Harnessing ISO/IEC 12207 to examine the extent of SPI activity in an organisation

Paul Clarke [1], Rory O'Connor [2, 3]

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

Abstract. The quality of the software development process directly affects the quality of the software product. To be successful, software development organisations must respond to changes in technology and business circumstances, and therefore software process improvement (SPI) is required. SPI activity relates to any modification that is performed to the software process in order to improve an aspect of the process. Although multiple process assessments could be employed to examine SPI activity, they present an inefficient tool for such an examination. This paper presents an overview of a new survey-based resource that utilises the process reference model in ISO/IEC 12207 in order to expressly and directly determine the level of SPI activity in a software development organisation. This survey instrument can be used by practitioners, auditors and researchers who are interested in determining the extent of SPI activity in an organisation.

Keywords: SPI, Survey, ISO/IEC 12207.

1 Introduction

The past two decades have witnessed significant growth in the software development business and in parallel there has been a sustained investment in research into the process of software development. One of the principal developments in the software process domain has been the emergence of prescriptive process maturity models, such as ISO/IEC 15504 [1] and the Capability Maturity Model Integrated (CMMI) [2]. Such models present a progressive roadmap for maturing the software development process, and employ process assessments in order to provide a process capability rating. The initial stage on a maturity model roadmap generally represents a state of low process implementation, with subsequent stages gradually enhancing the process implementation, finally culminating with the process optimisation stage, wherein the software development process is continually being optimised in order to best address the software development needs of the organisation.

The ability to optimise a process is related to a theory from the field of economics, the *evolutionary* theory of the firm [3]. This theory is centred on the concept of *dynamic capability*, which suggests that with knowledge, skills and experience accumulating over time, it is the ability to continually learn from the accrued understanding that gives rise to the dynamism that will ultimately propel the organisation to success [4]. The firm, therefore, is promoted as "a locus where competencies are continually built, managed, combined, transformed, tested and selected", where the vital consideration relates to how "new knowledge [is] materialised in new competencies", and where "a lock-in to inefficient routines" is perceived as a major threat to a company's prospects [5]. Consequently, a dynamic capability to transform routines is considered to provide a basis for competitive advantage [5], a point that has already been observed in relation to the software development routines by Poulin [6], who suggests that with respect to software process capability, establishing an organisation's ability to optimise the development process may provide a better approach than traditional audits. Therefore, rather than examining process capability and prescribing an improvement path, an alternative view suggests that one should focus on maximizing the capability to transform the process, and that this transformational capability will render an improved process.

If dynamic capability is a key ingredient for company success, as is suggested by the evolutionary theory of the firm, then software development companies should be dynamically capable with respect to the software development process. In order to examine the dynamic capability with respect to the software development process, it is necessary to determine the extent of SPI activity in an organisation. For the purpose of this paper, SPI activity is defined as *"the set of SPI actions implemented by an organisation, which is manifested as a series of modifications to the software development process"*. The ISO/IEC 12207 [7] based SPI activity survey instrument developed and presented below can be used to examine the extent of SPI activity, providing a direct and valuable insight into the degree of SPI activity in an organisation. With the software development process constituting an important and complex component of the overall business process for software developing organisations, and acknowledging the importance of dynamic process capability as encapsulated in the evolutionary theory of the firm [3], software development and quality management practitioners, as well as auditing agents, could apply the SPI activity survey instrument in order to directly determine the extent to which the software development process is being evolved. Researchers can also use the SPI activity survey instrument, and the authors of this paper are presently applying the approach as part of a broader research project that is examining the influence of SPI on the evolution of small to medium sized (SME) software development companies [8].

This remainder of this paper is structured as follows: section two provides some background on the contemporary adoption of SPI models in practice and why such models are not well suited to examining SPI activity, followed by section three which presents a new method for transforming an international standard into a survey instrument, as well as the application of this method to transform ISO/IEC 12207 [7] into an SPI activity survey; finally, section four presents a conclusion.

2 Motivation

Although the successful management of the software development process is important for business success, there is a lack of adoption of published models to support the development of software [9, 10, 11, 12], with some research suggesting that temporal contextual factors are critical in identifying the most appropriate process [13,14], especially in SMEs [15]. It should also be acknowledged that even if SMEs did implement SPI models such as ISO/IEC 15504 [1] and CMMI [2], it is not likely that they would consistently achieve the highest, optimising level. This, however, is not sufficient reason to abandon the pursuit of the dynamic capability described in the evolutionary theory of the firm [3], and perhaps a case could be made that process optimisation as a principle should be more integrated into all levels of process maturity reference models.

Although process maturity reference models are not widely implemented, it is possible to utilise the process assessment vehicles associated with these models in order to directly determine the amount of SPI activity. This would involve conducting two process assessments on two different dates, and thereafter performing a finite difference analysis on the assessment results. However, this twin assessment approach has a number of drawbacks. Firstly, it requires two engagements with the software development organisation, which is time consuming and which can be difficult to orchestrate from a practical researching perspective. Secondly, process assessments, such as those in ISO/IEC 15504 [1] and CMMI [2] would collect data related to process maturity rather than just SPI activity and therefore represent a somewhat inefficient tool for evaluating SPI activity. Thirdly, adopting an ISO/IEC 15504 [1] or CMMI [2] process assessment vehicle to determine SPI activity might diminish the capacity to secure candidate participants in the SME sector, since prescribed process maturity reference models have themselves already met with resistance to implementation in SMEs. For the three reasons outlined above, traditional process assessments are not considered efficient vehicles for making express SPI activity determinations.

Taking these drawbacks into account, and owing to the apparent absence of any established dedicated resource for determining the amount of SPI activity, this paper presents a new method for evaluating SPI activity, a method based around the application of a dedicated SPI activity survey instrument.

3 Evaluating SPI activity using a dedicated survey instrument

In the case of ISO/IEC 15504 [1], the ISO/IEC 12207 [7] process listing is used as the underlying process reference list. ISO/IEC 12007 [7] is an internationally developed and maintained listing for software processes and therefore represents a useful reference point when examining software processes in any setting.

It is the premise of this paper that in order to evaluate the amount of SPI activity in an organisation, ISO/IEC 12207 [7] can be used as a comprehensive point of reference. However, the creation of a survey instrument based on ISO/IEC 12207 [7] needs to be structured and systematic, and this paper presents an approach suited to converting an international standard into a survey instrument, followed by an explanation of how the method was applied in the case of transforming ISO/IEC 12207 [7] into an appropriate survey instrument for evaluating the extent of SPI activity in an organisation.

3.1 Method for converting an international standard to a survey instrument

Many international standards consist of verbose text that seeks to accurately and completely describe an item of technical matter. However, such comprehensive text-based descriptions are not easily fashioned into survey instruments, especially when practical considerations, such as the time required to conduct the survey, are taken into consideration. Therefore, this paper presents a technique for resolving verbose text-based international standards back to comprehensive, yet practical, survey instruments. An overview of this technique is presented in Figure 1.

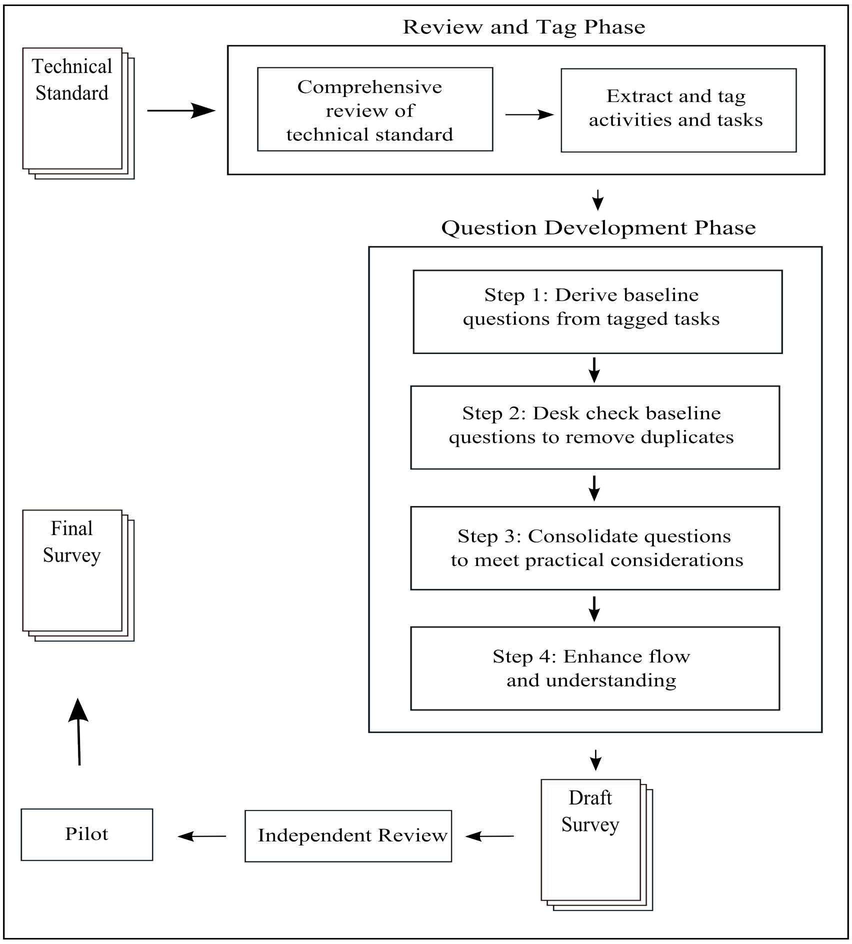

Fig. 1. Survey Instrument Development Technique

The initial phase, the Review and Tag phase, involves reviewing the international standard, so as to develop a thorough understanding of all the material comprising the standard. Thereafter, the various components of the international standard are tagged – in order to identify the key activities and artefacts. This requires that close attention is paid to all actions and artifacts in the international standard, ensuring that no important detail is overlooked.

Following the tagging exercise, the Question Development phase is undertaken. This is a four-step activity that involves transforming the tagged details, as output from the initial phase, into a representative, accurate, comprehensive and readable survey instrument. Notes that explain any modifications, along with rationale for changes, must be maintained at each step in the question development phase – this allows for later examination of the survey construction exercise, including the possibility of auditing the artefacts so as to verify that appropriate decisions have been taken throughout the survey construction activity. Such artefacts can thereafter be published along with the survey findings if required.

The first step of the question development phase involves using the tagged details in order to derive a baseline set of questions. This results in a baseline suite of questions that preserve all of the essential details that are present in the international standard itself. In the second step of the question development phase, the baseline suite of questions is desk-checked so that any duplications or areas of overlap are resolved. This is necessary in order to efface cross-references that can exist in international standards.

The third step of the question development phase consolidates the list of questions with respect to practical considerations. The target survey duration is among the practical considerations, and the survey constructor must judge the appropriate type and number of questions for the survey. The consolidation of questions also requires a considerable deal of judgement, coupled with expertise, on the part of the survey constructor, but should nonetheless seek to preserve the original makeup and structure of the international standard, retaining all major components such that the resulting survey is clearly identifiable as a derivative of the original standard. Having consolidated the questions in an appropriate fashion, the fourth and final step of the question development phase involves reviewing the survey so as to enhance the clarity of individual questions and to optimise the flow of the survey so as to best achieve the survey objectives.

Having completed the question development phase, the survey constructor presents a draft version of the survey instrument to software process and process standards domain experts so as to elicit independent feedback on the content, accuracy, and likely effectiveness of the interview in obtaining the required information.

Following completion of the independent review, the survey instrument should be revised so as incorporate the feedback from the expert reviewer. Once again, a copy of the changes applied should be maintained so as to allow for later examination of the technique.

3.2 Application of conversion method to ISO/IEC 12207

This section outlines the application of the survey instrument development technique, presented above, to the development of an SPI activity survey instrument based on the ISO/IEC 12207 [7] international standard.

3.2.1 Review and Tag Phase

A comprehensive review of ISO/IEC 12207 [7] reveals that the standard consists of seven process groups, forty three processes, one hundred and twenty one activities, and four hundred and six individual tasks. Tasks represent the finest level of detail, with ISO/IEC 12207 [7] defining a task as a "requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more outcomes of a process"; while an activity represents a grouping of "the set of cohesive tasks of a process". The topology of ISO/IEC 12207 [7] is outlined in Figure 2.

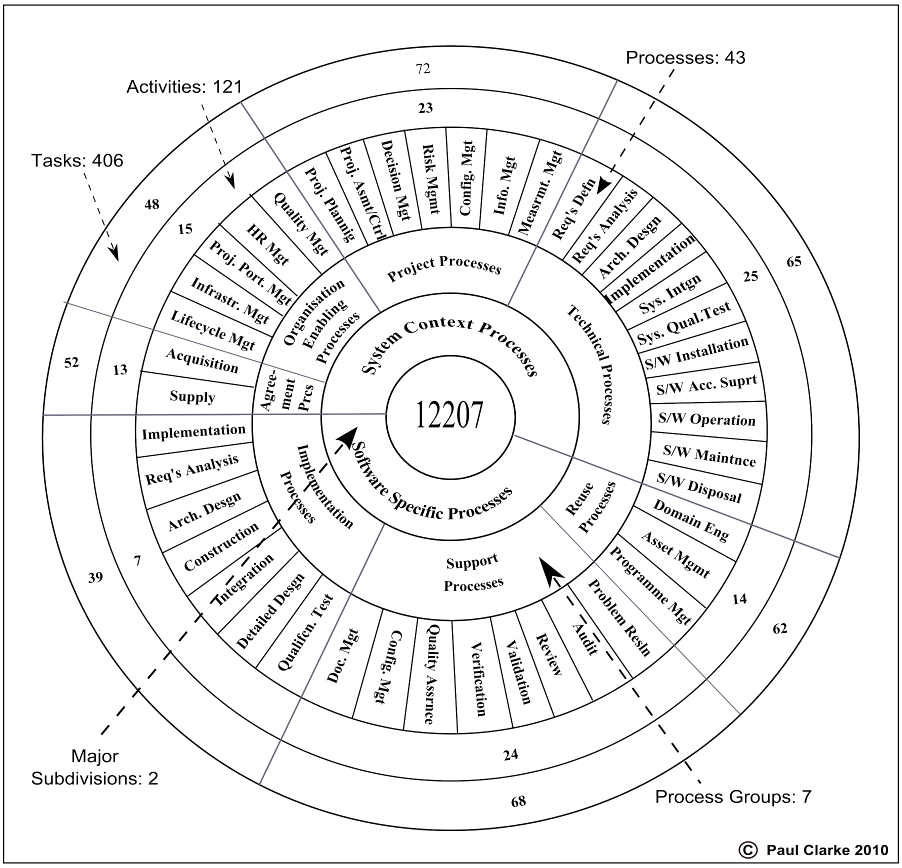

Fig. 2. ISO/IEC 12207 Topology

The completion of the comprehensive review of ISO/IEC 12207 [7], and the development of a clear understanding of its constituent parts, permits the commencement of the tagging stage. For the purpose of this paper, one of the forty three processes, the *Software Implementation* process, is used to demonstrate the development of the survey instrument from its original form as a list of activities and tasks in ISO/IEC 12207 [7] into its final rendering as a set of questions in a survey.

In ISO/IEC 12207 [7], the *Software Implementation* process consists of one activity, the *Software Implementation Strategy*, which is further broken down into five individual tasks. These tasks are tagged, as shown in the highlighted text in Figure 3.

Software Implementation Process	Activities	Tasks	
	1. Software Implementation Strategy	1	If not stipulated in the contract, the developer shall **define** or select a **life cycle model** appropriate to the scope, magnitude, and complexity of the project
		2	The implementer shall: a) **Document** the **outputs** in accordance with the Software Documentation Management Process; b) Place the **outputs** under the **Software Configuration Management** Process and perform **change control** in accordance with it; c) **Document** and **resolve problems** and **non-conformances** found in the software products and tasks in accordance with the Software Problem Resolution Process; d) **Perform supporting processes** as specified in the contract; e) **Establish baselines** and **incorporate configuration** items at **appropriate times,** as determined by the acquirer and the supplier
		3	The implementer shall **select, tailor,** and **use** those **standards, methods, tools,** and computer **programming languages** (if not stipulated in the contract) that are documented, appropriate, and established by the organisation for performing the activities of the Software Implementation Process and supporting processes
		4	The implementer shall **develop plans** for conducting the activities of the **Software Implementation** process. The plans should include specific **standards, methods, tools, actions,** and **responsibility** associated with the **development** and **qualification** of **all requirements** including safety and security
		5	**Non-deliverable items** may be employed in the **development or maintenance** of the software product

Fig. 3. Software Implementation Process Tagging

3.2.2 Question Development Phase

Step one of the question development phase involves the construction of a set of baseline questions using the tagged task items from the initial review and tag phase. In the case of the *Software Implementation* process, the baseline set of questions are as depicted in Figure 4.

Regarding Software Implementation, describe any modifications that have been applied to the approach to:

- The life cycle model definition
- Documenting the software
- Configuration management of outputs
- Performance of change control
- Documenting and resolving problems and non-conformances discovered in the software
- Performing supporting processes as defined in contracts
- Establishing baselines and incorporating configuration items at appropriate times
- Selecting, tailoring and using standards, methods, tools and programming languages
- Developing plans for software implementation, including standards, methods, tools, actions and responsibilities associated with the development and qualification of all requirements
- The employment of non-deliverable items in the development or maintenance of the software product

Fig. 4. Software Implementation Process: Question Development Step 1

As per the survey instrument development technique described earlier, the baseline questions are desk checked to remove any duplicate items. In the case of the software implementation process baseline questions, the life cycle model, software documentation, configuration management, problem resolution, change control, support processes and the establishment of baselines items are all covered in more detail elsewhere in ISO/IEC 12207 [7]. For example, the configuration management, problem resolution, documentation management, life cycle management, and support process are all afforded there own explicit process in ISO/IEC 12207 [7]. Consequently, these items are identified as duplicates and removed from the software implementation questions. The resulting set of questions is as depicted in Figure 5.

Regarding Software Implementation, describe any modifications that have been applied to the approach to:

- Selecting, tailoring and using standards, methods, tools and programming languages
- Developing plans for software implementation, including standards, methods, tools, actions and responsibilities associated with the development and qualification of all requirements
- The employment of non-deliverable items in the development or maintenance of the software product

Fig. 5. Software Implementation Process: Question Development Step 2

Step 3 of the question development phase involves the reduction of the question burden in order to meet practical considerations. This SPI activity survey instrument is designed to take a maximum of two hours to complete and therefore, considering the type of questions required, eighty-five individual questions is set as the threshold and target for the question burden in the final survey product. The reduction of questions in the case of this survey instrument has applied the following principle: the original forty three processes of ISO/IEC 12207 [7] must be retained and obvious in the final survey.

During the *Software Implementation* process question reduction, no reductions were performed on the questions that were output from step two of the question development phase. However, question reduction was carried out in other areas of the survey instrument. For example, six *Software Disposal* process baseline questions were resolved back to a single question – as depicted in Figure 6 and Figure 7. The rationale for this reduction is based on the judgement of the survey constructor, believing that software disposal is not likely to be a detailed and organised event for software SMEs. However, the single question that is retained ensures that the software disposal process is not overlooked in the survey, and the retention of many of the keywords from the original six questions provides for suitable trigger points for survey participants. In this way, much of the concentration of detail from ISO/IEC 12207 [7] in relation to software disposal is retained while also satisfying the practical survey duration consideration.

Regarding Software Disposal, describe any modifications that have been applied to the approach to:

- Defining and documenting a software disposal strategy
- Executing a software disposal plan
- Notifying users of the plans and activities for the retirement of software products and services
- Operating retiring and new software products in parallel for smooth transition to a new system
- Notifying all concerned parties regarding the scheduled retirement time
- Accessing data associated with retired software products in accordance with contract and data protection/audit requirements

Fig. 6. Software Disposal Process: Question Development Step 2

Regarding Software Disposal, describe any modifications that have been applied to the approach to:

- Defining and executing a software disposal strategy, which may include the parallel operation of retiring and new systems, the notification of associated activities, and the control of access to data associated with retired software products in accordance with contract and data protection/audit requirements

Fig. 7. Software Disposal Process: Question Development Step 3

The gradual development of the survey instrument up to the completion of step three of the question development phase has witnessed a phased consolidation of the detailed task information in ISO/IEC 12207 [7] into a survey instrument that can be practically discharged. The various versions of the survey instrument development are preserved in the event that later verification is requested and Table 1 provides a summary of the question development activity.

Table 1. Question Development and Consolidation

Grouping	Number of Questions Step 1	Step 2	Step 3
Agreement Process	6	4	4
Organisational Project-Enabling Processes	15	14	10
Project Processes	23	21	13
Technical Processes	27	21	12
Software Implementation Processes	60	53	30
Software Support Processes	28	20	14
Software Reuse Processes	14	10	6
Total	**173**	**143**	**85**

The fourth and final step of the question development phase involves the adjustment of the survey instrument in order to improve the understandability and flow of the survey. In the case of the *Software Implementation* process presented in Figure 5, the questions were reviewed and updated with a view to ease of understanding, the results of which are presented in Figure 8.

Regarding Software Implementation, describe any modifications that have been applied to the approach to:

- Selecting, tailoring and using standards and methods
- Planning for software implementation, identifying the actions and responsibilities associated with the development and qualification of all requirements
- The employment of non-deliverable items in the development or maintenance of the software product, for example programming languages and tools such as software building tools

Fig. 8. Software Implementation Process: Question Development Step 4

In addition to improving the readability and understandability of individual questions, the survey instrument was re-shaped so as to sequence the questions in a manner that addressed specific details towards the start of the survey, with more general questions placed later. For example, the *Software Implementation* process, a detailed and specific process is placed at the start of the survey, while broader processes such as *Human Resource Management* and *Infrastructure Management* are placed towards the end of the survey. Ordering the questions in this way permits the elicitation of specific details earlier in the survey while also allowing for broader question discussion later in the survey, at which stage the specific details are better understood. In order to validate the SPI activity survey instrument, the first draft of the survey is presented for an independent review, with feedback items from the review – which are broadly similar in nature to the type of items discussed this far, being incorporated into the completed survey instrument. The survey instrument now enters a further validation step, the pilot, during which the instrument is deployed to a software SME. Once again, feedback is incorporated into the SPI activity survey instrument. Finally, the SPI activity survey instrument is ready for practical deployment.

4 Discussion and Conclusion

The quality of the software process directly affects the quality of the software product, and since the technology, business environment and company circumstances are subject to continual change, there is an ongoing requirement for SPI. Existing approaches to SPI, such as ISO/IEC 15504 [1] and CMMI [2] assess the maturity of processes in an organisation. These process maturity reference models prescribe a phased process maturity roadmap, with the earlier stages characterised by minimum process implementation and the later stages gradually improving the process maturity, with the final stage being dedicated to continuous process optimisation.

The concept of process optimisation is related to the evolutionary theory of the firm [3], which suggests that the dynamic capability of an organisation to modify its business processes is an important driver for business success. If it is the case that dynamic capability is central to the formula for business success, then software development organisations would benefit from being dynamically capable with respect to the software development process. Maturity models such as ISO/IEC 15504 [1] and CMMI [2] do acknowledge process optimisation as an important attribute, but it is only evident at the most mature stage. Therefore, organisations that adopt such process maturity references models, and who do not progress to the most mature stage, may fail to realise the benefits of dynamic capability as described by the evolutionary theory of the firm [3].

For SMEs in the software development sector, there is an acknowledged lack of adoption of process maturity reference models [9, 10, 11, 12]. However, even if SMEs did implement these models, it is unlikely that they would achieve the highest process rating, and consequently they would be at risk of not realising the benefits of dynamic process capability.

If dynamic capability is important, then there should be method for examining the extent of dynamic process capability in an organisation, and for software development organisations, there should be an approach for examining SPI activity. SPI activity has been defined by this paper as the set of SPI actions implemented by an organisation, and the process assessment vehicles associated with process maturity reference models could be applied in order to carry out this examination. However, such an approach would require multiple assessments and the performance of a finite difference analysis on the individual process assessment results. As well as being time consuming, process assessments collect data that is pertinent to process maturity rather than expressly examining SPI activity, and therefore, they do no represent an efficient method for the examination of SPI activity.

This paper presents an efficient and direct method for examining SPI activity, using a survey instrument that has been systematically derived from the ISO/IEC

12207 [7] process model. The survey instrument can be implemented in a single engagement with a software development organisation, the output from which captures the extent and nature of the SPI activity in the organisation. This approach can be used by software development practitioners to determine the SPI activity in their organisation, and it can also be used as part of a software process assessment. Furthermore, the approach can be used for research purposes and is presently being applied to a research project that is examining the influence of SPI on the successful evolution of software SMEs [8].

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: ISO/IEC 15504-1 Information technology - Process assessment - Part 1: Concepts and vocabulary. ISO, Geneva, Switzerland (2004)
2. CMMI Product Team: CMMI for Development, Version 1.2. CMU/SEI-2006-TR-008. Software Engineering Institute, Pittsburgh, PA, USA (2006)
3. Jacobson, D., Andreosso-O'Callaghan, B.: Industrial Economics and Organization - A European Perspective. McGraw-Hill, London, UK (1996)
4. Chandler, A.D.: Organizational capabilities and the economic history of the industrial enterprise. J. Econ. Perspect. 6(3), 79--100 (1992)
5. Cohendet, P., Kern, F., Mehmanpazir, B., Munier, F.: Knowledge coordination, competence creation and integrated networks in globalised firms. Cambridge J. Econ. 23(2), 225--241 (1999)
6. Poulin, L.A.: Achieving the right balance between process maturity and performance. IEEE Canad. Rev. 56, 23--26 (2007)
7. ISO/IEC: Amendment to ISO/IEC 12207-2008 - Systems and Software Engineering – Software Life Cycle Processes. ISO, Geneva, Switzerland (2008)
8. Clarke, P., O'Connor, R.: Towards the identification of the influence of SPI on the successful evolution of software SMEs. In: Proceedings of SQM 2010. British Computer Society, London (2010)
9. McConnell, S.: Closing the gap. IEEE Soft. 19(1), 3--5 (2002)
10. McAdam, R., Fulton, F.: The impact of the ISO 9000:2000 quality standards in small software firms. Manag. Serv. Qual. 12(5), 336--345 (2002)
11. Ludewig, J.: Software Engineering in the Year 2000 Minus and Plus Ten. Springer-Verlag, Berlin, Germany (2001)
12. Coleman, G., O'Connor, R.: Investigating software process in practice: A grounded theory perspective. J. Syst. & Softw. 81(5), 772—784 (2008)
13. Benediktsson, O., Dalcher, D., Thorbergsson, H.: Comparison of software development life cycles: a multiproject experiment. IEE Proc.-Softw. 153(3), 87—101 (2006)
14. MacCormack, A., Verganti, R.: Managing the sources of uncertainty: Matching process and context in software development. J. Prod. Innov. Manag. 20(3), 212-232 (2003)
15. Kautz, K.: Software process improvement in very small enterprises: does it pay off? Soft. Proc.: Imp. & Prac. 4(4), 209—226 (1998)