To appear in: The proceedings of the 12th International Software Process and Capability dEtermination Conference (SPICE 2012), Palma, Mallorca, May 2012. Springer-Verlag, Berlin, CCIS 290, pp. 62-74.

A hierarchy of SPI activities for software SMEs: results from ISO/IEC 12207-based SPI assessments

Paul Clarke [1], Rory V. O'Connor [2, 3] and Murat Yilmaz [1]

[1] Lero Graduate School in Software Engineering, Dublin City University, Ireland
[pclarke, myilmaz]@computing.dcu.ie
[2] Dublin City University, Ireland
[3] Lero, the Irish Software Engineering Research Centre
roconnor@computing.dcu.ie

Abstract. In an assessment of software process improvement (SPI) in 15 software small- and –medium-sized enterprises (software SMEs), we applied the broad spectrum of software specific and system context processes in ISO/IEC 12207 to the task of examining SPI in practice. Using the data collected in the study, we developed a four-tiered pyramidal hierarchy of SPI for software SMEs, with processes in the higher tiers undergoing SPI in more companies than processes on lower level tiers. The development of the hierarchy of SPI activities for software SMEs can facilitate future evolutions of process maturity reference frameworks, such as ISO/IEC 15504, in better supporting software development in software SMEs. Furthermore, the findings extend our body of knowledge concerning the practice of SPI in software SMEs, a large and vital sector of the software development community that has largely avoided the implementation of established process maturity and software quality management standards.

Keywords: Software development process, SPI, software SMEs.

1 Introduction

Software process maturity frameworks such as ISO/IEC 15504 [1] and the Capability Maturity Model Integrated (CMMI) [2] provide structured and proven paths to improved process maturity. Software process maturity is "the extent to which a specific software process is explicitly defined, managed, measured, controlled and effective" [3], with higher levels of process maturity being associated with higher product quality, reduced production costs [4], and with increased predictability of the process results [5], [6]. Although process maturity reference frameworks can deliver benefits to any type of software development organisation, evidence from earlier studies suggests their adoption would appear to be mostly concentrated in large organisations [7], [8]. Some earlier research has investigated the reasons for non-adoption of process maturity reference frameworks in the software SME sector, with the finding that software SMEs view process maturity frameworks as being infeasible (i.e. overly time-consuming or costly to implement) rather than non-beneficial [9-11].

Although software SMEs tend to not implement process maturity frameworks, they nonetheless require a software development process in order to produce and maintain software products. The software process can be implemented in a formal or informal manner, but in order to best address the needs of the organisation “it is reasonable to assume that the optimal process is not static but is organization-dependent and time-dependent, and will have to be modified as the context in which the organization operates evolves” [12]. With organisational context regularly changing, companies need to continually adapt their software development processes in order to maximise the efficiency and effectiveness of their software development efforts. However, despite the obvious theoretical benefits of adopting a strong software development process focus, evidence from recent studies suggests that in practice, software SMEs can adopt a low process priority [13], tending only to implement SPI in response to negative business events [14]. Given this gap between the theory and practice, it is the view of the authors of this paper that we need to develop a much greater understanding of SPI as practiced in software SMEs.

In order to develop a better understanding of SPI as practiced in software SMEs, we designed a study that investigates SPI across the broadest possible range of software development processes. As we shall present in this paper, our study permits the development of a hierarchy of process areas in terms of their importance for SPI in software SMEs. The development of this classification extends our knowledge of SPI as practiced in software SMEs, and provides valuable information that can assist future evolutions of process reference frameworks in addressing the needs of software SMEs.

The remainder of this paper is structured as follows: Section 2 provides an overview of our study, including details of the approach to data collection and the participating organisations; Section 3 presents a hierarchy of SPI as practiced in software SMEs, along with some recommendations for future research directions and field studies. Finally, Section 4 discusses the importance of the findings along with some concluding remarks.

2 Study Overview

The study presented in this paper is primarily concerned with examining the extent of SPI implemented in software SMEs over a 12 month period. In order to examine the extent of SPI implemented in an organisation, it is possible to utilise the process assessment vehicles associated with process maturity reference frameworks. For example, an ISO/IEC 15504 assessment could be conducted at the commencement of the 12 months under investigation, hence establishing the process maturity at the start of the period under investigation. At the end of the period of investigation, a second process assessment could be conducted, this time establishing the process maturity at the end of the year. A comparative analysis of the two process assessment results could thereafter being employed so as to determine the amount of SPI conducted during the elapsed period of time.

While two process assessments, followed by a finite difference analysis, could be employed in order to determine the amount of SPI implemented in an organisation, there are a number of reasons why this approach is considered unsuited to the needs of this study. Firstly, this study is concerned with examining SPI in software SMEs, a sector that has traditionally declined to implement process maturity reference frameworks. Secondly, process assessments are concerned with collecting data in relation to process maturity. Although this study is interested in examining the extent of SPI implemented in software SMEs, it is not concerned with the degree process maturity in the participating organisations and therefore, the collection of process maturity data would represent an inefficient approach to collecting the data required for this research. A third reason for not employing process assessments in this research relates to the time requirement for the discharge of two process assessments. Gaining access to participating organisations is a difficult challenge for researchers and the research team felt that the large amount of time required to conduct two process assessments might act as a further disincentive for the candidate software SMEs who might consider participating in this study.

Given the limitations of process assessments as outlined above, we formulated a new, more efficient approach to examining the amount SPI implemented in software SMEs, an approach that does not require the collection of process maturity data and a subsequent finite difference analysis. Conceptually, our new approach requires that an organisation is asked to identify the instances of SPI as implemented in their organisations over the past 12 months. In order for such an approach to be reliable, it is important that as a basis, the instrument of inquiry is developed from a recognised, comprehensive process reference framework. Although a number of candidate reference frameworks exist, it is the view of the authors that no single framework offers greater scope than ISO/IEC 12207 [15].

2.1 ISO/IEC 12207 Software Development Process Reference Framework

ISO/IEC 12207 identifies a comprehensive set of software development processes – covering not just the core activities related to software developed (which ISO/IEC 12207 terms *Software specific processes*) but also including the additional related processes required for the housing of software development activities in the broader processes that are required for conducting business (which ISO/IEC 12207 terms *System context processes*). In total, there are 43 individual processes identified in ISO/IEC 12207, with these processes being broken down into over 400 process tasks (refer to Figure 1).

As well as offering a broad set of core and supporting software development processes, ISO/IEC 12207 is also considered to be suited to the needs of this study because of the consensual nature of the development and maintenance of the process reference framework. This consensual nature is exemplified by the approach adopted by the International Organization for Standardisation (ISO) when drafting, accepting and evolving standards – whereby 75% of the participating national bodies must approve a standard prior to publication [15]. In addition to the rigorous and consensual nature of the ISO's standard acceptance criteria, ISO/IEC 12207 has also been developed in collaboration with the Institute of Electrical and Electronic Engineers (IEEE) Computer Society, hence further ensuring that ISO/IEC 12207 is comprehensive in nature and generally accepted by the broader software development community.

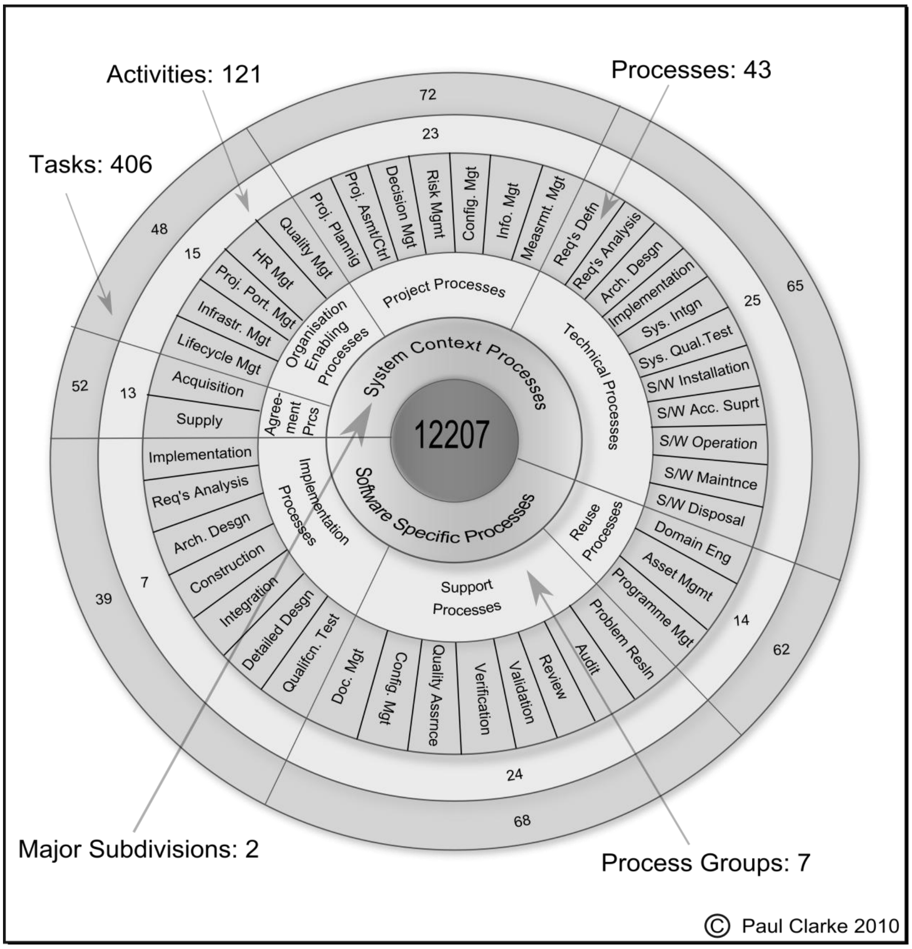

Fig. 1. ISO/IEC 12207 Topology

In addition to the reasons cited above for adopting ISO/IEC 12207 for this study, a further important consideration relates to the independence of ISO/IEC 12207. Since a number of software SMEs will participate in this study, and considering that they may all adopt different process development methodologies, it is important that the chosen software development process reference framework is independent of any particular, prescribed software development approach. Since ISO/IEC 12207 provides a "meta-model that defines common software engineering activities independently of a particular life-cycle model" [16], it is considered to offer an ideal reference framework for the type of study that we wanted to conduct. While ISO/IEC 12207 is well suited to the requirements of this study, the standard does not natively present in a form that permits the investigation of the amount of SPI implemented in a software development organisation. Therefore, this study developed a systematic method for converting ISO/IEC 12207 to a survey instrument suited to the task of investigation the amount of SPI conducted in a company over a period of time.

2.2 Technique for converting ISO/IEC 12207 to a SPI survey instrument

In order to adapt ISO/IEC 12207 [15] for the purpose of investigating the amount of SPI implemented in software SMEs over a period of time, it is necessary to first fully analyse the standard, becoming intimately familiar with all of the details that are contained within the standard. These details are first extracted to a master list of questions that can be employed in order to determine the amount of SPI in an organisation over a period of time. Since ISO/IEC 12207 incorporates a large degree of cross-referencing of processes, the next step in the survey instrument development is to remove instances of duplication in the questions. Following the removal of duplications, the list of questions is further distilled in order to meet practical considerations – for example, the time required to discharge the survey instrument.

Subsequent survey instrument development steps include a reordering of the questions so as to ensure that the survey instrument has a suitable flow, and the engagement of external expert reviewers. In the case of the independent expert review, current and former editors of ISO/IEC 12207 along with other experts familiar with ISO/IEC 12207, are engaged so as to examine multiple aspects of the draft SPI survey instrument, including content, scope and look and feel. Following feedback from the independent expert reviewers, a number of updates were made to the SPI survey instrument. Thereafter, the survey instrument was subject to a pilot phase with an industry partner, after which the survey instrument was again updated based on the industry feedback. Extensive details on the conversion of ISO/IEC 12207 to the SPI survey instrument are available in an earlier published work [17].

In its final form, the SPI survey instrument contained 63 questions that took the general form of: "*Over the past year, has there been any modification in the approach to…[some aspect of the software development process]?* In responding to the questions, the interviewees were asked to rate the degree of process modification according to a four point Likert scale, as follows: 0 = no modification; 1 = minor modification; 2= major modification; and 3 = significant modification.

Having systematically developed our SPI survey instrument from ISO/IEC 12207, the next step in the study was to secure the engagement of participating software SMEs, followed by a discharge of the survey instrument in order to collect SPI data from the participating organisations.

2.3 SPI data collection

The SPI survey instrument was deployed to a total of 15 participating companies between March and July 2011. Each of the participating organisations satisfied the European Commission definition of an SME [18]. Within each of the participating organisations, a suitable participant was identified; most commonly, the interviewee held the job title of Software Development Manager, Engineering Manager or Director of Engineering – in all cases, the interviewee was identified as the most appropriate person in the organisation to address the broad scope of inquiry covered by the SPI survey instrument. The participating software SMEs varied in terms of the headcount: 3 of the participating companies had less than 10 staff; 4 companies had between 10 and 19 staff; the remaining 8 companies had between 20 and 129 staff.

Predominately, the participating organisations were primarily located in Ireland. However, in some cases, the organisations were mostly located outside of Ireland, in places such as the US and Chile. Where possible, the interviews were conducted face-to-face with telephone interviews being employed in a small number of cases (for example, where the interviewee was based in a remote location). The interviews required approximately 2 hours to complete, giving a total interview time of ~30 hours. Irrespective of whether the interview was conducted face-to-face or via telephone, the interview was (with the consent of the interviewee) recorded and later, the interview recording was carefully examined to ensure that the responses of the interviewee were accurately and complete documented in electronic form.

In addition to generally being extremely busy, candidate organisations were somewhat cautious about revealing information regarding the internal workings of the company. In order to assuage such concerns, a number of procedures were implemented: (1) each of the participating organisations was allocated a random pseudonym such that the identity of the organisations was not divulged; (2) all recordings, be they stored on portable or fixed devices, were securely encrypted; and (3) the researchers developed a bi-lateral non-disclosure agreement that could be employed to further reinforce the confidence of the participating organisation regarding the privacy and security of the data.

Following the completion of the interviews in the 15 participating organisations, the researchers collected a large volume of data in relation to SPI as practiced in software SMEs. The next step in to apply the study data towards the development a hierarchy of SPI activities for software SMEs.

3 Hierarchy of SPI activities for software SMEs

An analysis of the SPI reported in this study permits the development of a hierarchy of SPI processes, as implemented in practice in software SMEs. Since the SPI survey instrument was developed from ISO/IEC 12207 (refer to Section 2), it is possible to map each of the questions in the survey instrument back to the originating process in ISO/IEC 12207. Using this mapping, we constructed a hierarchy of ISO/IEC 12207 processes in terms of the processes being targets for SPI, i.e. processes that constitute the top tier of the pyramidal hierarchy underwent SPI in a greater number of organisations than processes that are on the second tier of the hierarchy; with processes on the second tier of the hierarchy undergoing SPI in a greater number of organisations than processes that are on the third tier, etc. The resulting hierarchy is presented in Figure 2.

From the SPI hierarchy pyramid presented in Figure 2, it can be seen that there are nine key software processes that undergoing SPI most frequently in software SME. Some of these processes have been reported in earlier studies, while others have not. For example, earlier related studies demonstrated that software SMEs derived both short- and long-term benefits from SPI in areas such as requirements analysis, configuration management and project planning [19-23]. However, no earlier published study indicated that infrastructure management, installation, and supply were key process improvement targets for software SMEs. This is perhaps owing to the broad nature of the inquiry in this study. Earlier studies may have focused just on the software specific processes as identified in ISO/IEC 12207. However, the infrastructure management, installation, and supply processes are all system lifecycle processes and therefore, broader in scope than the purely software specific process grouping.

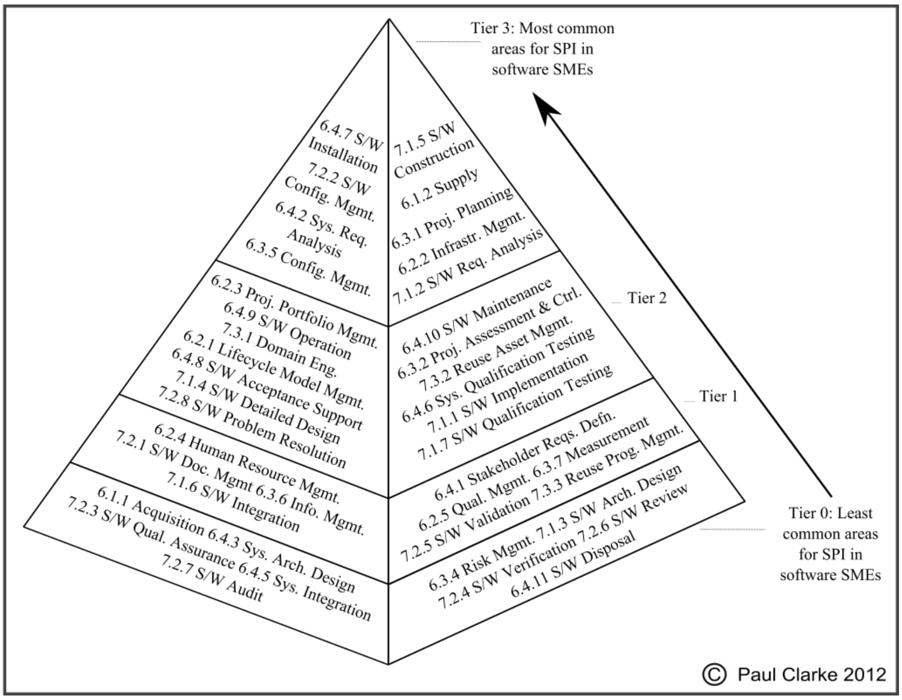

Fig. 2. ISO/IEC 12207 based SPI hierarchy for software SME †

In the lowest tier of the SPI hierarchy, Tier 0, there are ten key software processes. As with the processes in the highest tier of the pyramid, a number of these processes have been identified in earlier studies, while others have not been. For example, earlier related studies have highlighted that software SMEs can adopt a low process focus [13], electing only to implement process improvements in response to negative business events [14]. Therefore, it is not surprising to discover that the software quality assurance, software review, risk management, and software audit processes are all to be found on the lowest tier of the hierarchy – though clearly, for the software process and SPI community, this is certainly disappointing.

On the lowest tier of the SPI hierarchy pyramid, we also find a number of additional processes that do not appear to have been identified in earlier published studies – though it should be noted that earlier related works were not necessarily concerned

† There are 43 individual processes identified in ISO/IEC 12207. The Implementation Process (6.4.4) is wholly defined by the Software Implementation Process (7.1.1) and is therefore not separately included in the hierarchy. Following feedback from independent expert reviewers, the Decision Management Process (6.3.3) was not included in the survey instrument – it was the view of the reviewers that this process was beyond the scope of software SMEs. Therefore, the resulting hierarchy presented in Figure 2 has 41 individual processes included.

with identifying the low priority processes for SPI in software SMEs. The additional low priority processes identified in this study are: acquisition, system and software architecture design, system integration, software verification and software disposal. Of these processes, the inclusion of the software architecture and system architecture design processes is of considerable interest. The feedback from the study participants indicates quite strongly that software SMEs do not necessarily make a distinction between architectural and detailed design efforts. Essentially, the responses from participants indicate that the architecture and detailed design processes are effectively bundled into a single activity. Therefore, this represents a good example of where the ISO/IEC 12207 process reference listing is beyond the scope of software SMEs. While ISO/IEC 12207 does permit that the process selection can be tailored for different settings, it does not permit that two separate processes can be merged into a single process. Therefore, the general infrastructure of the ISO/IEC 12207 framework is perhaps over-extended for the purposes of software SMEs. This finding also has implications for ISO/IEC 15504 which utilises ISO/IEC 12207 as the underlying process reference framework, a finding which we discuss further in section 3.1.

3.1 Recommendations for future SPI assessments and studies

Taking the data from this study, it is possible to develop a number of recommendations for future SPI assessments and studies:

Recommendation 1: In order to maximize the effectiveness of future SPI studies and assessments in software SMEs, it is considered beneficial to treat architectural and design activities as a single activity (rather than as two separate activities as identified in ISO/IEC 12207).

Recommendation 2: In order to maximize the effectiveness of future SPI studies and assessments in software SMEs, where there is strong overlap between system lifecycle and software specific processes (e.g. requirements analysis and configuration management processes), researchers should consider merging the system life cycle and software processes into a single thread of inquiry.

Recommendation 3: Future SPI studies seeking to identify the high priority process improvements for software SMEs should consider omitting those processes that are in Tier 0 of the SPI hierarchy pyramid (ref. Fig. 2).

In the case of Recommendation 1, the evidence of this study indicates that software SMEs rarely differentiate between architectural and design considerations. Future assessments and studies of SPI in software SMEs should not necessarily remove either the design or architecture activities but rather, should merge them into a single thread of investigation. In the case of Recommendation 2, although we find considerable evidence of important SPI occurring to the system life cycle processes of ISO/IEC 12207, SMEs can find it difficult to distinguish between system life cycle level activities and core software specific activities. As indicated by our independent expert reviewers, this can particularly be the case where there is a software specific process that has a corresponding parent system level process; for example, the system requirements analysis process and the software requirements analysis processes. We therefore recommend that where there is a strong overlap between a system life cycle process and a software specific process, these two processes can be merged into a single spoke of inquiry. In relation to Recommendation 3, since our study (along with earlier studies [13], [14]) has indicated that software SMEs can have a low process priority, future studies examining the key processes for SPI in software SMEs should consider omitting processes on Tier 0 of the SPI hierarchy (ref. Figure 2).

3.2 Recommendations for versions of process reference frameworks

We have one further important recommendation, this time regarding future versions of process maturity frameworks. As a research community, we need to start developing new thinking regarding the utility of reference frameworks and quality management standards for the software SME sector. Furthermore, framework authors and those responsible for writing software process standards should consider extending existing process reference and process assessment models to more accurately reflect and support software development as practiced in software SMEs. Although software SMEs are a large and vital component of the overall software development landscape, they have to-date failed to embrace established process maturity and quality management frameworks. Whatever the reason for the failure to adopt these approaches, it seems unlikely at this stage that software SMEs will ever implement these approaches to any great extent. Given the extensive wealth of knowledge, wisdom and experience incorporated into existing process maturity and quality management standards, it is very disappointing that they are not more widely adopted in software SMEs. In an earlier published work, the authors of this paper highlighted the importance of further integration of the dynamic capability concept into process maturity and quality management standards [24]. Dynamic capability relates to the ability to adapt a process in response to changes in the environment and is considered to be representative of an evolutionary mechanism, as espoused in the *evolutionary* theory of the firm [25].

The dynamic capability concept is not entirely absent in existing process maturity frameworks, it is in fact the embodiment of the highest level of process maturity, the *optimising* level. That the concept of optimising is an existing component of process maturity frameworks is considered by the authors to be hugely positive, however, that an organisation should only optimise of the highest maturity level is considered a drawback. The need to optimise, adapt and evolve is a continuous consideration and one for which it may be difficult to fully prescribe a maturity framework – since all successful organisations and organisms are considered successful if they respond to their particular set (as opposed to some general set) of environmental challenges and changes. As Prof. Harvey Fineberg states: "*Evolution… doesn't necessarily favour the biggest or the strongest or the fastest and not even the smartest. Evolution favours those creatures best adapted to their environment, that is the sole test of survival and success*" [26]. Companies, like creatures, must also evolve relative to their particular environment.

Therefore, our fourth recommendation is that we develop new thinking in terms of how process maturity and quality management standards can better assist software SMEs, and to this end, we recommend greater integration of dynamic capability concepts. Naturally, increased integration of dynamic capability into existing process maturity and quality management standards should benefit organisations of all sizes, not just software SMEs.

> **Recommendation 4:** In order to benefit software SMEs, future evolutions of process maturity frameworks and quality management standards should further integrate dynamic capability concepts.

4 Conclusion

In our study of 15 software SMEs, we find that the software quality assurance, software audit, software review and risk management processes are in receipt of very little process improvement focus. The collective lack of attention on these three processes highlights a major impediment for the adoption of process maturity frameworks and quality management frameworks (such as ISO 9000 [27]) in software SMEs. The evidence of this study suggests that software SMEs do not embrace some of the most basic principles of process maturity and quality management frameworks. Such basic principles include the adoption of a process focus, the explicit awareness of process activities (preferably in documented form), the reflection on the efficacy of the adopted software process, and the improvement of the software process in line with recommended process improvement paths. There is therefore a significant gap in the fundamental process thinking promoted by process maturity and quality management frameworks, and the reality of process implementation in software SMEs.

Despite the noted gap in process thinking between software SMEs and process maturity and quality management frameworks, there is some encouragement to be taken from the fact that some process areas were reported as having experienced relatively high levels of SPI. These areas include configuration management, requirements analysis, infrastructure management and project planning. We are therefore presented with something of a conundrum: on the one hand software SMEs appear unwilling to embrace a strong software process focus while on the other hand they do appear to be quite capable of realising instances of software process improvements.

It is the view of the authors of this paper that as a research community, we should work to find new ways to maximize the ability of software SMEs to realise software process improvements. This view is based on the premise that some SPI is better than no SPI at all. Software SMEs have largely failed to implement long-established process maturity and quality management frameworks; with the result that we have good reason to believe that they will continue to avoid adopting such approaches to software processes and SPI. However, as with organisations of all sizes, it is important to continually optimise business processes in software SMEs (incl. the software development process) and consequently, there is an unavoidable need for SPI. Perhaps the extensive knowledge and experience encapsulated in established process maturity and quality management frameworks can be reoriented such that the principle of process optimisation is more central to the basic demands of the frameworks (rather than being a highest maturity level activity alone). Developments in this direction may encourage software SMEs to consider the implementation of software process and SPI frameworks and in so doing would help to improve the competitive advantage and success prospects of software SMEs. Since most software development companies are SMEs, this would appear to be a worthy pursuit.

Acknowledgments. This work is supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

References

1. ISO/IEC: 15504-1 information technology - process assessment - part 1: Concepts and vocabulary. ISO / IEC, Geneva, Switzerland (2004)
2. SEI: CMMI for development, version 1.2. Software Engineering Institute, CMU/SEI-2006-TR-008. Pittsburgh, PA, USA (2006)
3. Paulk, M.C., Curtis, B., Chrissis, M.B. et al.: Capability maturity model for software. Version 1.1 edn. Software Engineering Institute, Carnegie Mellon University, CMU/SEI-93-TR-24. Pittsburgh, Pennsylvania, USA (1993)
4. Harter, D.E., Slaughter, S.A.: Quality Improvement and Infrastructure Activity Costs in Software Development: A Longitudinal Analysis. Management Science, 49 (6), 784-800 (2003)
5. Ferguson, P., Leman, G., Perini, P. et al.: Software process improvement works!. CMU/SEI-99-TR-027. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA (1999)
6. Harrison, W., Settle, J., Raffo, D.: Assessing the Value of Improved Predictability due to Process Improvements. In: 3rd International Workshop on Economics-Driven Software Engineering Research (EDSER-3), IEEE Computer Society, Los Alamitos, California, USA (2001)
7. Gibson, D., Goldenson, D. and Kost, K.: Performance results of CMMI-Based Process Improvement. Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2006-TR-004. Pittsburgh, Pennsylvania, USA (2006)
8. Herbsleb, J., Goldenson, D.: A systematic survey of CMM experience and results. In: Proceedings of the 18th International Conference on Software Engineering (ICSE 1996), pp. 323-330. IEEE Computer Society, Los Alamitos, California, USA (1996)
9. Saastamoinen, I., Tukiainen, M.: Software Process Improvement in Small and Medium Sized Software Enterprises in Eastern Finland: A State-of-the-Practice Study. In: Proceedings of the 11th European Conference on Software Process Improvements (EuroSPI 2004), pp. 69-78. Springer-Verlag, LNCS 3281/2004. Berlin / Heidelberg, Germany (2004)
10. Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., Murphy, R.: An Exploratory Study of Why Organizations do Not Adopt CMMI. Journal of Systems and Software, 80 (6), 883-895 (2007)
11. Khurshid, N., Bannerman, P., Staples, M.: Overcoming the First Hurdle: Why Organizations Do Not Adopt CMMI. In: Proceedings of the International Conference on Software Process (ICSP 2009), pp. 38-49. Springer-Verlag, LNCS 5543/2009. Berlin / Heidelberg, Germany (2009)
12. Poulin, L.A.: Achieving the Right Balance between Process Maturity and Performance. IEEE Canadian Review, 56 (-), 23-26 (2007)
13. Baddoo, N., Hall, T.: De-Motivators for Software Process Improvement: An Analysis of Practitioners' Views. Journal of Systems and Software, 66 (1), 23-33 (2003)
14. Coleman, G., O'Connor, R.: Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software, 81 (5), 772-784 (2008)
15. ISO: Amendment to ISO/IEC 12207-2008 - systems and software engineering – software life cycle processes. ISO, Geneva, Switzerland (2008)

16. Tilley, T., Cole, R., Becker, P. et al.: A survey of formal concept analysis support for software engineering activities. Springer-Verlag, IN: Formal Concept Analysis. LNCS 3626/2005. Berlin / Heidelberg, Germany (2005)
17. Clarke, P., O'Connor, R.: Harnessing ISO/IEC 12207 to examine the extent of SPI activity in an organisation. In: Proceedings of the 17th Conference on European Systems & Software Process Improvement and Innovation (EuroSPI2010), pp. 25-36. Springer-Verlag, Heidelberg / Berlin, Germany (2010)
18. European Commission: Commission Recommendation of 6 may 2003 Concerning the Definition of Micro, Small and Medium-Sized Enterprises. 2003/361/EC. Official Journal of the European Union, L (124), 36-41 (2003)
19. Sanders, M. (ed.): The SPIRE handbook. better, faster, cheaper software development in small organisations. Centre for Software Engineering Limited, DCU, Dublin, Ireland (1998)
20. Sanders, M., Richardson, I.: Research into Long-Term Improvements in Small- to Medium-Sized Organisations using SPICE as a Framework for Standards. Software Process: Improvement and Practice, 12 (4), 351-359 (2007)
21. Cater-Steel, A., & Rout, T.: SPI long-term benefits: Case studies of five small firms. In: H. Oktaba (ed.): Software Process Improvement for Small and Medium Enterprises - Techniques and Case Studies. Information Science Reference, London, United Kingdom (2008)
22. Fleck, D.: A Process for Very Small Projects. In: Proceedings of the 22nd Annual Pacific Northwest Software Quality Conference, pp. 107-115. PNSQC/Pacific Agenda, Portland, Oregon, USA (2004)
23. Montoni, M., Rocha, A.R.: A Methodology for Identifying Critical Success Factors That Influence Software Process Improvement Initiatives: An Application in the Brazilian Software Industry. In: Proceedings of the 14th European Conference on Software Process Improvement, pp. 175-186. Springer-Verlag, Heidelberg/Berlin, Germany (2007)
24. Clarke, P., O'Connor, R.V.: An Approach to Evaluating Software Process Adaptation. In: Proceedings of the 11th International Conference on Software Process Improvement and Capability Determination (SPICE 2011), pp. 28-41. Springer-Verlag, Heidelberg / Berlin, Germany (2011)
25. Nelson, R.R., Winter, S.: An evolutionary theory of economic change. The Balknap Press of Harvard University Press, Cambridge, Massachusetts, USA (1982)
26. H. Fineberg, Are we ready for neo-evolution? http://www.ted.com/talks/harvey_fineberg_are_we_ready_for_neo_evolution.html
27. ISO: ISO/IEC 9003:2004 - software engineering - guidelines for the application of ISO 9001:2000 to computer software. ISO, Geneva, Switzerland (2004)