Systematic Review of Statistical Process Control: An Experience Report

Maria Teresa Baldassarre[§], Danilo Caivano[§], Barbara Kitchenham* Giuseppe Visaggio[§]
[§] Department of Informatics, University of Bari – RCOST Bari:
{baldassarre, caivano, visaggio}@di.uniba.it,

*Keele University, UK
ap_kitchenham@onetel.com

Background: A systematic review is a rigorous method for assessing and aggregating research results. Unlike an ordinary literature review consisting of an annotated bibliography, a systematic review analyzes existing literature with reference to specific research questions on a topic of interest.
Objective: Statistical Process Control (SPC) is a well established technique in manufacturing contexts that only recently has been used in software production. Software production is unlike manufacturing because it is human rather than machine-intensive, and results in the production of single one-off items. It is therefore pertinent to assess how successful SPC is in the context of software production. These considerations have therefore motivated us to define and carry out a systematic review to assess whether SPC is being used effectively and correctly by software practitioners.
Method: A protocol has been defined, according to the systematic literature review process, it was revised and refined by the authors. At the current time, the review is being carried out.
Results: We report our considerations and preliminary results in defining and carrying out a systematic review on SPC, and how graduate students have been included in the review process of a first set of the papers.
Conclusions: Our first results and impressions are positive. Also, involving graduate students has been a successful experience.

Systematic review, statistical process control, empirical software engineering

1. INTRODUCTION

A systematic review is a formal approach for reviewing research literature. As reviews are often limited to annotated bibliographies, a *systematic review* means giving appropriate breadth and depth, rigour and consistency, let alone effective analysis and synthesis of the literature. Furthermore, it can be considered as much more effort prone than an ordinary literature survey. The latter being formally defined as “the selection of available documents (both published and unpublished) on the topic, which contain information, ideas, data and evidence written from a particular standpoint to fulfil certain aims or express certain views on the nature of the topic and how it is to be investigated, and the effective evaluation of these documents in relation to the research being proposed” [18]. As so, it has less scientific value than a systematic review, formally defined as a “means of evaluating and interpreting all available research relevant to a particular research question or topic area or phenomenon of interest” [26].

Guidelines on systematic review have been defined and are quite stable in contexts such as medicine, social sciences, education and information sciences and used for analysing and synthesizing existing empirical results on a certain topic. Indeed, there are many existing guidelines in this field that include the Cochrane Reviewer’s Handbook [8], Guidelines of the Australian National Health and medical Research Council [2, 3]; CRD Guidelines for those Carrying Out or Commissioning Reviews [25].

Adaptations of these guidelines to software engineering have been made by Kitchenham in [26]. Also, applications of the procedure for performing a systematic review are becoming more and more common to the software engineering context in the past few years [4, 16, 17] and many studies have been carried out on various topics of interest that range from cost-estimation [24], within and cross company estimation models [27] to software process improvement [28], to statistical power [9].

In this scenario, the authors have defined and begun a systematic review of Statistical Process Control (SPC). This paper decribes details of the search strategy, the review process carried out and how graduate students have been involved in conducting the review.

The rest of the paper is organized as follows: the next section gives some preliminary details on SPC, so the reader can gain familiarity with the application domain; section 3 describes our review process, and the preliminary results achieved for the completed phases. Section 4 describes how graduate students are being implied as support to the data extraction step. The study is ongoing, so at the current time, we have drawn only preliminary conclusions.

2. STATISTICAL PROCESS CONTROL IN PILLS

Software process is human intensive and dominated by cognitive activities. Each input into and the output from a software process are different for each process execution. The predominant human factor implies differences in process performances and thus multiple outputs. The phenomena known as “Process Diversity” [19],

implies difficulty in predicting, monitoring and improving a software process. Nevertheless, Software Process Improvement (SPI) is strongly reccomended. To face these problems the software engineering community stresses the use of time series analyses that allow monitoring process performances in order to decide when to improve and verify the effectiveness of the improvements made on software processes. There is a well-established technique for time series analysis that has shown to be effective in manufacturing but not yet in software process contexts. This technique is known as Statistical Process Control (SPC) [33, 34]. It was originally developed by Shewhart in the 1920s and then used in many other contexts. It uses several "control charts" together with their indicators to: establish operational limits for acceptable process variation; monitor and evaluate process performances evolution in time. Process performance variations are mainly due to: common cause variations (the result of normal interactions of people, machines, environment, techniques used and so on); assignable cause variations (arise from events that are not part of the process and make it unstable).

SPC is a statistical based approach that determines whether a process is stable or not by discriminating between common cause variation and assignable cause variation. A process is said to be "stable" or "under control", if it is affected by common causes only. Each control chart evaluates process performance by comparing it with a measure of its central tendency, an upper and lower limit of admissible performance variations. The interest of using SPC in software is highlighted by many contributions in literature: applications in inspections and review [10, 13, 14, 38], testing [5, 6, 23], maintenance [37, 39], personal software process [30], and other topics [7, 12]. They present experiences of outcomes in using SPC in the context of software.

Software Process	Manufacturing Process
Human intensive, cognitive activity	Machine intensive
Input and Output are different for each process execution	Input and Output are always the same for each process execution
High variation in process performances due to human factors	Low variation in process performance
Risks always present in all phases	Most Risks concentrated in design phase than in production
Product conformance to requirement specifications is difficult to obtain	Product conformance to requirement specifications is provable

FIGURE 1: software process vs manufacturing process [29]

Software processes differ substantially from manufacturing ones [29]. The differences are summarized in Figure 1. A cause is the predominance of cognitive activities that encourage multiple outputs and the possibility of alternative ways for doing the same thing. Another cause is the influence of the person's performances on process. The input into and the output from the software process are different for each instance and consequently, the risk of process instability is constantly present.

Furthermore, each innovation determines a process destabilization and less predictability. This side effect influences software process until the innovation is fully integrated within it and thus becomes part of organization practices. This is often called "maturity effect" [41]. In manufacturing process this effect is not so strong. In fact, when a new machine is introduced within a product line it will most likely work well immediately.

Another important issue that differs between the two types of processes is the ease of verifying the adherence of an end product to its specifications: verifying that the dimensions of a piston meet the customer specification is simpler than verifying that software works correctly, that it is maintainable or robust, as the customer wants.

The large number of software process attributes and variables (people, tools, skills, experience, application domain, development language and technique etc.) suggests that, in order to address process stability, more indicators in software than in manufacturing processes are needed.

In SPC, process performances are tracked overtime on a control chart, control limits are determined and if one or more of the values fall outside these limits, an assignable cause is assumed to be present (i.e. the process is unstable). A control chart usually adopts an indicator of the process performances central tendency (CL) and upper and lower control limits (UCLs and LCLs) to discriminate between assignable and common cause variations. In software processes, since data are scare and measurements often occur only as individual values, XmR are the most commonly used control charts i.e. individual and moving range charts (Figure 2) [15, 31, 42, 38].

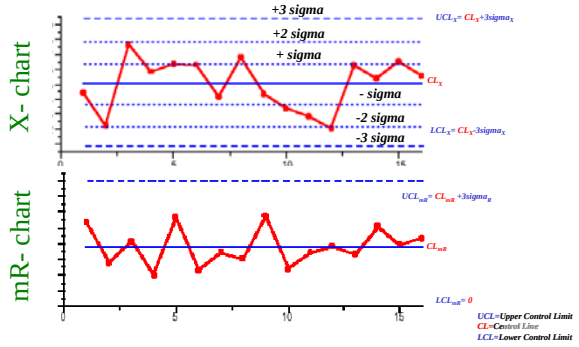

FIGURE 2: Example of Individual and moving ranges charts (XmR charts)

In the X chart each point represents a single value of the measurable characteristic under observation. The Central Line (CLx) expresses the process central tendency and is calculated as the average of the available values. The control limits are set at 3sigmaX around the mean, i.e. 3sigmaX around the CLx, where sigmaX is the estimated standard deviation of the observed sample of values. In the mR chart each point represents a moving range. The Central Line (CLmR), is computed as the average of the moving ranges. A moving range is the absolute difference between a successive pair of observations. The control limits are set at: 3sigmamR upon the CLmR for what concerns the UCLmR and to 0 for LCLmR. SigmamR is the estimated standard deviation of the moving ranges sample. Sigma is calculated by using a set of factors tabulated by statisticians.

Considerations and discussions on six sigma go beyond the scope of this paper. This section merely aims to provide the reader with rudiments of SPC.

3. SPC REVIEW PROTOCOL

Since the introduction of the Capability Maturity Model (CMM) in the early 90's, researchers and practitioners interested in achieving level 4 or above have become interested in applying principles of Statistical Process Control (SPC) to software production. We propose to undertake a systematic review in order to assess the evidence on how SPC is being used in software production and whether it is being applied effectively by practitioners.

This motivation for the review is twofold: SPC was originally developed in manufacturing industries to monitor the quality of objects produced by a manufacturing process and to take corrective action if the manufacturing process began to go out of control (i.e. the objects produced by the process no longer conformed to quality requirements). Software production is unlike manufacturing because it is human rather than machine-intensive, and results in the production of single one-off items. In addition, most software measurement data is not normally distributed.

It is therefore pertinent to assess how successful SPC is in the context of software production. The quality staff of companies using SPC and certified to be at CMM Level 5, are often concerned that SPC does not appear to be providing any major benefits either to the company or to the software project managers. In particular, managers complain that information is not precise enough to help them in their production activities.

This seems to suggest that either SPC is not of value in software production, or it is being incorrectly used, perhaps because the specific characteristics of software objects are not being properly addressed. Moreover, it suggests that it is important to assess whether SPC is being used effectively by practitioners.

This aspect has motivated our systematic literature review. Of course, any type of review may suffer from a biased sample of existing literature that most likely reports only successful and positive experiences on research topics, giving an incorrect idea of the situation. Nevertheless, it is a risk that we must take into account, as well as keep in mind that "negative results" correspond to "not published results" or "rejected for publication". We have addressed this issue by not limiting the literature search to journals and proceedings, but also considering grey literature, contacting authors, research groups and experts.

In defining the protocol and carrying out the review process, represented in Figure 3, we followed the guidelines in [26]. The process is made up of three main phases: planning the review, conducting the review, reporting the review.

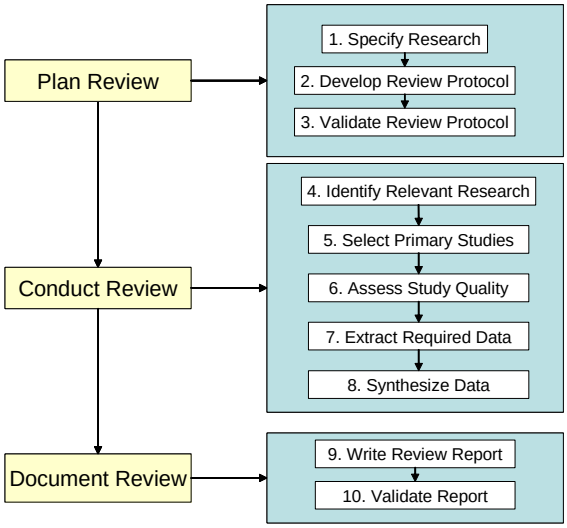

FIGURE 3: Systematic review process phases (figure adapted from [4])

A brief description of each phase is now provided, together with details on what point our research is at.

3.1. Planning the Review

This first phase of planning involves specifying the research, developing the review protocol and validating the protocol.

Specifying the research

The importance and motivations of the research have been illustrated at the beginning of this section. Moreover, our intention was to investigate sources on SPC in literature and provide evidence that it supports software production.

Developing the review protocol

The review protocol is what formally specifies the steps and procedures used for carrying out the systematic review. It is important for the protocol to be defined before starting the review in order to avoid that results are in any way influenced by researcher expectations and desiderata.

Given the motivation of the study, we proposed to investigate the following research questions:

RQ1: What software activities and control methods are being proposed for control by SPC?

RQ2: What adjustments are suggested for the software characteristics being addressed?

RQ3: What are the claimed benefits and identified limitations of software SPC?

RQ4: What evidence is there that the claimed benefits are being delivered?

RQ5: Is SPC being used correctly in software production?

Search strings were based on an analysis of the questions. In defining a search string it is important to keep in mind that: major terms are identified from the topic area, intervention and outcomes; search strings should include synonyms, related terms and alternative spelling for major terms; boolean “or” is used to incorporate alternative spellings and synonyms; boolean “and” is used to link major terms.

In this sense the following details have been used for constructing search strings:

Population: Software development and maintenance projects and tasks

Intervention: Statistical process control

Outcomes: Reported benefits , reported limitations, task type, software attribute controlled (e.g. productivity, defect rate)

Synonyms and related terms for statistical process control and outcomes are: SPC, process control, evaluation, assessment, analysis, error, error rate, problems.

Given these terms, we defined 4 search strings:

SearchString1: Software + (SPC or “Statistical Process Control” or “Six Sigma”)

SearchString2: Software + “productivity” + (“assessment” or “estimation” or “control” or “analysis”)

SearchString3: Software + (“defect” or “error” or “defect rate” or “error rate”) + (“assessment” or “estimation” or “control” or “analysis”)

SearchString4: SPC or “Statistical Process Control” + (“limitations” or “problems” or “shortcomings”)

The search process is divided into two parts. First primary sources were identified. SPC dates back to the 20’s as technique used in manufacturing contexts. Since then, a lot of literature has been produced. Nevertheless, its application to software is quite recent. Besides this review, over the years, authors of this paper have carried out much of their research on SPC in software. As so, to their knowledge the first paper on the application of SPC to software is Gardiner, J. S. and Montgomery, D. C., "Using Statistical Control of Charts for Software Quality Control," Quality and Reliability Engineering International, vol. 3 pp. 15-20, 1987. So we considered this paper the baseline and therefore we did not search for any articles published before 1987. The sources to search included databases, specific journals and conference proceedings with respect to publications ranging between 1987-2005. Next, secondary sources were searched. The secondary search phase consisted in checking the primary sources, identified in the initial search phase, for other relevant publications; and in contacting researchers who authored primary sources who we believed could be working on the topic, to enquire whether they had other unpublished papers or technical reports (i.e. grey literature).

Specific tables have been defined for documenting all the outcomes of the search process. References were planned to be stored in an excel spreadsheet with a specific format (author’s surname + year of publication + letter in case of multiple publications for an author in a same year + (primary reference | secondary reference)).

Selection criteria and procedures were also defined. They determine the criteria for including or excluding sources from the systematic review. In our study we decided that papers included in this review should be primarily concerned with the *application of SPC to software*. In particular, we included: theoretical papers that discussed SPC in the context of software; empirical papers that include analysis of real software data; papers with a theoretical and empirical content. We excluded: papers that discussed methods for controlling software development using metrics that do not explicitly discuss SPC; papers that do not address the application of SPC to software production; papers/reports for which only an abstract or a powerpoint slideshow are available.

An initial selection of primary sources was based on a review of title and abstract. It was intended to exclude primary sources that appeared completely irrelevant. For the primary searches two researchers were assigned to review the search list and keep a record of the selected papers prior to coming to an agreement. Selected papers were to be reviewed against the inclusion/exclusion criteria using the same process used for the abstracts. Reasons for inclusion/exclusion were recorded on the excel sheet.

Data extraction tables and aggregation tables have also been defined for accurately collecting and recording the information of all reviewed papers.

We have introduced quality assessment in order to provide more detailed inclusion/exclusion criteria and attribute importance to the papers. In particular, this was done in two parts: first we used a checklist of questions divided into two different sections, according to the nature of the paper (theoretical or empirical). In case of a paper with theoretical and empirical elements, both types of quality assets would be used to estimate the quality of the

work. These questions have been integrated in the data extraction forms, and evaluated on a [0, 1] range scale. Second, we defined a "quality assessment form" to extract information needed to address RQ5, i.e. questions that investigated whether SPC is correctly used in software production.

Validating the protocol

Two researchers defined the protocol, while other two validated it. Validation was carried out on behalf of a researcher of the research unit and of a researcher external to the research group, Barbara Kitchenham, and who had contributed to a first version of the protocol.

At the current state the planning phase is complete, and all the criteria to follow for conducting the review have been defined, reviewed and accepted on behalf of the researchers.

3.2. Conducting the Review

Once the protocol is finalized and validated, the next step is to conduct the review, i.e. apply all the steps as they have been formally defined. This points out the importance of the first phase in making the process replicable and adoptable by other researchers that may differ from those having defined the protocol itself. Documentation of the steps in this phase is also crucial to keep track of results.

Identify Relevant Research

As specified in the protocol, we defined 4 search strings. They were used to identify publications in 8 databases. The results of the search were stored in excel documents according to the structure in table 1.

ID	Authors	Title	Journal name	Vol	No	Pages	Year	Copyright	DOI	URL	Abstract

TABLE 1: Format for reporting search results

The search process was carried out by a researcher and a PhD student.

Select Primary Studies

Once the search had been carried out, the next step was to select the primary studies according to the inclusion/exclusion criteria. Given the high number of publications that the search terms had selected, two researchers independently read the title, keywords and abstract of the papers. If this wasn't enough to decide whether to include/exclude a paper, then conclusions were also read. Both researchers kept track of their own results and, once finished the selection, they compared their results with the ones of the other researcher. In about 90% of the cases, both researchers came to the same conclusions. In case of discordant results, the researchers discussed their opinions, in 4 cases they read the paper together and came to a final agreement.

Table 2 represents the results of the selection process on the databases.

	IEEE Xplore [20]		INSPEC [21]		Emeroteca Virtuale [11]		Springer Link [35]		Science Direct [32]		Web of Science [36]		ACM Digital Library [1]		Wiley Interscience [40]	
	Total	Incl.	Total	Incl.	Total	Incl.	Total	Incl.	Total	Incl.	Total	Incl.	Total	Incl.	Total	Incl.
Search string1	586	0	2135	2	6192	2	263	3	256	4	2805	5	NA	9*	NA	11*
Search string2	2493	3	5467	4	4883	0	0	0	106	3	2215	2	NA	NA	NA	NA
Search string3	512	5	2370	5	825	1	79	0	198	7	492	3	NA	NA	NA	NA
Search string4	120	12	1232	19	4718	1	51	5	0	0	205	9	NA	NA	NA	NA
Selected total	20		30		4		8		14		19		9		11	

TABLE 2: Results of inclusion/exclusion criteria of primary studies

* results refer to all the search strings and not only to the search string1.

As it happens, many of the articles were not relevant for the search and did not satisfy the inclusion criteria. For what concerns the ACM digital library and Wiley Interscience, we only dispose of the general results, because the search process was not documented as it was carried out (cell values are marked as Not Available).

For what concerns searches of individual journals and conference proceedings, not included in the databases, we only have the final results of the selected papers (included), and not the detail for each search string. The results are reported in table 3. A total of 129 publications were selected. Papers were then further analyzed for duplications, 33 duplicated papers were found. So, the final sample was of 96 papers (primary sources).

	Crosstalk	International Conference on Software Maintenance	International Conference on Software Engineering	International Symposium on Software Metrics	Conference on Product Focused Software Process Improvement	European Conference on Software Maintenance and

		(ICSM)	(ICSE)	(METRICS)	(PROFES)	Reengineering
Included papers	5	2	1	1	3	2

TABLE 3: Results of inclusion/exclusion criteria of primary studies for journal and conference proceedings

Study Quality Assessment, Extract Required Data

Given the set of included papers, we planned to carry out data extraction by filling in the data extraction tables for each paper. The quality assessment questions (for both empirical and theoretical papers) were included in the data extraction tables. The "quality assessment form" for answering RQ5 was produced separately.

According to the protocol, and following analogous criteria, proven to be effective used in other systematic reviews,. we had agreed that "for each paper a researcher would be nominated at random as data extractor, checker, or adjudicator. The data extractor would be responsible for reading the paper and filling in the form; the checker would read the paper and check that the form was correct. In case of any disagreement in the extracted data between extractor and checker that could not be resolved, the adjudicator would read the paper and make the final decision after discussions with the extractor and checker. Also, roles were to be assigned at random with the following restrictions: No one should be data extractor on a paper they authored; All reviewers should have an equal work load. Extracted data was planned to be stored in word tables, one file per paper, using the data extraction form. After checking the extracted data a single word file containing the final agreed data was to be constructed." [27]

Considering the large number of primary sources, and the limited number of researchers allocated on the review (only 4 researchers dedicated part time to the project), we involved graduate students in the data extraction process. This has allowed us to speed up the process and while students have been nominated as extractors or checkers of a paper, researchers are always either checkers or adjudicators. In this way each paper is read by two students and final comments are revised by a researcher. Also, it assures that conflicts are solved with experts. Details of how graduate students have been involved in the process are reported in a separate section (section 4).
We were not able to retrieve all the papers (only 58 of the 96 total), since our university does not subscribe to some of the journals or document databases. In this case, we have contacted the authors or other Italian libraries and are waiting for a response. Also, the list above (table 2 and table3) does not include secondary sources. This search is still in progress. At the current state, the data extraction step has been carried out on a preliminary set of 24 papers, based on the resources and the time available in this first phase of the review.

Synthesize Data

Although this step has not been completed yet, it is worth describing how we intend synthesizing the collected data. Procedures have all been defined in the protocol, together with the data synthesis tables which show the relationship between the research questions (RQ1, …, RQ4) and the results of the review.
For RQ1, table 4 will synthesize the portion of papers addressing each type of software activity, point out whether some activities are considered more appropriate for SPC than others, and address the range of SPC construction and monitoring methods recommended in software.

Activity type (e.g. Testing, Development, Inspections, Maintenance)	Reference	Metrics (e.g. Productivity rate, Defect Counts Defect rate, Inspection effort Inspection Rate)	What type of control chart is used (for empirical studies)	What type of control chart is recommended

TABLE 4: software activities proposed for control by SPC and recommended control methods

For RQ2, table 5 will be completed. We expect to have several different entries for each software characteristic corresponding to different adjustment approaches. We will calculate the proportion of theoretical and empirical studies that made no adjustment for software specific characteristics. RQ3 and RQ4 will synthesize data respectively in table 6 and table 7 with entries that point out benefits and limitations of SPC.

Software Characteristic	References	Adjustment

TABLE 5: Adjustments for software features

Benefits	References	Evidence

TABLE 6: Benefits of software SPC

Limitations	References

TABLE 7: Limitations of software SPC

3.3. Documenting the Review

This phase is the conclusive part of the review, important for communicating the results of the study. It can be done in formats such as technical reports, journal or conference papers, as well as non technical articles or web pages. At the current stage, our systematic review is at the data extraction step (within the 'conducting the review' phase). So, this paper represents the first form of communication of our results carried out on 24 papers, although they are only preliminary, as it can be seen from the content of the paper itself. Further results will be communicated once data extraction and data synthesis are concluded.

With refer to RQ1, 4% of the analyzed papers were classified as strictly empirical, 17% theoretical, and 79% contained both empirical and theoretical elements. Figure4 illustrates the main software activities that SPc is applied to. It can be seen that they involve testing (26%), code development and review (19%), and design and planning (13%). Furthermore, the most used control method (figure 5) is the XmR chart (37%), probably because it is the easiest one to use and because it applies to single points rather than to groups of observations.

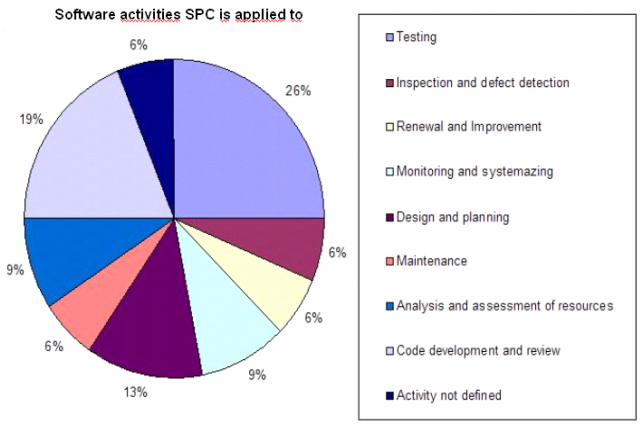

Figure 4: Software activities SPC is applied to

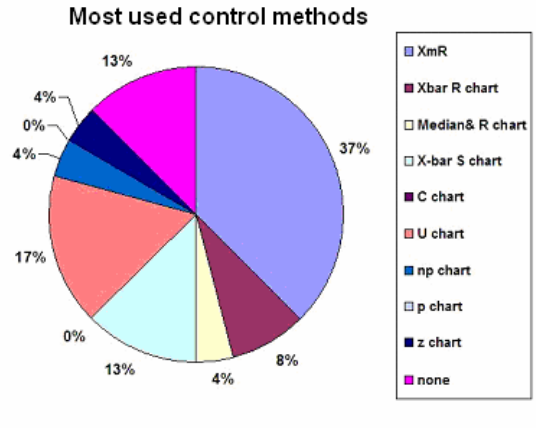

Figure 5: Control charts used in SPC

Given the software characteristics, RQ2 has been difficult to answer RQ2. We are further investigating the data. In general, an important aspect emerged from the papers is the presence of small and non homogeneous datasets, or with missing or non normalized data. Solutions to such problems suggest using project rather than product metrics and control charts that manipulate single rather than groups of observations.

Benefits pointed out by the papers (RQ3) include possibility of studying process performances and estimating future performances; SPC points out anomalous behaviours in process performances and allows to enact improvements before it is too late; SPC is a decision support strategy and introduces a systematic approach to software production and improvement. The analyzed literature also pointed out some limitations: it is difficult to apply to software given the small dimensions of datasets, must be applied to large projects; SPC is not mature enough for software and does not completely adapt to software processes as to manufacturing ones; low quality of data; incorrect use of control charts and stability tests.

RQ4, RQ5 and quality assessment are being processed.

4. DATA EXTRACTION WITH GRADUATE STUDENTS

Given the large amount of primary source papers and the small number of researchers, dedicated to the project only part-time, we decided to involve graduate students in the data extraction process.

The graduate students involved in the study are students at their first year of a MSc degree, all with a BSc degree in informatics or engineering. They all attended a course on Empirical Studies in Software Engineering held by the authors of this work. Also, SPC is part of their course program, so students were familiar to the topic. Therefore our effort was to introduce them to systematic review. Students participated on a volunteer basis. We defined a schedule similar to the one adopted in occasion of the International Advanced School on Empirical Software Engineering (IASESE 2005), with the difference that we had more time available for training students and receiving feedback before assigning them the reviews.

The schedule is reported in table 8.

Day 1	Systematic review guidelines	We introduced the systematic review methodology according to the guidelines in [26]
	SPC systematic review protocol	Students were introduced to the protocol so they could familiarize with all the material although only involved in the data extraction process.
Day 2	Experiences of undertaking a systematic review	Some examples of systematic review carried out in literature were presented to the students.
	Revise SPC concepts	A general overview of SPC
Day 3	Group work – guided exercise	Students were assigned a paper on SPC and were asked to extract data from the primary source, fill in extraction forms and aggregate data
	Group work – feedback on guided exercise	Correctly completed forms were handed out to students. Obtained results were discussed in groups of 2 and then with the class.

Day 4	Assignment of papers	Selected primary sources were assigned to students. They worked individually at home.
Day 5	Group work on assigned papers	Students that worked on the same paper individually confronted their data extraction tables and aggregation tables with other students.

TABLE 8: Schedule for systematic review on SPC with graduate students

First we introduced the students to systematic review and presented the guidelines illustrated by Kitchenham [26], then we presented our protocol on SPC and outlined all the details, tables, extraction procedures, inclusion/exclusion and quality assessment criteria. We also illustrated some examples of reviews carried out in literature, supported by published papers and technical reports [9, 27]. Although SPC is part of the students course program, we though it was the case to "refresh" their minds on the topic, so we dedicated a lesson on the concepts that would appear in the papers to revise. Before actually assigning the papers of the review, we decided to carry out a guided exercise. This was important for allowing students to familiarize with the documentation they would have to use for their final assignment.

In the guided exercise students were asked to extract data from either of two primary sources [8, 22] (which we ourselves selected) according to the data extraction form and aggregation tables defined in the SPC protocol. They were handed the following material: data extraction form, data aggregation tables, research questions of the systematic review, one of the two primary studies. So, for the exercise, half of the students were assigned to a paper [8] and half to another [22]. Students split into pairs and each pair carried out the following tasks: each pair member extracted the data from the paper independently; the pair members compared their data collection forms; any disagreements were solved or noted as disagreements to discuss with the rest of the class and with the researchers; all pairs that had worked on the same paper joined together and completed the aggregation tables. Once the tasks were completed, we distributed the correct complete data extraction forms for the two papers to the students. These papers had already been analyzed and forms had previously been completed. Results were discussed in class with other groups, that had worked on the same paper, and with the researchers.

Overall, feedback of the guided exercise was positive. Students became familiar with the data extraction tables. Most of the data they extracted was correct. Some students found it difficult to understand the meaning of the cells in the tables. Further explanations of the data extraction forms were provided. Also, we decided to translate the cell content in Italian as well, although answers were to be reported in English. Due to language problems, two students decided to not continue with the assignment.

Given the positive results of the presentation part and guided exercise, we considered it feasible to continue with our plan. Students were individually assigned to one of the 24 papers. We ensured that data extraction from each paper was assigned to at least 3 students. Next, all students reviewing the same paper met as checkers and confronted their work to obtain a unique version. At this stage a researcher was also assigned as checker, to guarantee that all pairs (and therefore papers) were controlled by an expert; also, conflicts were solved by an adjudicator. A PhD student was also recruited as adjudicator.

5. CONCLUSIONS

This paper represents an experience report on defining and carrying out a systematic review on Statistical Process Control, which is well known in manufacturing contexts but only recently of interest within the software application domain. As so, we have addressed the issue whether SPC is being correctly applied in software production or not. Answers to our research questions have been assessed through a protocol defined according to guidelines in [26]. We have defined a protocol, verified it, and started conducting the review. We considered it useful to involve graduate students in the data extraction phase of the process since they were familiar to SPC, because it is part of their curricula. We consider the training that students were given on systematic review complete and satisfactory because it was based on an international school on empirical software engineering, and because feedback was positive.

Although the definition phase is complete, the conducting phase is still in progress and this paper represents the first documentation of results on the systematic review of an initial set of 24 papers. We are aware that the results are still preliminary and that our considerations are limited to only part of the retrieved literature. Nevertheless, we consider it an important milestone achieved and are of the opinion that it may be a useful occasion for discussion and improvement.

REFERENCES

[1] ACM digital library, http://www.acm.org/

[2] Australian National Health and Medical Research Council. (2000) How to review the evidence: systematic identification and review of the scientific literature. ISBN 186-4960329.

[3] Australian National Health and Medical Research Council. (February 2000) How to use the evidence: assessment and application of scientific evidence. ISBN 0642432952.

[4] Brereton P., Kitchenham B., Budgen D., Turner M., Khalil M., (2005) Employing Systematic Literature Review: An Experience Report. Technical Report TR 05/01, School of Computing & Mathematics, Keele University.

[5] Card D., Berg R.A., (1989), An Industrial Engineering Approach to Software Development. *J. Systems and Software*, 10, 159-168.

[6] Card D., Glass R.L., (1990), *Measuring Software Design Quality*, Prentice Hall

[7] Card D., (1994), Statistical Process Control for Software, *IEEE Software*, May, 95-97.
[8] Cochrane-Collaboration. (2003) Cochrane reviews' handbook. Version 4.2.1.
[9] Dyba T., Kampenes V.B., Sjoberg D., (2006) A systematic review of statistical power in software engineering experiments, *Information and Software Technology*, 48, 745-755.
[10] Ebenau R.G., (1994), Predictive Quality Control with Software Inspections, *Crosstalk*, June.
[11] Emeroteca Virtuale, http://periodici.caspur.it/
[12] Florac W.A., Carleton A.D., (1999), *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, Addison-Wesley.
[13] Florac W.A., Carleton A.D., Bernard J.R., (2000), Statistical Process Control: Analyzing a Space Shuttle Onboard Software Process, *IEEE Software*, July/August.
[14] Florence A., (2001), CMM Level 4 Quantitative Analysis and Defect Prevention, *Crosstalk*, Feb. 2001.
[15] Gardiner J.S., Montgomery D.C., (1987), Using Statistical Control Chart for Software Quality Control. *Quality and Reliability Eng. Int'l*, 3, 40-43.
[16] Glass R., Vessey I, Ramesh V. (2002) Research in software engineering: An analysis of the literature.
Information & Software Technology, 44, 491–506.
[17] Glass R., Vessey I, Ramesh V. (2004). An Analysis of Research in Computing Disciplines. *Communications of the ACM*, 47:89–94.
[18] Hart C. (1998), *Doing a Literature Review: releasing the social science research imagination*, SAGE Publications, London 1998.
[19] IEEE Software.: Process Diversity. July – August 2000.
[20] IEEE Xplore, http://ieeexplore.ieee.org/Xplore/guesthome.jsp
[21] INSPEC, http://www.iee.org/publish/inspec/
[22] Jacob A., Pillai S.K., (2003), Statistical Process Control to Improve Coding and Code Review, IEEE Software, May/June, 50-55
[23] Jalote P., (1999), *CMM in Practice: Processes for Executing Software Projects at Infosys*, Addison-Wesley.
[24] Jorgensen M, Shepperd M., (2007), A Systematic Review of Software Development Cost Estimation Studies, IEEE Transactions on Software Engineering, 33 (1), 33-53.
[25] Kahan Khalid S., ter Riet Gerben, Glanville Julia, Sowden Amanda J., Kleijnen Jo. (March 2001), Undertaking Systematic Review of Research on Effectiveness; CRD's Guidance for those Carrying Out or Commissioning Reviews. CRD's Report Number 4 (2nd Ed), NHS Centre for Reviews and Dissemination, University of York. ISBN 1900640201.
[26] Kitchenham, B. (2004) Procedures for Performing Systematic Reviews. Technical Report TR/SE0401, Keele University, and Technical Report 0400011T.1, National ICT Australia.
[27] Kitchenham B., Mendes E., Travassos G., (2006), A systematic review of Cross vs. Within company cost estimation studies. *Proceedings of the 10th International Conference on Evaluation and Assessment in Software Engineering*, Keele University Staffordshire, UK, 10-12 April, pp.79-88, BCS, UK, ISBN 1-902505-74-3
[28] Staples M., Mahmood N., (2006), Experiences Using Systematic Review Guidelines. *Proceedings of the 10th International Conference on Evaluation and Assessment in Software Engineering*, Keele University Staffordshire, UK, 10-12 April, pp.79-88, BCS, UK, ISBN 1-902505-74-3
[29] Lantzy M.A., (1992), Application of Statistical Process Control to the Software Process. *Proc. 9th Washington Ada Symposium on Ada: Empowering Software Users and Developers*, July.
[30] Paulk M.C., (2001), Applying SPC to the Personal Software Process, *Proceedings of the 10th International. Conference on Software Quality*, October
[31] Radice R., (2000), Statistical Process Control in Level 4 an 5 Organization Worldwide. *Proceedings of the 12th Annual Software Technology Conference*, (Also at http://www.stt.com/)
[32] Science Direct, www.sciencedirect.com/science/search/allsources
[33] Shewhart W.A., (1980), *The Economic Control of Quality of Manufactured Product*, D. Van Nostrand Company, New York, reprinted by ASQC Quality Press, Milwaukee, Wisconsin.
[34] Shewhart W.A., (1986), *Statistical Method from the Viewpoint of Quality Control*. Dover Publications, Mineola, New York.
[35] Springer Link, http://www.springerlink.com/home/main.mpx
[36] Web of Science, http://scientific.thomson.com/products/wos/
[37] Weller E., (1995), Applying Statistical Process Control to Software Maintenance. *Proc. Applications of Software Measurement*
[38] Weller E., (2000), Practical Applications of Statistical Process Control, *IEEE Software*, May/June, 48-55
[39] Weller E., (2000), Applying Quantitative Methods to Software Maintenance, *ASQ Software Quality Professional*, 3 (1).
[40] Wiley Interscience, www.interscience.wiley.com/
[41] Wohlin C., Runeson P., Höst M., Ohlsson M. C., Regnell B., Wesslen A., (2000), *Experimentation in Software Engineering : An Introduction*. Kluwer Academic Publishers.
[42] Zultner R.E., (1999), What Do Our Metrics Mean?, *Cuttler IT J.*, 12.(4), 11-19.