaures	Processes	Goals
*Rework efficiency *(number of fixed defects / defect fixing effort)*	Fixing	5
*Anomaly resolution effort ratio *(effort spent at anomaly resolution/ number of solved anomaly)*		35
*Defect aging *(resolution date - creation date + 1)*		
*Percentage of Fixing Effort *(effort spent at defect fixing activities/ total effort of project * 100%)*		32
°Design inspection effetiveness *(detected defects/scaped defects)*	Inspection	0
°Inspection productivity *(number of detected defects/spent effort)*		
°Inspection effectiveness *(number of detected defects/number of escaped defects)*		
*Inspection performance *(size of inspected product/inspection effort)*		0
°Escaped defects *(number of detected defects - number of removed defects + number of injected defects)*		45
°Problem arrival rate *(problems detected/product size)*		
°Defect removal rate *(number of removed defects/effort spent removing defects)*		12; 45
°Defects detection rate (number of detected defects/inspection)		0
Test effectiveness *(number of detected defects by test / number of all defects)*	Testing	0
*Test efficiency *(detected defects/ detected defects + scaped defects)*		6
*Defect-detecting effort		8
*Development effort		
*Number of defects injected in coding		
*Number of defects injected in design		
*Number of defects injected in requirements		
°Difference of mean time between failures		15
*Percentage of Detecting Effort in System Testing *(effort spent at defect detecting activities/ total effort of project * 100%)*		32
*Test anomaly density (*number of test anomalies found by verification and validation/number of tests reviewed by verification and validation)*		40
*Test verification and validation effectiveness *(number of test anomalies found by verification and validation/ number of test anomalies found by all sources)*		
*Ratio of test development internal review effort *(test development internal review effort/test development effort)*		42
*Test development effort *(test design effort + test procedure preparation effort)*		
*Test development internal review effort *(test design internal review effort + test procedure preparation internal review effort)*		
*Test development productivity *(number of test cases/test development effort)*		
*Ratio of test design internal review effort *(test design internal review effort / test design effort)*		
*Test design effort		
*Test design internal review effort		
*Test design productivity *(number of test cases / test design effort)*		
*Test anomaly density in development *(number of test anomalies found by verification and validation in development/number of tests reviewed by verification and validation in development)*		39
*Unit test speed *(size of the tested product/time spent at unit test)*		
*Unit test verification and validation effectiveness *(number of unit test anomalies found by verification and validation/number of unit test anomalies found by all sources)*		

Table 1. Measures for SPC, process and goals related to them (cont).

Measures	Processes	Goals
*Action item detection efficiency *(number of action items / test development peer review effort)*	Testing	43
*Action item resolution efficiency *(number of action items / action items resolution effort)*		
*Action items density *(number of action items detected in peer review/ number of test cases)*		
*Action items resolution effort		
*Ratio of test procedure development internal review effort *(test script development internal review effort / actual test script development effort)*		
*Test development peer review effort		
*Test procedure development productivity *(number of test cases/actual test script development effort)*		
*Escaped defect density *(defects detected after product release / product size)*		14; 28
*System test effectiveness *(number of detected defects by system test / number of all defects)*		44
*System test effort estimation accuracy *(system test estimated effort / system test actual effort)*		
*System test speed *(size of the tested product/time spent at system test)*		
*System test verification and validation effectiveness (*number of system test anomalies found by verification and validation / number of system test anomalies found by all sources)*		
*Test anomaly density in system test *(number of test anomalies found by verification and validation in system test/number of tests reviewed by verification and validation in system test)*		
*Unit test effectiveness *(number of detected defects by unit test / number of all defects)*		39; 46
*Ratio of test procedure preparation internal review effort *(test procedure preparation internal review effort/test procedure preparation effort)*		0
*Test procedure preparation effort		
*Test procedure preparation internal review effort		
*Test procedure preparation productivity *(number of test cases/test procedure preparation effort)*		
°Completed Task Problem Density *(number of completed tasks/ number of defects of all completed tasks)*	Software Development	14
°Expected Task Problem Density *(number of tasks expected to be completed/ number of defects of all tasks)*		
*Rework percentage *(rework effort/total effort *100)*		2; 39
*Effective defect density *(detected defects in all reviews to the product /product size)*	Coding, Review	3
*Defect-fixing effort	Fixing, Testing	8; 35
*Inspection preparation rate *(size of the product to be inspected/time spent to prepare the inspection)*	Inspection, Review	12; 18; 45
*Inspection rate (*inspected product size/ time spent at inspection)*		11; 12; 16; 18; 45
*Review speed *(product size/spent time at review)*	Coding, Review	3; 18

Table 1. Measures for SPC, process and goals related to them (cont.)

Measures	Processes	Goals
*Percentage of defects caused by faulty logic *(number of defects caused by faulty logic /total number of detected defects *100)*	Inspection, Testing	7; 17; 25
*Percentage of defects found in operation *(number of defects found in operation/total number of detected defects *100)*		
*Percentage of high severity defects identified in production *(number of high severity defects identified in production /total number of detected defects *100)*		
*Percentage of high severity defects identified in testing *(number of high severity defects identified in testing/total number of detected defects *100)*		
*Percentage of rejected defects *(number of rejected defects/total number of detected defects *100)*		
*Percentage of effort saved for process automation	Project Management, Quality Assurance, Risk Management, Testing	7
*Defect density *(number of detected defects/product size)*	Review, Coding, Inspection, Maintenance, Software Development, Testing	3; 4; 6; 8; 12; 14; 18; 19; 24; 39; 45; 47; 48; 49
*Defect injection rate (by phase) *(number of injected defects/number of removed detected defects)*	Coding, Design, Requirements Development, Testing, Inspection	9; 10; 31; 34; 45; 49
*Number of defects	Coding, Inspection, Maintenance, Review, Testing	1; 7; 8; 14; 16
*Effort	Customer release, Software Development, Testing	8; 26
*Productivity *(product size(or task duration)/effort)*	Maintenance, Software Development, Testing	1; 7; 8; 16; 20; 39

Measures Category (RQ5): for identifying measures categories, we used the ones suggested in PSM [McGarry *et al.* 2002], namely: Cost, Effort, Performance, Quality, Size, and Time. From the 108 measures identified, 53 (49,07%) are realted to Quality, 27 (25,00%) to Effort, 15 (13,89%) to Performance, 8 (7,41%) to Time, and 5 (4,63%) to Cost.

Measures Use in the context of Standards/Maturiry Models (RQ6 e RQ7): the majority of the identified measures were applied in practice (105 measures, 97,22%) and most of these (86 measures, 79,63%) was used in SPC initiatives carried out in the context of standards/maturity models. All of these measures were used in the context of CMMI.

Among them, the following measures were also used in ISO 9001 [ISO 2015] initiatives (corresponding to 10,18% of the identified measures): defect density, effort estimation accuracy, duration estimation accuracy, percentage of effort saved for process automation, review effectiveness, average time spent prepararing for review, effective defect density, effective preparation speed, effective review speed, preparation speed and review speed.

5. Discussion

This section provides some discussions about the results presented in the previous one.

Most of the identified measures are related to defects and, consequently, to processes that deal with them (*Testing*, *Review*, *Inspection*). Measures related to defects are often used in SPC for two main reasons: (i) processes addressing defects-related measures are directly related to software quality and, thus, are critical to organizations and natural candidates to be submmited to SPC, since critical processes are the ones indicated to be statistically controlled [Tarhan and Demirors 2008; SEI 2010; Barcellos *et al.* 2013]; (ii) these processes are performed many times along projects, favoring data collection and geting the amount of data required to apply a measure in SPC. *Defect density* was the most cited measure, being used in 33 publications (66%), such as [Florence 2001; Jacob and Pillai 2003; Weller and Card 2008; Vijaya and Arumugam 2010; Tarhan and Demirors 2012; Alhassan and Jawawi 2014; Vashisht 2014]. In some studies, this measure is used to quantify different types of defects (e.g., code defect density and file defect density [Zhu et al. 2009]). *Testing* was the most cited process, being object of SPC in 17 publications (34%) (e.g., [Jalote *et al.* 2000; Komuro 2006; Tarhan and Demirors 2011a; Fernandez-Corrales *et al.* 2013]), and *Inspection* was the second most cited, being submitted to SPC in 15 publications (30%) (e.g. [Hayes 1998; Weller 2000; Narayana and Swamy 2003; Zhang and Sheth 2006; Vijaya and Arumugam 2010]). The *Project Management* process was object of analysis in 8 publications ([Keeni 2000; Wang *et al.* 2006; Chang and Chu 2008; Tarhan and Demirors 2008, 2011b, 2012; Zhu *et al.* 2009]). It is also a process prone to be submitted to SPC, because it is usually a critical process to organizations (budget and schedule are addressed by it) and data can be frequently collected. Other processes, such as *Risk Management* and *Customer Release* were cited in only one publication.

Some publications refer to the *Development Process* as the process submitted to SPC and related to the identified measures. Usually, the development process as a whole (involving requirements, design and coding) is not indicated to be controlled by using SPC, since it is too large and SPC is indicated to small processes [Tarhan and Demirors 2008; Barcellos *et al.* 2013]. However, although publications cite development process, measures are, in fact, related to pieces of the process and, thus, can be suitable for SPC. For instance, the measures *effort* [Wang *et al.* 2006] and *productivity* [Card 1994; Baldassarre *et al.* 2005; Wang and Li 2005; Boffoli 2006; Wang, Qing *et al.* 2006; Chang and Chu 2008; Gou *et al.* 2009; Tarhan and Demirors 2012; Vashisht 2014] are obtained for each task, activity or phase, producing data usefull to describe the process/subprocess performance. Considering that small processes are more suitable for SPC, some measures are related to parts of processes. For instance, the measures *test procedure preparation effort* and *test procedure preparation productivity* [Tarhan and Demirors 2011a] are related to the *Testing* process, but more specifically to the *Testing Preparation* subprocess.

Concerning measures categories, *quality* measures are almost half of the found measures (49,07%). This is a consequence of the fact that most measures are related to defects, which is directly related to quality aspects. *Performance* measures are the second most cited, particularly the ones related to productivity, which describe processes behaviour by means of the spent effort and the work done. There is no measure related to size. Size measures are not indicated to be used in SPC because they are not able to describe processes performance. They are often used to compose other measures able to provide information about processes 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 impacted in it).

As for measurement goals, some publications present explicitly the goals that motivated SPC use and measures selection. Others, do not mention explicitly the measurement goals, but it is possible to get them from the text. However, some publications do not present the measurement goals and it is not possible to infer them from the text. SPC should be performed to support monitoring goals [Florac and Carleton 1999; SEI 2010; Barcellos *et al.* 2013]. In this sense, it is important to make explicit which measurement goals are to be monitored and which measures are to be used for that. Among the identified measurement goals, there are some very specific, such as *Understand the effect of reviews as verification activities in test* [Tarhan and Demirors 2011a]. In line with the most cited measures, most goals are related to quality aspects (e.g., *Reduce defects in the products* [Bertolino *et al.* 2014; Narayana and Swamy 2003; Selby 2009], *Improve product quality* [Mohapatra and Mohanty 2001; Schneidewind 2011; Wang *et al.* 2006], *Improve defect detection* [Weller and Card 2008]). There are several measurement goals involving understanding process performance (e.g., *Understand fixing process performance* [Tarhan and Demirors 2012], *Undersand project management process performance* [Tarhan and Demirors 2008, 2011b, 2012]). We noticed that in most of these cases SPC practices were starting and, as a consequence, the first result expected from SPC was to get to know the processes behavior so that it would be possible to improve it. Finally, we found some goals that seem to be closer to business goals than measurement goals (e.g., *Win the market competition* [Zhu *et al.* 2009]). However, since measurement goals can be understood as goals that lead to measurement actions [Barcellos *et al.* 2013] and the publication does not present more specific goals, we identified it as the measurement goal related to the selected measures.

With respect to the measures use, most of measures (97,22%) were applied in practical initiatives. Only the measures *test effectiveness*, *review preparation rate* and *review rate*, cited in [Card 1994; Jalote *et al.* 2000], were not applied in a real situation reported in the selected publications. We did not eliminate these measures because the authors argue 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 like CMMI, aiming at a broader process improvement in a SPI program. 79,63% of the identified measures were used in practical initiatives involving CMMI or ISO 9001. This shows that in the context of software processes, SPC has been used in the context of SPI programs guided by standards or maturity models, particularly CMMI.

6. Final Considerations

This paper presented the main results of a systematic mapping that investigated measures suitable for SPC, the related process and goals supported by them. A total of 318 publications were analyzed and 108 measures were identified from 50 publications.

Before performing the systematic mapping, we investigated the literature looking for secondary studies about measures suitable for SPC. We did not find any, and, then, we decided to perform the study reported in this paper. Although there is no systematic study investigating measures for SPC, there are some related works to ours. Monteiro and Oliveira (2011), for example, present a catalog of measures to process performance analysis. However, although they claim to have done a broad literature review, they did not follow a systematic approach. Besides, measures categories, measurement goals and information regarding the measures use were not investigated in their study. Rocha, Santos and Barcellos (2012), in turn, suggest a set of measures related to the MR-MPS-SW processes, including measures that can be applied to SPC. However, the set of measures is not focused on SPC use. Also, the suggested measures were not obtained from a literature investigation, are not related to measurement goals, their categories are not analyzed and information regarding their use is not provided.

The main result of this study is the set of measures suitable for SPC, their categories, related processes and measurement goals. As shown in Table 1, a measure can be related to more than one measurement goal. It means that the measure was used in different publications to support monitoring different goals. From Table 1 it would be possible to infer that a certain measure could also to be used to support other identified goals. However, it is important to point out that our goal in the study was to identify the literature evidences about measures for SPC initiatives, thus, in this paper we have been limited to present the literature findings. As an ongoing work, we have been analyzing the study results aiming to get new information from them (for instance, which goals are related to each other and how they relate, which processes (besides the ones identified in the study) could be measured by the identified measures, which measures could be used in a combined way to support measurement goals, and so on).

According to Kitchenham and Budgen (2011), a mapping study gives an idea of shortcomings in existing evidence, which becomes a basis for future studies. The study results showed us that SPC has been focused on defect-related measures and processes, even there being many other processes that can be explored and improved by SPC thecniques. Also, we noticed a lack of concern with correlate measures that are necessary to support root causes investigation when analyzing a process behavior. Besides, by reading the publications we noticed that although they present the used measures, there is lack of approaches to select the appropriate measures considering a certain context. We also noticed that, although measures are cited, their operational definitions are not presented. Even basic information regarding the measures (e.g., how often are data is collected) is not explicit in the publications. This can limit the measures reuse, since the reader can missunderstood the measures. Finally, it is important to point out that for using a measure in SPC, some criteria should be observed, such as its operational definition and collected data [Barcellos *et al.* 2013]. Thus, when selecting the measures to be used it is also necessary to assure that they meet the required criteria.

Having observed that, we have been developing a pattern-based approach to support measures selection to SPC initiatives. We have been studying relations between measures and patterns of measures use according to goals to be achieved. The set of measures obtained in this study and a set of measures used by high maturity Brazilian organizations will serve as a basis to form a set of measures suitable for SPC (with operational definitions) that could be selected by considering patterns emerged from the relations between measures, processes and goals.

Acknowledgment

This research is funded by the Brazilian Research Funding Agency CNPq (Processes 485368/2013-7 and 461777/2014-2).

References

Alhassan, M. A. and Jawawi, D. N. A. (2014). Sequential strategy for software process measurement that uses Statistical Process Control. *8th Malaysian Software Engineering Conference (MySEC)*, p. 37–42.

Baldassarre, M. T., Boffoli, N., Caivano, D. and Visaggio, G. (2005). Improving dynamic calibration through statistical process control. *21st IEEE International Conference on Software Maintenance (ICSM'05)*, p. 273–282.

Barcellos, M. P., Falbo, R. de A. and Rocha, A. R. (2013). A strategy for preparing software organizations for statistical process control. *Journal of the Brazilian Computer Society*, v. 19, n. 4, p. 445–473.

Bertolino, A., Marchetti, E., Mirandola, R., Lombardi, G. and Peciola, E. (2014). Experience of applying statistical control techniques to the function test phase of a large telecommunications system. *IEEE Software*, v. 149, n. 4, p. 349–357.

Boffoli, N. (2006). Non-intrusive monitoring of software quality. *Proceedings of the European Conference on Software Maintenance and Reengineering,* p. 319–322.

Card, D. (1994). Statistical Process Control for Software? *IEEE Software*, v. 11, n. 3, p. 95–97.

Card, D. N. and Berg, R. A. (1989). An industrial engineering approach to software development. *Journal of Systems and Software*, v. 10, n. 3, p. 159–168.

Card, D. N., Domzalski, K. and Davies, G. (2008). Making Statistics Part of Decision Making in an Engineering Organization. *IEEE Software*, v. 25, n. 3, p. 37 – 47.

Chang, C.-P. and Chu, C.-P. (2008). Improvement of causal analysis using multivariate statistical process control. *Software Quality Journal*, v. 16, n. 3, p. 377–409.

Fernandez-Corrales, C., Jenkins, M. and Villegas, J. (2013). Application of Statistical Process Control to Software Defect Metrics: An Industry Experience Report. *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 323–331.

Florac, W. A. and Carleton, A. D. (1999). *Measuring the Software Process: Statistical Process Control for Software Process Improvement*. Pearson Education.

Florence, A. (2001). CMM Level 4 Quantitative Analysis and Defect Prevention. *Crosstalk*, Feb, p. 20–21.

Gou, L., Wang, Q., Yuan, J., et al. (2009). Quantitative defects management in iterative development with BiDefect. *Software Process: Improvement and Practice - Addressing Management Issues*, v. 14, n. 4, p. 227–241.

Hayes, W. (1998). Using a Personal Software Process(SM) to improve performance. *Proceedings Fifth International Software Metrics Symposium*, p. 61–71.

ISO 9001 (2015). Quality management systems — Requirements.

ISO/IEC (2007). *ISO/IEC15939—Systems and Software Engineering—Measurement Process*.

Jacob, A. L. and Pillai, S. K. (2003). Statistical process control to improve coding and code review. *IEEE Software*, v. 20, n. 3, p. 50–55.

Jalote, P., Dinesh, K., Raghavan, S., Bhashyam, M. R. and Ramakrishnan, M. (2000). Quantitative Quality Management through Defect Prediction and Statistical Process Control. *Proceedings of Second World Quality Congress for Software*,

Keeni, G. (2000). The Evolution of Quality Processes at Tata Consultancy Services. *IEEE Software*, v. 17, n. 4, p. 79–88.

Kitchenham, B. and Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. Staffordshire, UK.

Komuro, M. (2006). Experiences of applying SPC techniques to software development processes. *Proceeding of the 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*. Boston, MA, USA.

Mohapatra, S. and Mohanty, B. (2001). Defect Prevention through Defect Prediction : A Case Study at Infosys. *Proceedings. IEEE International Conference on Software Maintenance*, p. 260 – 272.

Narayana, V. and Swamy, R. (2003). Experiences in the inspection process characterization techniques. *Proceedings - International Conference on Quality Software*, n. Jan, p. 388–395.

Schneidewind, N. (2011). What can software engineers learn from manufacturing to improve software process and product? *Journal of Intelligent Manufacturing*, v. 22, n. 4, p. 597–606.

SEI (2010). CMMI for Development, Version 1.3. *Carnegie Mellon University*,

Selby, R. W. (2009). Statistical Process Control for System Development Using Six Sigma Techniques. *AIAA SPACE Conference & Exposition*, Sept.

Softex (2016). *MPS . BR - Melhoria de Processo do Software Brasileiro - Guia Geral MPS de Software*.

Takara, A., Bettin, A. X. and Toledo, C. M. T. (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)*, p. 91 – 99.

Tarhan, A. and Demirors, O. (2008). Assessment of Software Process and Metrics to Support Quantitative Understanding. *IWSM-Mensura*, p. 102–113.

Tarhan, A. and Demirors, O. (2011a). Investigating the effect of variations in the test development process: A case from a safety-critical system. *Software Quality Journal*, v. 19, n. 4, p. 615–642.

Tarhan, A. and Demirors, O. (2011b). Assessment of software process and metrics to support quantitative understanding: Experience from an undefined task management process. *Communications in Computer and Information Science (CCIS0)*, v. 155, p. 108–120.

Tarhan, A. and Demirors, O. (2012). Apply quantitative management now. *IEEE Software*, v. 29, n. 3, p. 77–85.

Vashisht, V. (2014). Enhancing Software Process Management through Control Charts. *Journal of Software Engineering and Applications*, Feb, p. 87–93.

Vijaya, G. and Arumugam, S. (2010). Monitoring the stability of the processes in defined level software companies using control charts with three sigma limits. *WSEAS Transactions on Information Science and Applications*, v. 7, n. 10, p. 1230–1239.

Wang, Q., Gou, L., Jiang, N., et al. (2006). Estimating fixing effort and schedule based on defect injection distribution. *Software Process Improvement and Practice*, v. 11, p. 361–371.

Wang, Q., Jiang, N., Gou, L., et al. (2006). BSR: A statistic-based approach for establishing and refining software process performance baseline. *Proceedings of the 28th international conference on Software engineering*, p. 585–594.

Wang, Q., Jiang, N., Gou, L., Che, M. and Zhang, R. (2006). Practical experiences of cost/schedule measure through earned value management and statistical process control. *Software Process Change*, p. 348–354.

Wang, Q. and Li, M. (2005). Measuring and improving software process in China. *International Symposium on Empirical Software Engineering*, p. 177–186.

Weller, E. F. (2000). Practical applications of statistical process control. *IEEE Software*, v. 17, n. 3, p. 48–55.

Weller, E. F. and Card, D. (2008). Applying SPC to Software Development: Where and Why. *IEEE Software*, v. 25, p. 48–50.

Zhang, Y. and Sheth, D. (2006). Mining software repositories for model-driven development. *IEEE Software*, v. 23, n. 1.

Zhu, M., Liu, W., Hu, W. and Fang, Z. (2009). Target Based Software Process Evaluation Model and Application. *Second International Conference on Information and Computing Science (ICIC)*, v. 1, p. 107–110.