A Strategy for Statistical Process Control Education in Computer Science

Julio Cezar Costa Furtado
Departament of Exact and Technological Sciences
Federal University of Amapá
Macapá, Amapá, Brazil
e-mail: furtado@unifap.br

Sandro Ronaldo Bezerra Oliveira
Graduate Program in Computer Science
Federal University of Pará
Belém, Pará, Brazil
e-mail: srbo@ufpa.br

***Abstract*—SPC (Statistical Process Control) was originally developed in the area of manufacturing, to support the implementation of continuous improvement programs in production lines. The use of SPC in process improvement is not new to the industry in general. In the context of software organizations, statistical control can be regarded as a relatively recent subject-area, although there are still many doubts about its application. Thus, this paper focuses on drawing up a curriculum for Statistical Process Control within a Computer Science course. The aim of this initiative is to provide software development organizations with computer science professionals who are fully able to perform this task.**

Keywords-software engineering; measurement and analysis; statistical process control; software process improvement; software quality; education.

I. INTRODUCTION

According to Lantzy [1], Statistical Process Control (SPC) was originally carried out in the area of manufacturing, to support the implementation of continuous improvement programs in production lines. The underlying principles of SPC are based on the works of Sherwar [2] in the Bell Telephone Laboratories. A process considered for statistical control must be stable and repeatable [3]. Thus, the SPC comprises a set of techniques that can achieve this goal.

Despite the fact that the use of SPC in process improvement is not new to the industry, in the context of software organizations, statistical control can be regarded as something that is relatively recent [4], although there are still many doubts about its application [5][6][7]. However, the importance of SPC for the software industry has grown in recent years, mainly owing to its use of internationally recognized quality models [8]. The Capability Maturity Model Integration (CMMI) [9] maturity level 4 requires an organization to manage the execution of its processes quantitatively and seek its continuous optimization.

In addition to this introductory section, which covers aspects of the justification and identification of the issue under study, this paper is structured as follows: Section II will formulate the problem statement of this PhD and examine the related work, together with its limitations, in Section III, a number of research questions and hypotheses will be raised. Section IV will outline the research methods employed to answer the research questions and test the hypothesis, and the contribution made by this paper and expected results are discussed in Section V.

A. A Gap in the Area

The reason why the use of SPC in software development organizations has proved to be complex is that these techniques exist in a context that does not take account of the particular features that currently exist in a software development process [8]. As a result, many software organizations tend to seek the assistance of outside professionals in the area of Computer Science to assist in the execution of statistical process control. It thus fails to draw on the expertise of Computer Science professionals who form a part of the hierarchy of an organization. Owing to their position in the software development organization, these professionals are real experts who have a full understanding of the processes and the reality of an organization, as well as being active agents in them.

This difficulty in finding Computer Science professionals may be due to the type of training these professionals are given, and the approach that is adopted for teaching SPC during their graduation course. The studies by Soare [9], Castro *et al.* [10] and Hazzan and Dubinsky [11] have shown that there are situations where there is too much content to teach in a short time. This leads to low motivation among the students and makes it difficult for teachers to prepare students to engage in professional practices within the academic environment.

B. Scope of the Research

With regard to the quality of the development process and the CMMI reference model [12], in the early levels of these improvement programs, organizations employ the measurement techniques that simply consist of collecting data from the execution of the project and comparing them with the planned values. Although this may seem a reasonable approach, it is not suitable for organizations that seek high maturity, to evaluate and evolve their processes. In these organizations, it is necessary to perform statistical control of software processes to find out about their behavior, determine their performance in previous executions and predict their performance in current and future projects. At the same time, it is necessary to make sure that they are able to achieve their established goals and take corrective measures and make improvements when appropriate [13].

Thus, this quantitative management of the organizational process involves a process that should lead to a controlled and predictable performance [12]. In this context, this PhD research study will:

- examine the Statistical Process Control practices that are relevant to the software industry. This involves conducting an analysis to determine what knowledge about SPC is required for professionals in this area to carry out their activities with maximum performance,

- identify the existing SPC practices in computer courses. The results of this analysis will be used to define a new approach for the teaching of SPC in computer courses, which will be aligned with the recommendations of CMMI [12] and the needs of the software industry.

C. *Identification of the Issue*

The great difficulty of employing these professionals for statistical process control is the fact that most of them do not have the necessary knowledge and skills for such an undertaking. This leads to a paradoxical situation when assessing the performance of computer professionals, since their basic education often covers the discipline of Probability and Statistics. Moreover, the many disciplines of Software Engineering (SE) / Software Quality should provide a solid basis that is enough to allow this professional to act with more confidence in the market when there is a need for statistical process control in an organization. The industry complains that undergraduate courses do not teach the necessary skills that can enable students to undertake their jobs efficiently [14]. In the case of most companies, up to 80% of their employment opportunities are taken up by incoming students (freshmen), and up to 80% of the training budget is spent on them [37].

In general, the software industry suffers from a lack of qualified professionals who are able to work in activities involving the software development process [15][38][39]. Thus, although this may not be a global problem, it is, at least, a problem in developing countries. In addition, both Association for Computing Machinery (ACM) / Institute of Electrical and Electronics Engineers (IEEE) Curricula [18] and Brazilian Computer Society (SBC) Reference Curricula [22] fail to address specific aspects of Statistical Software Process. This difficulty is particularly accentuated when dealing with the activities of an organization that have a high maturity level in their processes.

D. *Statement of the Position*

The Brazilian software industry has widely adopted the use of quality models, such as CMMI for Development (CMMI-DEV) [12] and the Brazilian Reference Model for Software Process Improvement (MR-MPS-SW) [16], which normally takes place through the Process Improvement Software (SPI) programs implemented by consultancy firms.

The consultants adopt several training strategies for the transfer of the knowledge that is needed to the specific practices or expected results included in the process area, specifically in Statistical Process Control. The purpose of this is to develop the skills and competencies needed in the technical team of the organization.

The consultants noticed that so much time and resources were spent on training programs for effective teamwork, that many of these professionals do not have adequate knowledge in areas of Software Engineering [17].

On the basis of these observations and our professional experience as SPI consultants and SE lecturers, we believe that if the Software Engineering discipline adopted training approaches to Statistical Process Control, the students would be better prepared for the high maturity software industry and have more appropriate skills than those provided by the current teaching approaches.

II. Problem Statement and Related Works

The research problems of this PhD thesis can be categorized into two groups, which are as follows:

- The need to analyze the SPC practices used in the software industry, i.e. to determine which activities are really relevant to the training of a software engineering professional,
- The need to analyze the reference curricula and teaching approaches used by teachers in the area, to identify which SPC activities are covered.

A. *The Background of Software Engineering Teaching*

According to the ACM / IEEE [18], SE is a discipline that is concerned with the application of theory, knowledge and practice to the effective and efficient development of software systems that meet user requirements. The SE professionals must have the ability to understand software development as a process and to ensure deadlines are met, costs are reduced, and the quality of the product is maintained [19].

A survey was conducted by Wangenheim and Silva [14] to gauge the opinion of professionals in the Software Engineering area about the relevance of the topics covered in the Computer Science courses. As a result, it was found that some SE topics are neglected. In certain topics, there was clearly a complete lack of attention paid by the lecturers and students. For example, there was limitied interest in “Software Configuration Management”, which in practice is considered to be an essential basis not only for software engineers, but also for any professional software [14].

On the other hand, despite the importance of this knowledge with regard to the activities of SE, in [20], it was found that professionals learn more about these activities during their work than from their university courses / education. This may be due to the simple question of the study schedule. Assuming that out of a total of at least 280 hours for a Computer Science course [21], about 36 hours are allocated to SE, this does not correspond to the general recognition of the importance of these subject-areas and thus not enough time can be spent on areas of real importance.

There are studies in the literature on project-based learning within computer courses as a means of learning soft skills and complex technical competencies. A teaching approach has been put forward as a means of integrating contextualized project experiences with the basic concepts of Software Engineering. This approach, called the Software Enterprise, tends to represent the most common

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-498-5

system that is employed to teach these skills and competencies during an undergraduate course and is a software development practice in SE disciplines. This approach makes it more possible to be aware of the fruits of one's labor than when working within student teams as they attempt to "put it all together" and produce a real software product [22].

Gary *et al.* [22] propose a pedagogical model for the teaching of SE in undergraduate courses in computing. This involves the students attending lectures and practising the learned concepts in lab sessions performed in each week of the course. This proposal combines traditional classroom activities with Problem-Based Learning (PBL). In this approach, the professor plays the role of a coach and the "veteran" students perform the role of a mentor. We believe that Gary's pedagogical model [22] provides the main supporting evidence for this research.

B. Problems Area

SE professionals working in the industry have expressed dissatisfaction with the level of preparedness of recently graduated students entering the job market [23][24]. Software companies find they have to complement the knowledge of recent graduates with training and have to provide technical and non-technical skills related to the software development process [25].

According to Lethbridge *et al.* [23], this failing in the training of graduates in the SE area is the result of an inadequate education. This finding is corroborated by the research carried out by Sargent [26], which reveals that: (i) only 40% of Information Technology (IT) professionals in the United States have training in this area, (ii) 40% of those are aware of the main fields of SE, such as requirements, architecture, testing, human factors, and project management.

Although we did not find statistical data with regard to SPC, we believe that the reality of SE professionals in this specific area is no different, from the situation observed by the authors of this paper on numerous consulting assignments involving the implementation of CMMI high levels.

C. Limitations of Related Works

Studies such as [22][25][27] propose teaching approaches to software engineering. However, these approaches restrict the scope of the evaluation to the content of the course or the process and the resulting product of carrying out a practical discipline of Software Engineering. This means that these approaches do not adequately prepare students for working in organizations of software development.

In [25], a game is designed that simulates real-world environments in the software industry to support the learning of SE. Although the game really succeeds in empowering students, it restricts this learning to cover only certain curricular areas. The approach of [25] does not properly explore the topics of Software Engineering, and thus fails to cover the area of statistical process control. By adopting an overly complex approach, it misses the point of the game which is the teaching and learning process.

A multidisciplinary approach is proposed in [27], which offers a set of guidelines that can be applied in the teaching of many disciplines of Computer Science. However, this approach is of limited value because its only concern is with the application and teaching of curriculum topics of Computer courses. It does not bother to find out what are the real professional skills that students must acquire to be able to perform their tasks efficiently in a real software project.

We also evaluated the [28][29][30][31] courses with regard to their instructional strategies for teaching the SE topics. As a result, we were able to observe the effects of combining traditional lectures with indirect instruction through academic projects. We also found evidence of the use of other instruments that could be used for experimental learning by means of simulations and interactive instruction by peer group learning. However, the focus of all these studies is on their approaches to teaching Software Process Modeling (SPM), without offering proven instrumental alternatives that can be used in SPC teaching.

Finally, Gary *et al.* [22] propose a pedagogical model for SE teaching in undergraduate courses in computing.

III. Questions, Hypotheses and Discussion

The main goal of our research is to propose a teaching approach to Statistical Process Control in computer courses, and this approach is based on quality standards and the recognized market practices.

Although we did not find many references, apart from [8] that support our innitiative, it is based on our field observations as SPI consultants and lecturers on SE, (as described in section I.D), and we strongly believe that a better approach to training SPC will improve the preparation of students and enable them to work in high maturity software companies.

To this end, the following research questions (RQ) should be addressed:

- **RQ1.** *What are the Statistical Process Control practices that are most widely used by the software industry?*
- **RQ2.** *What are the Statistical Process Control topics covered in the curriculum guidelines of computer courses?*
- **RQ3.** *What are the Statistical Process Control topics covered in the curricula of computer courses?*
- **RQ4.** *What are the Statistical Process Control topics that are effectively learned by computer students?*
- **RQ5.** *What are the Statistical Process Control skills required by the software industry and which of them were acquired in the computer courses?*

These research questions were defined in an attempt to refute the following null hypothesis:

- **H0.** *The current approach to teaching Statistical Process Control meets needs of the the software industry.*

If the null hypothesis is refuted, we intend to test our alternative hypothesis:

- **H1.** *The current approach adopted to teach Statistical Process Control does not meet the needs of the software industry owing to the lack of an alignment between the Software Engineering curriculum and the real needs of industry.*

IV. Research Method and Progress

The research methods employed to answer the research questions and to test the hypothesis will be outlined in this section.

A. *Identifying SPC Practices Relevant to the Software Industry*

To answer our RQ1 and help RQ5, there will be a systematic review with the aim of determining which SPC practices used by the software industry, can be regarded as following these guidelines [40][41]. This will allow us to create a catalog based on the main practices selected from this review. This catalog will form a part of the analysis that will either support or refute hypothesis H0, by providing the necessary information about the real needs of the software industry. We are currently working on this systematic review.

B. *Identifying SPC Practices Included in Computer Courses*

To answer RQ2, we will conduct a literature review of the the curriculum guidelines from ACM / IEEE [18] and SBC [21] with the aim of establishing which Software Process topics are covered by them. The results of this review will either support or refute hypothesis H0, by giving us evidence that the process activities suggested in the curriculum guidelines meet the requirements of the software industry.

To answer RQ3, a survey will be conducted with the lecturers of undergraduate Computer courses. The purpose of this survey is to analyze which SPC activities found in the literature review are included in the SE syllabus. These results may validate the H0 hypothesis.

With regard to RQ4, another survey will be conducted with students that completed their Software Engineering courses. This survey aims to assess whether the students are learning SPC activities, and if these are covered in SE syllabus. The results might validate H0 too, if they provide evidence that the problem may be in the teaching approach adopted in the classrooms.

Both surveys will be applied to undergraduate Computer Science courses in both public and private universities in Brazil and will follow the guidelines of Kitchenham and Pfleeger [42].

C. *Defining an Approach to SPC Teaching in Computer Courses*

After the five research questions had been answered, they yielded the following results:

- The catalog of SPC practices used by the software industry,
- The recommendations for the guidelines of the SPC curriculum,
- The existing teaching approaches to SPC in computer courses,
- Which topics are considered important by industry.

These results will guide the development of our SPC teaching approach. The approach will adopt Problem Based Learning as a teaching method [25]. The application of PBL in engineering education is increasing [32], and has been recognized as one of the most effective ways for students to become active participants in design learning [33]. PBL is combined with lectures, and focuses on the application and integration of knowledge that has been acquired previously [34].

On the basis of the results obtained in [19], it is clear that the students are more interested in carrying out practical activities, such as development projects in laboratories that simulate situations close to those that will be found in the market. It is believed that this is due to the fact that software engineering has many topics (83 in total), which ultimately make it less attractive to students who do not have an affinity with the area. The practical approach allows these students to grasp the concepts more easily from their application.

D. *Performing Case Studies to Evaluate the Teaching Approach*

Our proposal of a teaching approach can be evaluated and validated by conducting a controlled experiment in an area of software engineering in a Computer Science undergraduate course. Initially, we plan to carry out this experiment in two computer science classes in two Brazilian public universities.

This experiment will follow the guidelines proposed by Wohlin [37].

V. Conclusion and Future Works

This paper has shown the importance of statistical process control in software development, focusing on a PhD research project that aims to study on drawing up the curriculum for Statistical Process Control within the Computer Science course. This initiative is to provide development organizations software with computer science professionals who are fully able to perform this task. Thus, this paper showed the discussions about the gap in this area, the problems of statistic process control in computer science, some related works, the research questions and hypotheses, and the method used to conduct this research.

The aim of this PhD research study is to identify practices of Statistical Process Control and their relevance to the current software industry, by taking note of possible problems in the teaching approach to software engineering, in particular in SPC. This can be achieved through the implementation of the curriculum guidelines and a new educational approach that ensures that undergraduate students of computer courses will be given a background training in SPC, as required by the market.

The current educational scene shows that certain topics are regarded as less important by teachers and thus students have a low learning levelin these areas. It turns out that these topics impose a heavy workload on software engineering, while some topics considered to be more important, have a low learning priority. Perhaps this is due to the fact that there is not enough time to teach all these units effectively.

Finally, we intend to allow free access to the results that have been obtained in this research, so that the experiments and results obtained can be replicated. This should ensure that more gaps in our knowledge are filled and problems solved and clarified, since the teaching of Software Engineering is of the utmost importance in computer courses [36].

ACKNOWLEDGMENT

The authors would like to thank the Dean of Research and Postgraduate Studies at the Federal University of Pará (PROPESP/UFPA) by the Qualified Publication Support Program (PAPQ) for the financial support.

REFERENCES

[1] M. A. Lantzy, "Application of Statistical Process Control to the Software Process", In: Proceedings of the 9th Washington Ada Symposium on Empowering Software Users and Developers, ACM Press, pp. 113-123, 1992.

[2] W. Shewhart, "Economic Quality Control of Manufactured Product", Bell System Technical Journal, vol. 9, pp. 364-389, 1930.

[3] D. C. Montgomery, "Introduction to Statistical Quality Control". John Wiley & Sons, 2007.

[4] M. A. Alhassan and D. N. Jawawi, "Sequential Strategy for Software Process Measurement that Uses Statistical Process Control", in: 8th Malaysian Software Engineering Conference (MySEC), pp. 37-42, 2014.

[5] F. García, M. Serrano, J. Cruz-Lemos, F. Ruiz, and M. Piattini, "Managing Software Process Measurement: A Metamodel-Based Approach", Information Sciences, v. 177, n. 12, pp. 2570-2586, 2007.

[6] N. Boffoli, G. Bruno, D. Caiavano, and G. Mastelloni, "Statistical Process Control for Software: a Systematic Approach", Proceedings of the Ninth European Conference on Software Maintenance and Reengineering, pp. 288-293, 2008.

[7] A. Tarhan and O. Demirors, "Assessment of Software Process and Metrics to Support Quantitative Understanding", Lecture Notes in Computer Science, v. 4895, pp. 102-113, 2008.

[8] C. Fernández-Corrales, M. Jenkins, and J. Villegas, "Application of Statistical Process Control to Software Defect Metrics: an Industry Experience Report", in: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 323-331, 2013.

[9] M. Soares, "An experience in teaching software engineering oriented practical work", I Workshop on Computer Education, Vitória / Rio das Ostras, Brazil, 2004.

[10] J. Castro, I. Gimenes, and J. Maldonado, "A proposal for pedagogical plan for the Software Engineering discipline", II quality course of undergraduate of Computing and Informatics area, Curitiba, Brazil, pp. 251-270, 2000.

[11] O. Hazzan and Y. Dubinsky, "Teaching a software development methodology: The case of Extreme Programming", 16th Conference on Software Engineering Education and Training, Madrid, Spain, pp. 176-184, 2003.

[12] Software Engineering Institute, "CMMI® for Development, Version 1.3, Improving processes for developing better products and services", No.CMU/SEI-2010-TR-033. Carnegie Mellon University, Pittsburgh, PA, 2010.

[13] M. P. Barcellos, R. A. Falbo, and A. R. Rocha, "Establishing a Well-Founded Conceptualization about Software Measurement and High Maturity Levels", in: 2010 7th International Conference on the Quality of Information and Communication Technology, pp. 467-472, 2010.

[14] C. Wangenheim and D. Silva, "What software engineering knowledge is important for the professional software", in Proceedings of II Forum on Education in Software Engineering, Fortaleza, Brazil, 2009.

[15] ABES, "Brazilian software market: scenario and trends", in Brazilian Software Market and Services 2014, 1st ed. São Paulo, Brazil: Brazilian Association of Software Companies, 2014.

[16] Association for Promotion of Brazilian Software Excellence, "SPI Reference Model for Software, version 2012", SOFTEX, General Guide SPI for Software, Dec. 2012.

[17] G. Leal, P. Stadzisz, C. Almeida, M. Perez, S. Reinehr, A. Malucelli, "Empirical study about the evaluation of the implantation of MPS.Br in enterprises of Paraná", In Proceedings of XXXVIII Conferencia Latinoamericana en Informatica, pp. 1-9, 2012.

[18] ACM/IEEE. Computer science curricula 2013. Curriculum guidelines for undergraduate degree programs in Computer Science. December 20, 2013.

[19] C. S. Portela, A. M. L. Vasconcelos, and S. R. B. Oliveira, "How to Develop Competencies and Abilities Professional in Software Engineering in Undergraduate Students?", in Int'l Conf. Frontiers in Education: CS and CE - FECS'15, pp. 91-94, 2015.

[20] T. LethBridge, "What knowledge is important to a software professional?", Journal Computer, 33(5), IEEE Computer Society Press, Los Alamitos, CA, USA, pp 44-50, 2000.

[21] SBC. Reference curriculum for undergraduate courses in Bachelor in Computer Science and Computer Engineering, 2005.

[22] K. Gary, T. Lindquist, S. Bansal, and A. Ghazarian, "A Project Spine for Software Engineering Curricular Design", In Proceedings of 26th Conference on Software Engineering Education and Training (CSEET), pp. 299-303, 2013.

[23] T. Lethbridge, J. Diaz-Herrera, R. LeBlanc, and J. Thompson, "Improving software practice through education: Challenges and future trends", Conference Future of Software Engineering, Minneapolis, MN, pp.12-28, 2007.

[24] T. B. Hilburn, and M. Towhidnejad, "A case for software engineering", 20h Conference on Software Engineering Education and Training, Dublin, Ireland, pp. 107-114, 2007.

[25] B. Bessa, M. Cunha, and F. Furtado, "ENGSOFT: Simulation tool for real environments to support the Problem Based Learning (PBL) in teaching Software Engineering", XX Workshop on Computer Education, Curitiba, Brazil, 2012.

[26] J. Sargent, "An overview of past and projected employment changes in the professional IT occupations", Computing Research News, vol. 16, no. 3, pp. 1-21, 2004.

[27] J. Braga, "Guidelines for the interdisciplinary teaching of Software Engineering", in Proceedings of II Forum on Education in Software Engineering, Fortaleza, Brazil, 2009.

[28] M. L. Jaccheri and P. Lago, "Applying Software Process Modeling and Improvement in Academic Setting", Proceedings of the 10th Conference on Software Engineering Education & Training, Virginia Beach, Virginia, IEEE Computer Society Press, pp 13-27, 1997.

[29] D. Groth and E. Robertson, "It's All About Process: Project-Oriented Teaching of Software Engineering", in Proceedings of the Fourteenth Conference on Software Engineering Education and Training. Charlotte, USA, pp 7-17, 2001.

[30] J. Hawker, "A Software Process Engineering Course", Proceedings of the 2009 American Society for Engineering Education Annual Conference. Austin, TX, 2009.

[31] C. Wangenheim and J. Hauck, "Teaching Software Process Improvement and Assessment", Proceedings of 17th European Systems & Software Process (EuroSPI'2010). Grenoble, France, 2010.

[32] H. Hadim and S. Esche, “Enhancing the engineering curriculum through project-based learning”, Proceedings of 32nd Frontiers in Education Conference. Boston, USA, Section F3F, pp 1-6, 2002.

[33] C. Dym, A. Agogino, O. Eris, D. Frey, and L. Leifer, “Engineering Design Thinking”, Teaching, and Learning. Journal of Engineering Education, 94(1), pp 103–120, 2005.

[34] J. Mills and D. Treagust, “Engineering education: is problem-based or project-based learning the answer?”, Australasian Journal of Engineering Education, pp. 2-16, 2004.

[35] C. Wohlin et al., “Experimentation in software engineering: an introduction”, in Kluwer Academic Publishers, Norwell, MA, 2000.

[36] IEEE Computer Society, Guide to the Software Engineering Body of Knowledge – SWEBOK, Version 3.0, 2014, Available: www.swebok.org, retrivied: 06/2016.

[37] K. Garg andV Varma. “Software Engineering Education in India: Issues and Challenges”, Proceedings of 21st Conference on Software Engineering Education and Training, Charleston, pp., 110-117, 2008.

[38] C. O’Leary, D. Lawless, D. Gordon, L. Haifeng, and K. Bechkoum, “Developing a Software Engineering Curriculum for the Emerging Software Industry in China", 19th Conference on Software Engineering Education & Training(CSEET'06), pp. 115-122, 2006.

[39] G. Taran and M. Rosso-Llopart, “Software Engineering Education in Russia: A Comparative Study of People, Process and Technology: a Four Year Perspective”, 20th Conference on Software Engineering Education & Training (CSEET'07), 2007.

[40] B. Kitchenham, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE - 2007-01, Departament of Computer Science Keele University, Keele, pp. 19-28, 2007.

[41] T. Dyba, T. Dingsoyr, and G. Hanssen, “Applying Systematic Reviews to Diverse Study Types: An Experience Report”, First International Sysmposion on Empirical Software Engineering and Measurement, pp. 225-234, 2007.

[42] B. Kitchenham and S. Pfleeger, Personal Opinion Surveys, in Guide to Advanced Empirical Software Engineering, Springer, 2008.