Exploring the Relationship between Software Process Adaptive Capability and Organisational Performance

Paul Clarke, Rory V. O'Connor, Brian Leavy, and Murat Yilmaz

Abstract—Software development is a complex socio-technical activity, with the result that software development organisations need to establish and maintain robust software development processes. While much debate exists regarding the effectiveness of various software development approaches, no single approach is perfectly suited to all settings and no setting is unchanging. The capability to adapt the software process is therefore essential to sustaining an optimal software process. We designed an exploratory study to concurrently examine software process adaptive capability and organisational performance in 15 software development organisations, finding that companies with greater software process adaptive capability are shown to also experience greater business success. While our exploratory study of the complex relationship between these phenomena is limited in some respects, the findings indicate that software process adaptive capability may be worthy of further integration into software process engineering techniques. Software process adaptive capability may be an important organisational strength when deriving competitive advantage, and those responsible for the creation and evolution of software process models and methodologies may want to focus some of their future efforts in this area.

Index Terms—Software engineering, software engineering process, software development, software management

1 Introduction

The present software development process (or software process, for short) landscape is constituted by numerous software development frameworks and methodologies. Some of these frameworks are designed to provide consistency of product quality and budgetary/schedule adherence (e.g., CMMI [1], ISO/IEC 15504 [2]). Other frameworks are designed with a strong focus on process quality, with the expectation that product quality will ensue (e.g., ISO-9001 [3]). Agile methods (e.g., Scrum [4], XP [5], Lean Software Development [6]), which are based upon the Agile Manifesto [7], provide a different type of approach again, one that promotes customer interaction and flexibility in the treatment of requirements.

With software development companies being so widespread and varied, it is not surprising to discover that no single process framework or methodology is perfectly suited to all software development settings [8], [9]. Each individual approach has strengths and weaknesses [10], [8], and when implementing a process, the paramount consideration is that it "*should fit the needs of the project*" [11]. Guidance on the type of process that might be required may be found in the Boehm-Turner model [12] which helps to identify the extent to which an agile philosophy may be appropriate. Beyond this general guidance, other research has examined a yet broader set of factors that influence software process decisions [13], [14]. Such factors range from the experience of personnel, to the application under development, and even to broader business and organisational demands.

Since aspects of the world are inevitably subject to change, it is reasonable to assume that an optimal process should require regular adaptation—ranging from minor improvements over time to major redesign depending on the nature and extent of the situational change. Indeed, the fast pace of technological change in the software development domain may demand higher levels of process adaptation than other business sectors.

Numerous earlier research efforts have identified the benefits of process change for software companies. Some of this earlier research focused on the benefit of adopting a capability maturity framework such as CMMI or ISO/IEC 15504 [15], [16], [17], [18], [19], while other studies reported benefits of process improvement in smaller organisations [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], or of using agile software development methods [33], [34], [35], [36], [37], [38], [39], [40], [41], or on establishing hard financial evidence of the benefits of process adaptation [42]. Earlier research also identifies the important role of agile method adaptation in organisations [43]. However, no previously published research effort explicitly examines software process adaptation in light of the situational demand for process change, and no earlier published research presents data from a longitudinal study of the relationship between process adaptation and organisational performance across multiple companies.

Earlier published works indicate that research into process adaptation may be particularly valuable for smaller

- *P. Clarke and R.V. O'Connor are with the School of Computing, Dublin City University, Ireland, and Lero—The Irish Software Research Centre. E-mail: {pclarke, roconnor}@computing.dcu.ie.*
- *B. Leavy is with Dublin City University Business School, Ireland. E-mail: Brian.Leavy@dcu.ie.*
- *M. Yilmaz is with Çanyaka University, Ankara, Turkey. E-mail: MYilmaz@cankaya.edu.tr.*

Manuscript received 24 Feb. 2015; revised 1 July 2015; accepted 27 July 2015. Date of publication 19 Aug. 2015; date of current version 11 Dec. 2015. Recommended for acceptance by H. Sharp. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TSE.2015.2467388

companies, where it has been reported that a low software process priority may persist [44], and where process improvements may only be implemented in response to negative business events [45]. Given that proactive software process management may be conferring a competitive advantage on some companies, exploratory studies such as the one reported upon herein can aid our understanding in this area. In findings published previously, we analysed the extent of process improvement in small software companies [46], and identified a positive relationship between the amount of process improvement and organisational performance [47]. However, this earlier work examines neither situational change nor software process adaptive capability (SP-AC), nor does it explore the relationship between adaptive capability and organisational performance.

Process adaptation is supported in some of the existing software development frameworks and methods. For example, the highest level in capability maturity frameworks—*optimising*—supports continuous process adaptation. However, few organisations progress to this level (an observation that appears to be particularly true for smaller sized organisations [48], [49]). Certain agile software development approaches also support process adaption, including Scrum which contains a retrospective step that is designed to review the effectiveness of an implementation cycle (a *Sprint*).

While existing software development frameworks and methods enable process adaptation to some extent, no attempt to quantify process adaptation performance is undertaken. Just as we should adopt measurements and indicators as part of our strategy to evaluate software implementation and test, so too should we incorporate similar techniques when examining process adaptation. As Lord Kelvin is reputed to have stated: "*When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind*" (an observation that has previously been acknowledged to be pertinent for software development [50]). Of course, hard data alone can have a soft underbelly [51] and therefore complementary forms of information should also be considered—a concern that is discussed in more detail in Section 6.

The remainder of this paper is structured as follows: Section 2 presents the research design; Section 3 outlines the field study; Section 4 presents the data quantification; Section 5 describes the data analysis; Section 6 discusses the findings, while also highlighting limitations and areas for future research; the conclusion is in Section 7.

2 Research Design

This study adopted a mixed method research methodology, an approach that combines qualitative and quantitative methods to collect, analyse and present both types of data [52]. Mixed method research adopts the pragmatic view that a research methodology should be designed to suit the study context, whereby finding the closest match of theory and method is the paramount consideration when judging the legitimacy of the method. Our exploratory study context is concerned with software process adaptive capability and organisational performance, and examining these phenomena would require the collection of both quantitative data (for example, in the case of financial targets for companies) and qualitative data (for example, in relation to the opinion of a participant regarding the amount of change in an aspect of the software process). With both quantitative and qualitative data required in order to fully explore the research subject, a mixed method approach is therefore desirable.

An indication of the extent to which a software development organisation has adapted its software process relative to its changing situational context can be obtained by examining two phenomena: 1. The amount of software process change; and 2. The degree of change in the situational factors that affect the software development process. Since change occurs over time, it is further required that a time period be identified for the simultaneous examination of both software process change and situational change. Our exploratory study examined both of these phenomena along with business performance over 12 months—a time period commonly used for assessing business success though not without some limitations which we address in Section 6.

In order to reliably quantify the change in some phenomenon of interest, it is vitally important that a reference framework that comprehensively identifies the salient aspects of the phenomenon is adopted. Therefore, this research adopted ISO/IEC 12207 [53] as the underlying reference for software process, with the situational factors reference framework [14] being adopted in the case of situational change. Although the situational factors reference framework lacks the consensual validation that ISO/IEC 12207 offers, it is a recognised reference framework for the situational factors affecting the software process [54], [55], [56], [57], [58]; furthermore, it is, in the authors' opinion, the most comprehensive such framework available at the present time [59].

In the business literature, the term *success* is used interchangeably with the term *performance*, with both terms representing the achievement of something desired, planned or attempted [60]. However, some disagreement exists concerning what exactly is meant and understood by the term business performance [61], with many reasons existing for measuring performance [62], and many different performance measures that can be broadly classified into two groups: *financial* and *non-financial* [63].

Traditionally, business performance has been measured in financial terms [64], using measures such as return on sales and profit per unit production [65]. However, it is return on investment (ROI) that is conventionally regarded as the preeminent financial test of success [61], , [62], [63], [64], [65], [66], [67], and the pursuit of the highest rate of ROI is a primary consideration for owners and managers [66].

Financial return, however, is not the only important business success measure [60], and a short term focus on financial performance alone can inhibit long term growth [68]. Therefore, additional performance measures, for example of less tangible phenomena such as customer satisfaction, should also be assessed [69]. As a result of the limitations of purely financial performance measurement, there has been a "*shift from treating financial measures as the foundation for performance measurement to treating them as one among a broader set of measures*" [70]—and this has given rise to *multidimensional* performance measurement frameworks. Of the numerous multidimensional performance measurement frameworks that have been developed (incl. The *Macro Process Model* [71], the *Performance Pyramid* [72], and the *Performance Prism* [73]), it is the *Balanced Scorecard* (BSC) [74] approach that is the most popular and influential [75], [76].

Through its four perspectives (*financial*, *customer*, *internal business processes* and *learning and growth*), the BSC is considered to offer appropriate coverage of the dimensions of performance in a general business setting [77]. However, for the software development business sector, a broader approach to performance measurement is desirable, motivating the development of the Holistic Scorecard (HSC) which comprises of six perspectives: *financial*, *customer*, *business process*, *intellectual capital*, *employee* and *social* [78]. The HSC also provides an additional 16 critical success factors (CSFs) and sample key performance indicators (KPIs) for use when examining business success in software development organisations, and based on these strengths, it was selected for use in our exploratory study.

Having identified comprehensive reference frameworks for the phenomena of interest, the next step involved the development of corresponding survey instruments.

2.1 Software Process Change Survey Instrument

The amount of software process change in an organisation could be determined—if only indirectly—by conducting a differential analysis of the results of two process assessments conducted at two distinct points in time. However, with process assessments essentially concerned with the collection of data in relation to process capability, they are considered to be an inefficient approach to determining the amount of software process change in an organisation. Consequently, this study developed a new approach to examining process change that requires just a single engagement with a participating organisation.

Firstly, a robust technique for transforming the extensive text-based process descriptions in ISO/IEC 12207 into a comprehensive survey instrument was devised. This technique involved tagging and extracting the activities and tasks contained within ISO/IEC 12207. A baseline set of questions was then derived to permit an examination of the amount of process change, adopting the structure: *Has there been any modification to [an aspect of the software development process]?* A four point Likert scale was used to rate the reported amount of process change: *0 (none)*, *1 (minor)*, *2 (major)* and *(3) significant.*

The baseline set of questions were assembled into a draft survey instrument, which four members of the ISO/IEC 12207 working group (incl. current and former ISO/IEC 12207 editors) formally and independently reviewed. The reviewers, who had collectively accumulated in excess of 45 years of academic experience and more than 65 years of industrial software development experience, were asked a series of focused questions: (1) How well the survey instrument represented the ISO/IEC 12207 standard? (2) How effective the survey instrument would be for examining software process change? and (3) How appropriate the survey instrument was for SMEs (Small- to Medium-sized Enterprises; the target study domain).

Feedback from the reviewers indicated that the survey instrument preserved the structure and components of ISO/IEC 12207, while the expert reviewers recommended that the system and software contexts in ISO/IEC 12207 would probably not be applicable to SMEs. The draft survey instrument was revised accordingly, for example, the System Requirements Analysis and Software Requirements Analysis processes were merged into one line of inquiry. This rendered a set of 63 individual questions that covered the full spectrum of process concerns in ISO/IEC 12207, which were then subject to an industrial Pilot.

The Pilot, which confirmed the time required to discharge the survey instrument, did not generate any additional areas for investigation—however, a closing question regarding survey instrument coverage was retained in the instrument. The Pilot participant did stress that it was important to re-emphasise throughout the survey that the inquiry was concerned with process change over the past 12 months only, a recommendation that was applied.

While the entire survey instrument is too large to present within this paper, the following three sample software change questions (SWQs) serve as examples:

- *SWQ1: Regarding Software Construction, has there been any modification to the approach to developing and documenting software units and databases, including programming languages and build procedures/tools?*
- *SWQ2: Regarding Software Configuration Management, has there been any modification to the approach to performing change request (CR) activities which may include: the identification and recording of CRs; the analysis and evaluation of changes; the approval/disapproval of CRs; the implementation, verification and release of modified software?*
- *SWQ3: Regarding Project Planning, has there been any modification to the approach to project initiation, which may include the setting of project objectives, the establishment of feasibility (e.g., resources), and establishing the achievability of timescales?*

2.2 Situational Change Survey Instrument

A survey instrument was systematically derived from the situational factors reference framework, ensuring that at a minimum each of the 44 individual factors in the framework were addressed via individual questions in the survey instrument. As with the software process change survey instrument, a baseline set of questions was initially established, again using the basic structure: *Has there been any modification to [an aspect of the situation that can affect the software development process]?* The four point Likert rating scale adopted in the software process change examination was reused.

Since the creators of the situational factors reference framework were also responsible for the development of the associated situational change survey instrument, it was not possible to enter into the type of independent survey instrument validation that was undertaken for the ISO/IEC 12207-based software process change survey instrument. Rather, the validation of the situational change survey instrument employed two central tactics. First, and as noted in the previous paragraph, all of the essential components of the underlying situational factors reference framework were explicitly carried forward into the survey instrument, hence ensuring that the scope and content of the framework were preserved in the situational change survey instrument. Second, a Pilot industrial deployment of the situational change survey instrument was undertaken. In common with the feedback received for the software process change survey instrument Pilot, the participant recommended that the interviewer reiterate throughout the survey discharge that the period under investigation was the previous 12 months. The Pilot participant noted that with which a total of 49 individual questions, the scope of the survey instrument was broad and did not identify any additional situational factors.

While the entire survey instrument is too large to present herein, the following three sample situational change questions (SCQs) serve as examples:

- *SCQ1: Regarding the characteristics of requirements, have there been any modifications to the changeability of the requirements?*
- *SCQ2: Regarding the application(s)/product(s) under development, has there been any modification to the required performance of the application(s)/product(s)—including reliability and real time performance of both hardware and software components?*
- *SCQ3: Regarding operations, have there been any modifications to the volume or profile of end users?*

2.3 Business Success Survey Instrument

Scorecard-based approaches are beneficial for SMEs when implemented in a light-weight fashion that supports the definition and measurement of strategic business goals [79]. Therefore, we developed a HSC-based survey instrument to enable the task of measuring the extent of achievement of business goals (or business success) [80].

An initial baseline of questions was developed from the HSC, adopting the form “*For the forthcoming year, list any objectives that exist in relation to [a business objective]*”. Since small organisations are not necessarily explicit in the definition of business objectives, and to mitigate the risk of erroneous or biased recollections in relation to the achievement of business goals, the business success investigation consisted of two phases. The initial phase identified the prioritised business goals for the forthcoming year, using the rating scheme: *0 (No objective exists)* up to *4 (A high priority objective exists, with an explicit target)*. At the end of the year under investigation, the participating organisations were revisited in order to evaluate the achievement of the prioritised goals using the rating scheme: *0 (Not achieved to any extent)* up to *3 (Totally achieved)*. An industrial Pilot of the business success survey instrument resulted in a number of modifications. Firstly, the initial use of two separate questions for employee expertise and employee competence was merged into a single question. Secondly, headcount goals which were initially absent were added to the survey instrument. Following the Pilot phase, the business success survey instrument comprised of 51 individual questions.

The entire survey instrument is too large to present herein, however, the following three sample business objectives questions serve as examples:

- *BOQ1: For the forthcoming year, what is the target for the number of new client acquisitions?*
- *BOQ2: For the forthcoming year, list any employee objectives that exist in relation to employee qualifications?*
- *BOQ3: For the forthcoming year, list any objectives that exist in relation to patents, copyrights, branding and trade secrets?*

In our exploratory study, we applied the survey instruments identified above to examine the relationship between software process adaptive capability and the extent of business success in SMEs, testing the following hypothesis:

H1: To maximise business success in software SMEs, software process change should be in proportion to the degree of situational change.

3 Field Study

Over a 16 month period and adopting person-to-person interviewing, the central phenomena of interest were recorded in 15 SMEs.

3.1 Timing of Interviews

A total of four separate engagements were required within each of the 15 companies: one each for software process change and situational change, and a further two for business success (as outlined in Section 2.3). In Phase 1, the participating organisations were engaged so as to: (1) Identify their business objectives for the forthcoming year. In Phase 2, each organisation was revisited to examine: (2) The extent to which their business objectives were achieved; (3) The amount of software process change undertaken during the preceding 12 months; and (4) The degree of change to the software process situational factors over the preceding 12 months. The average total interview time per company was ~5.5 hours, giving a total interview time in the region of 83 hours.

3.2 Participating Organisations

Three of the companies had fewer than 10 staff. Another three companies had between 10 and 19 staff, and the remaining nine companies had between 20 and 120 staff. None of participating organisations had an annual turnover exceeding €50 million and/or an annual balance sheet total exceeding €43 million, thus meeting the European Commission’s SME definition [81]. The majority of the organisations retained their head office in the Republic of Ireland (RoI), however, two of the companies were predominately based outside the RoI, with a further four organisations retaining either development or operational centres internationally. The participating companies operated within diverse sectors. Four of the organisations developed web-based software, with another four organisations developing software for the telecommunications domain. The remaining seven organisations operated in a variety of domains, including, content management, data mediation, and embedded software.

3.3 Participating Individuals

The number of persons interviewed varied from organisation to organisation. In the smaller companies, just one person (often the owner/founder) tended to be aware of the business objectives, with that same person often doubling as development manager. In these companies, it was

TABLE 1
Participating Organisations and Personnel

Company Pseudonym	Business Success	SP Change	Situational Change
Silverback	CTO	CTO	CTO
Grenoble	CEO	EM	CEO, EM
Mega	MD	MD	MD
Cameron	MD	DM	MD, DM
Colleran	CEO	CEO	CEO
Lakes	MD	CTO	MD, CTO
United	MD	MD	MD
Watch	DF, CTO, DE	DE	DF, DE
BocaJ	MD	MD	MD
Tribal	DE	DE	DE
Dynamic	DE	DE	DE
Michelin	DE	DM	DE, DM
LordHenry	DE	DE	DE
When	COO	COO	COO
Oryx	COO	DM	COO, DM

typically the case that just that person was interviewed. In other cases, a number of persons participated in the interviews. The initial discussions with each of the organisations aimed to identify, among other things, the appropriate participants, who shared a range of job titles: Chief Executive Officer (CEO), Chief Operating Officer (COO), Managing Director (MD), Director of Finance (DF), Director of Engineering (DE), Chief Technical Officer (CTO), Engineering Manager (EM), and Development Manager (DM). A complete listing of the study participants—by role and company pseudonym—is provided in Table 1.

3.4 Collected Data

A considerable volume of data was collected in the study—too great to present in its totality in this paper. In order to provide some sample information on the collected data, Table 2 contains a sample of the data collected in relation to the software process change, situational change and business objectives questions presented earlier in Section 2.

It should also be noted that additional data was collected in the study—by way of interviewee comments in terms of the reported modifications or business objectives. A broader analysis of this additional information is outside the scope of this paper, however, some additional information may be found in [46], [47], [80] and [82].

3.5 Data Confidentiality Considerations

In order to support the elicitation of data and to promote trust, the participating organisations were assured that their data would be treated in confidence. Each of the companies was allotted a pseudonym and individuals were identifiable only by job title. A number of additional steps were taken to further support confidentiality, for example: interview recordings and transcriptions were encrypted, and a bilateral non-disclosure agreement (NDA) was drafted for use with the organisations.

4 Data Quantification

Using the data collected from the survey instruments, indicators were developed for software process change, situational change, and business success. It should be noted that for the most part, these indicators take the form of basic summations, which is in effect a simplistic view of the phenomena under study. Being simplistic has the benefit of being reasonably easy to understand and apply, but also suffers from the limitation that simple summation offers only an approximation to the phenomenon of interest as opposed a more precise measurement. Therefore, although expressed as mathematical formulae, the indicators are summations of subjective data which inherently lack mathematical objectivity. A further indicator introduced in this study relates to software process adaptive capability which attempts to capture the relative relationship between the amount of software process change and the amount of change to software development situational factors.

4.1 Quantifying Software Process Change

As presented in Section 2.1, software process change is examined using an ISO/IEC 12207-based survey instrument. This survey instrument was deployed to the participating companies, with participants quantifying individual modifications to the software process according to the Likert scale: *0 (none)*, *1 (minor)*, *2 (major)* and *(3) significant*. In order to quantify the total amount of software process change in an organisation, the individual recorded instances of process modification are summated, rendering the total software process change:

$$SP\,Change = \sum_{i=1}^{N} ModificationValue(\mathrm{i}),$$

TABLE 2
Sample of Data Collected

Question (Ref. to Section 2)	Company Pseudonym														
	Silverback	Michelin	Grenoble	When	Tribal	BocaJ	Watch	LordHenry	Lakes	Colleran	Cameron	Mega	United	Oryx	Dynamic
SWQ1	0	0	1	2	3	3	2	1	2	2	3	1	0	2	0
SWQ2	2	3	0	3	0	0	2	2	1	0	3	0	3	0	3
SWQ3	2	0	1	0	0	0	0	2	3	1	3	0	0	1	3
SCQ1	0	3	0	0	0	0	1	2	2	2	1	0	2	0	0
SCQ2	3	2	2	1	2	2	0	0	3	0	2	2	3	2	2
SCQ3	3	2	1	0	2	2	3	3	0	3	3	2	2	2	3
BOQ1	4	3	4	4	4	4	4	0	4	4	4	0	4	4	3
BOQ2	0	1	0	1	1	0	0	2	0	0	0	1	1	1	0
BOQ3	0	1	2	2	0	0	0	1	0	2	2	0	0	0	2

where *i* is a software process change question and *N* is the total number of process change questions in the survey instrument (as outlined in Section 2.1, there are 63 software process change questions).

4.2 Quantifying Situational Change

As presented in Section 2.2, situational change is examined using a survey instrument that is based on the software development situational factors framework [14]. This survey instrument was deployed to the participating companies, with participants quantifying individual modifications to the situational factors according to the Likert scale: *0 (none), 1 (minor), 2 (major)* and *(3) significant*. In order to quantify the total amount of situational change in an organisation, the individual recorded instances of change to situational factors are summated, rendering the total situational change:

$$SituationalChange = \sum_{j=1}^{M} ModificationValue(\mathrm{j}),$$

where *j* is a situational change question and *M* is the total number of situational change questions in the survey instrument (as outlined in Section 2.2, there are 49 situational change questions).

4.3 Quantifying Business Success

When making determinations in relation to the extent of business success, two pieces of information were considered: (1) the strength of the business objectives; and (2) the extent of achievement of the objectives. In order support a robust examination, two different business success quantifications were developed.

Firstly, a basic quantification of business success was employed whereby the overall success rating for an organisation was increased each time an objective was achieved to any extent, with the achievement of higher rated objectives having a greater proportional impact on the quantification. For example, the total achievement of a high priority objective with an explicit target will increase the overall success by 4 (4 [value for a high priority objective with an explicit target] × (3 [actual achievement value] / 3 [maximum possible achievement value])). In contrast, the total achievement of a low priority objective with no explicit target will increase the overall success by 1 (1 [value for a low priority objective with no explicit target] ×(3 [actual achievement value] / 3 [maximum possible achievement value])). The summing formula for quantifying the total basic success score for an organisation is as follows:

$$BasicBusinessSuccess = \sum_{k=1}^{O} ObjectiveRating(k) * \frac{ObjectiveAchievement(k)}{MaxAchievementValue},$$

where *k* is an individual business objective and *O* is the total number of business objectives questions in the survey instrument. As outlined in Section 2.3, there are 51 business objectives questions, with the stated *ObjectiveRating* ranging from 0-4 and the *ObjectiveAchievement* ranging from 0-3 (hence, *MaxAchievementValue* is 3).

Since the basic business success interpretation does not have a mechanism for lowering the overall success score in instances of failure to achieve stated objectives, it is possible for an organisation with a large number of objectives (some of which are not achieved) to appear more successful than another organisation with fewer stated objectives (all of which are totally achieved). This could be problematic as some companies may list relatively large numbers of objectives, with others listing relatively few. To address this issue, the second approach to quantifying business success involves the introduction of a weighted negative marking (WNM) scheme.

TABLE 3
WNM Business Success Scoring Scheme

	Objective Rating				
	No Objective (0)	Low Priority (1)	Low Priority Explicit (2)	High Priority (3)	High Priority Explicit (4)
Achievement					
Not achieved (0)	0	−1	−2	−3	−4
Partially achieved (1)	0	−0.67	−1.33	−2	−2.67
Mostly achieved (2)	0	0.67	1.33	2	2.67
Totally achieved (3)	0	1	2	3	4

Under the second approach, a company still receives a weighted score for each objective that is achieved to some extent; however, unlike the basic interpretation, an organisation receives a weighted negative score for each objective that is not achieved or only partially achieved. This type of WNM acts as a counter-balance; in cases where there are large numbers of objectives which are not achieved (or only partially achieved), the total score will take account of the extent to which stated objectives were not achieved. The essential benefit, therefore, of the WNM interpretation is that it enables the punitive measure of lowering overall success scores if there are objectives that are not achieved or only partially achieved. Furthermore, the weighting scheme is designed such that individual weights are related to the objective rating.

Applying the WNM scheme outlined in Table 3, the non-achievement of a low priority objective with no specific target will result in a—1 being applied to the total business success score, whereas the non-achievement of a high priority objective with an explicit target will result in a—4 being applied to the WNM business success score. For "mostly achieved" and "partially achieved" objectives, these weightings are applied in terms of "mostly achieved" tending to be close to "totally achieved", and "partially achieved" tending to be close to "not achieved". This has the effect of rewarding mostly achieved objectives generously, while instances of partial achievement are treated conversely.

With the participating organisations reporting varying numbers of objectives in the first instance, WNM ensures that a company with many objectives cannot score relatively highly unless they are achieving relatively well in most or all objectives. By quantifying business success using WNM, the resultant business success score offers the potential to provide a fairer and more accurate representation of the relative business success of the participating organisations—and it is the dimension of relative success that is of primary importance to this study. However, the basic business success score is

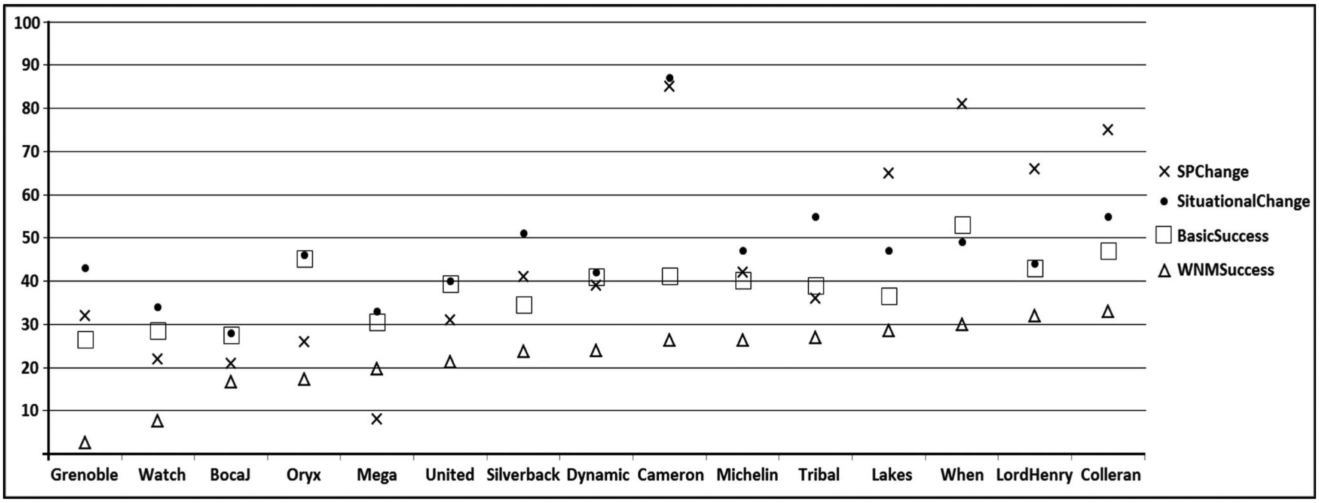

Fig. 1. Recorded software process change, situational change, basic business success and WNM success.

merited in that it generates a non-adjusted score for the achievement of stated objectives, and therefore, it is also considered to provide a valid indication of business success.

Using the WNM scheme presented in Table 3, the WNM business success score is quantified as follows:

$$WNMBusinessSuccess = \sum_{k=1}^{O} ModificationValue(\mathrm{k}).$$

As with the Basic Business Success score, *k* is an individual business objective question and *O* is the total number of business objectives questions in the survey instrument. The *ModificationValue* is determined based on Table 3.

4.4 Quantifying Software Process Adaptive Capability

The key purpose of the SP-AC indicator is to provide insight regarding the performance of an organisation vis-à-vis software process adaptation—in a general sense. As such, it is an indicator that could potentially be used as a complement to the existing set of performance indicators adopted in organisations (e.g., profit, ROI, customer satisfaction). In order to calculate the SP-AC[1], both *SPChange* and *SituationalChange* are required, as follows:

$$SPAC = \left(\frac{SPChange}{MaxSPChange}\right) \Big/ \left(\frac{SituationalChange}{MaxSituationalChange}\right).$$

5 Data Analysis

Although the primary purpose of this paper is to examine the relationship between software process adaptive capability and business success, the salient details regarding the data collected for software process change, situational change and business success are also presented here.

5.1 Software Process Change Data

In general, a considerable degree of variation was evident in the data collected across the field study. For example, the organisation Mega reported the lowest amount of software process change, with two other organisations, When and Cameron reporting over 10 times as much process change. Although a broad spectrum of process change was reported in the study, certain areas tended to receive more attention than others. The treatment of requirements was one such area, where the general trend was towards greater control, including requirements sign-off, and definition of requirements. Other common areas for process change included the provision of IT infrastructure for development and operations, and the use of partnering to deliver solutions to the market—the comprehensive nature of ISO/IEC 12207 enabled inquiries in areas such as these that might have fallen outside the scope of other reference frameworks. Significant additional details regarding software process change as reported in our study may be found in [46] and the quantifications for software process change for each company are depicted in Fig. 1.

5.2 Situational Change Data

Variation also presented in the reported situational change, most notably in staff headcount where all fifteen companies reported change, with eleven reporting increases. Nine of these eleven companies reported increases of 25 percent and greater, while of those companies reporting decreases in headcount, two reported a reduction in staffing of upwards on 40 percent. This staffing statistic alone would likely present a challenge to those responsible for managing the software process.

The number and profile of end users of software products was also an area for significant situational change, with

[1] Note: An SP-AC indicator is calculated for each organisation in the study group. *MaxSPChange* is the maximum possible amount of software process adaptation that could be calculated using the techniques adopted in this research. This is calculated by taking the number of individual questions in the software process change survey instrument (63) and multiplying it by the maximum possible amount of change reported in each question (3-significant), totalling 189 (refer to Section 2.1 for supporting details). *MaxSituationalChange* is the maximum possible amount of situational change that could be calculated using the techniques adopted in this research. This is calculated by taking the number of individual questions in the situational change survey instrument (49) and multiplying it by the maximum possible amount of change reported in each question (3-significant), totalling 147 (refer to Section 2.2 for supporting details).

11 of the companies reporting increases in the volume of users or transactions processed. Thirteen of the companies reported increases in the knowledge of technology, ranging from support for new operating systems, to integrated development environment usage and compilers. Twelve of the participating companies reported an increase in the required performance of their products, with 10 companies reporting increases in the size of their products, including increased code bases as well as increased data storage requirements. Just one of the companies reported a (small) decrease in code base, the result of intensive refactoring. There was also demand—in a total of nine companies—for improving the ease of use and installation of products. The quantifications for situational change for each company are presented in Fig. 1.

5.3 Business Success Data

In terms of setting business objectives, the single most important reported goal related to revenue, with all of the companies having explicit, high priority targets. Profit too was an important consideration, with all but three of the companies again having explicit, high priority targets, and only one company reporting that profit was a low priority. All of the companies expressed a desire to extend product offerings though there was varying ambition in this respect and many were unable to identify the exact extensions in advance, indicating that they would be client-led. All but two of the companies had objectives in relation to new client acquisitions, though these tended to be lower in priority than goals in relation to gaining repeat business from existing clients (where eight of the participating companies had high priority objectives). Objectives in relation to business process management were also reported, with eight of the participating companies having high priority objectives in this area.

The extent of achievement of business objectives, as collected in the second phase of the inquiry, demonstrated that the participating companies tended to struggle to fully achieve their highest priority objectives. Some moderate success was reported in terms of securing the desired repeat business from existing clients and in extending the product offering. However, the companies reported that they were less successful again in achieving revenue and profit targets, where five of the participating companies reported low or complete non-achievement of revenue objectives. Five of the companies also reported low or non-achievement of business process management objectives. Of the ten companies with objectives in relation to obtaining aids and subsidies, or tax exemptions, from government (which was a low priority objective in all cases), all ten were successful to some extent and six were mostly successful. Overall, this data suggests that smaller software development companies may be more successful at gaining repeat business from existing clients and deriving financial assistance from the State than they are at winning entirely new customers or at meeting their own financial targets. Additional details regarding business objectives and their achievement as observed in our study may be found in [82] and the quantifications for business success (both Basic and WNM) for each company are presented in Fig. 1.

5.4 Software Process Adaptive Capability

An initial analysis of the data demonstrated that higher levels of the business success quantifications tended to present alongside higher levels of both software process change and situational change. This may indicate that higher levels of business success are resulting in increased levels of software process change and situational change or it could also suggest that higher levels of software process adaptive capability are supporting organisational performance. However, it may be more likely again that what we are witnessing here is an amethodical relationship [83] between complex phenomena, whereby a change to one phenomenon results in a change another phenomenon, which may in turn result in change to a further phenomenon (or a change to the phenomenon that kicked off the process); in effect, a state of flux exists in the relationship between these complex phenomena.

Adaptive capability relates to an organisation's ability to identify, understand and address changes in its operating environment—and in so doing, support successful business outcomes. In order to explore adaptive capability in software development companies, this research has simultaneously examined both software process change and situational change. These two summations can be jointly considered to provide an indicator of software process adaptive capability in an organisation via the SP-AC indicator, which is in effect a simple ratio of the amount of software process change to the amount of change to the situational factors that are known to affect the software process. To facilitate an examination of the relationship between SP-AC and the basic and WNM business success quantifications, statistical data correlation was employed.

Several different correlation techniques were adopted in order to reduce the dependency on any single correlation method. Furthermore, the correlation techniques satisfied the demands of both ordinal and measurement data. As presented in Section 4, much of the data collected in this study is of an ordinal nature and therefore, it may be best suited to ranked order correlation using techniques such as the Spearman and Kendall coefficients [84]. However, Pearson product moment correlations can be considered to be permissible and useful for ordinal data, especially where the underlying data is of a broadly normal distribution [85] (and may offer higher precision than ranked order correlation [86]). Using R [87], the various correlations were calculated; with the results demonstrating that SP-AC is positively correlated with the two business success quantifications adopted in this research.

Examining the correlation coefficients for SP-AC and business success (both basic and WNM), it is found that the coefficients range from 0.49 to 0.79 (refer to Table 4). When taking the sample size, the correlation coefficients and the p-values into consideration, it can be declared that the correlation coefficients are statistically significant. Therefore, the analysis of the data collected in this study would appear to support hypothesis H1. In essence, the finding from our exploratory study is that where the total software process change is relatively greater than the total situational change, businesses are also reporting increased levels of business success. However, correlation should not be interpreted as causation and our quantifications exhibit a number of

TABLE 4
Spearman, Kendall and Pearson Correlation Coefficients

		SP-AC & Basic Business Success	SP-AC & WNM Business Success
Spearman	R	0.58	0.79
	p-value	0.03	0.0004[2]
Kendall	T	0.49	0.61
	p-value	0.011	0.0015[2]
Pearson	r	0.63	0.64
	p-value	0.01	0.01

[2] *Spearman & Kendall correlation p-values cannot necessarily be reliably computed where a variable has the same value for two separate cases, as is the case for the participating organisations* Cameron *and* Michelin *(both of which got a WNM Business Success score of 26.33).*

limitations, therefore, the following section presents an evaluation of the study data and findings.

6 Discussion

The exploratory study finding that there is a statistically significant positive correlation between SP-AC and business success is congruent with the fundamental proposition of evolutionary theory as it applies in the business and economics literature—that an ability to continually perceive change and adapt accordingly is an enabler for business success. While the observed correlation can be considered to be quite strong, the techniques introduced in this research for examining adaptive capability do exhibit a number of limitations which are important when interpreting the findings. There are discussed later in this section. First, however, we evaluate the potential meaning of the findings.

6.1 Study Findings

Perhaps the most striking feature of the data is that those organisations with greater SP-AC figures are also tending to report higher levels of business success. There are three core quantifications underlying this observation: software process change, situational change and business success. An obvious question arises as to what exactly is the nature of the relationship between these three phenomena—for which various possible explanations can be proposed. One explanation is that factors relating to business success are the catalyst for situational change and software process change. Furthermore, factors related to business success can be outside the scope of control of an organisation—for example, if a new competitor enters the marketplace with an equivalent product at a substantially lower price, it could prove difficult to develop a viable survival strategy. In such a scenario, it may be the case that only substantial, painful and potentially impractical process change (for example, outsourcing large aspects of the development effort to a geographic region with a lower cost base) could overcome this challenge.

A second possible explanation for the relationship between the phenomena is that software process change is supporting improved organisational performance. Software process change and business success (under both success interpretations) are positively correlated (ranging from 0.48 to 0.81). It could be further proposed that increased awareness of situational change is a prerequisite for effective software process change—such a suggestion elevates situational awareness to be of considerable importance. Indeed, the statistical correlations for situational change and business success (under both success interpretations) are also positively correlated in our exploratory study (ranging from 0.39 to 0.66). However, just being aware that something is changing is potentially of little value to a business unless it also possesses the ability to respond with appropriate measures.

Rather than viewing the possible explanations for the relationships presented above as being mutually exclusive, it is possible that the various interrelationships are amethodical [83] in nature—and given the complexity of the phenomena under investigation in this study, it is difficult to argue in favour of an alternative explanation. An amethodical relationship is one in which a change to a phenomenon can give rise to a change in related phenomena. In effect, something that can be considered similar to symbiosis is at play, whereby multiple phenomena constitute a system, influencing each other in a series of reflexive type inter-relationships.

The correlations established in our study suggest that the phenomena may interact in an amethodical system. For example, increases in business success are positively associated with increases in software process change. Similarly, increases in situational change are positively associated with increases in business success. To further support this view, an additional correlation was performed on software process change and situational change. Again, there is a positive correlation (ranging from 0.63 to 0.79). The three phenomena are therefore presenting as being all positively associated with each other—and this observation can be considered to be aligned with the core philosophy of adaptation. For example, if a business is more successful, then it may experience growth which may require process change. Similarly, if a business is more perceptive in identifying situational change, this may result in process adaptation, which in turn results in business success.

It is therefore also the case that the absolute values recorded for the various phenomena may themselves be important quantifications for consideration. For example, if an organisation reports little process change over a protracted period, one can reasonably be apprehensive—since a company is not likely to have a perfectly adjusted set of processes on a continual basis given the rate of environmental change in software development settings (as demonstrated through the situational change recorded in our study). It should equally be a potential concern if an organisation reports low levels of situational change, as this may indicate a shortcoming in terms of perceiving or creating change (something that is not desirable from an evolutionary perspective).

Considered collectively, the various correlations that are evident on our exploratory study data would appear to support the theoretical role of adaptive capability in supporting business success. The implication for those responsible for designing and applying software process frameworks and methods may be that they should provide strong and versatile mechanisms for process adaptation.

6.2 Study Limitations

While our exploratory study was carefully designed and discharged, it does exhibit a number of noteworthy limitations that should be taken into account when interpreting the study findings.

6.2.1 Generalizability of Findings

Within the study component of the research, the sample size can be considered to be small. A total of just 15 companies participated across the full lifetime of the study. When compared to the total population of software development organisations globally, the sample is very limited. Indeed, it could be claimed that the sample is so small that the findings could be the outcome of chance. However, and as elaborated upon in the following paragraphs, there are sound reasons to suggest that this is not the case.

As noted in the previous section, increased absolute values for the three primary phenomena under investigation are positively associated in the study data—and this is consistent with certain reasonable expectations: for example, companies reporting little software process change may not be actively managing their software process which is not likely to be a desirable situation. In this respect, it should be noted that out of the five least successful companies (under both success interpretations—which themselves are quite different in nature), four of the companies present with the lowest absolute values for software process change (refer to Fig. 1).

A further indication that the results are not the product of mere chance can be observed in the most successful company profiles. The three most successful companies (under both success interpretations) report among the highest levels of software process change in the study group. Furthermore, each of these three highest performers has a profile wherein the amount of software process change is proportionally greater than the corresponding reported situational change (with such an observation absent outside the top few performing companies).

One final indication that the results are not the product of chance can be found in the profile of the organisation *Cameron*. This company reports the single highest amount of software process change—yet it is not the highest performing company (it is the fifth highest performing company under the basic interpretation and seventh highest under the WNM interpretation). However, *Cameron* also reports the highest level of situational change across the entire study group—and this may be necessitating such high levels of software process change. This particular observation may highlight the important role of situational change when interpreting adaptive capability in companies. It should nonetheless be noted that the field study is of an exploratory nature, and although the outcomes do support the theory under exploration, the sample size is very small. However, it should also be noted that certain practical limitations exist when conducting studies of this type, including the significant time requirements imposed on the researchers and participants (for example, our study required >80 hours of interview time). Additionally, obtaining access to organisations for studies of this type can prove to be challenging.

6.2.2 Study Group and Techniques

The study is also limited in the scope of the organisations that participated—which are exclusively drawn from the SME sector. This raises a number of potential issues. Firstly, the survey instruments that were designed and discharged in this research are potentially not well suited to organisations that fall outside the strict scope of an SME. Secondly, the architecture of the inquiries conducted in this research may not easily scale to larger organisations where many additional individuals and projects might need to be engaged in order to get an organisational view of the phenomena of interest. Indeed, the discharge of the survey instruments adopted in this study requires expertise in a broad number of areas such as may, in the absence of training and education, render them impractical for general application in their present form.

It could further be the case that attempting to measure adaptive capability may not be the most effective way of enabling adaptive capability. Rather, the qualitative data elicited from our software process and situational change instruments may provide the level of detail required to enable adaptive capability. Finally, since the study addresses only the SME sector, no claims of similar behaviour can be made with respect to other types of organisations (e.g., large and very large sized companies). However, and despite the additional challenges associated with obtaining software process change and situational change information in larger organisations, the basic principles of adaptation should apply just as much in large organisations as they do in their smaller counterparts.

6.2.3 Completeness of Quantifications

The general approach to making determinations in relation to the phenomena of interest exhibits some limitations. Although all three phenomena are examined through the lens of comprehensive reference frameworks (ISO/IEC 12207, the Situational Factors Reference Framework, and the Holistic Scorecard), there remains the possibility that some aspects of the phenomena under examination have been overlooked. This may be particularly the case for the situational factors examination which is based upon a relatively new reference framework (which itself is the product of related research).

To mitigate the risk of overlooking some factors, closing questions were included in all survey instruments to allow participants to identify additional items. Furthermore, and as articulated in Section 2, the survey instruments were systematically developed and subjected to a combination of external expert peer review and piloting with industrial partners. Nonetheless, the survey instruments (and the reference frameworks upon which they are based) can only provide an approximation to the fullness of the phenomena of interest.

6.2.4 Participant and Quantification Subjectivity

A further limitation in the field study concerns the very nature of the data being collected. The survey instruments adopted by this research are to some extent concerned with eliciting the views of practitioners and business managers. As outlined in Section 2, the data took the form of a combination of ranked order scales and absolute values for various aspects of the phenomena under investigation. For example, a company would disclose absolute values for the targeted and actual number of new clients acquired in the study timeframe, while the interview participant(s) would express a view on the extent of change in some aspect of the software process or situational context. Although an accepted and useful approach to conducting research of this type, the latter form of reporting (i.e. the point of view of the participant(s)), is potentially a source of error in the recorded data.

Concerns related to subjectivity and bias are introduced in this type of inquiry—and are difficult to completely eliminate. After all, what one person reports as a minor change, another may report as a moderate or major change. As such, it is important to highlight that the quantifications employed in our analysis should not be interpreted in the same manner as purely objective data. For example, it is not the case that the quantifications adopted in this study are objective in the same way that a physicist might objectively measure the speed of a ball. In this respect, the quantifications adopted are employed as an exploratory measure to examine the three phenomena of interest in some quantifiable fashion, with a view to examining adaptive capability vis-à-vis the software process. We believe that the quantifications nonetheless have a value for the study—but this should not be confused with entirely objective measurement as adopted in other fields or in other research. To some extent, this is related to a general limitation that affects all subjective exploration where quantification is employed.

It is also important to highlight that the quantifications employed relate to the opinions of study participants. In some cases, this involved just a single person in an organisation, in other cases multiple participants from a single organisation were involved. The key point here is that it is *opinion* that is being measured and therefore, the reported amounts of process change and situational change are inherently linked—that is, we might to some extent expect a positive correlation from the analysis of these phenomena.

It should further be recognised that while attempts to aggregate data into indicators can bring some benefits in terms of visualisation, the very practice of aggregating has the effect of eroding some of the richness in the primary data, and therefore can reduce the potential utility of the data in informing business direction. This effect, which Henry Mintzberg refers to as "the soft underbelly of hard data" [51], can also manifest itself in other undesirable ways, it can for example be the case that hard data is unreliable, and even where reliability is intact, the hard data can simply arrive too late to address the problems that it may identify. This latter concern is of direct relevance to the quantifications produced by our study—which are only available at the end of the year under examination. In this respect, it could be the case that later work in this space might seek to shorten the timeframe for data collection and quantification so as to reduce the effect of delayed information.

To reduce the potential impact of subjectivity-related concerns, individual questions could be examined in more detail by the interviewer or the participant(s) could discuss the question in more detail. Additionally, the questions themselves had been carefully crafted in the first instance to reduce the possibility of misinterpretation. Furthermore, in cases where major or significant change was reported, it was generally accompanied by some explanation of the change which permitted the interviewer to further discuss the ranking. Had the participant(s) been required to describe each and every reported process or situational change in detail, the researchers could have conducted some cross-company normalisation in order to reduce the impact of subjectivity. Furthermore, had multiple individuals from each company participated, perhaps in the form of focus groups, the potential impact of subjectivity could have been reduced. However, these steps were not conducted arising from practical limitations—the accumulated interviewing time would have been overbearing for companies and researchers alike. Hence, the potential for subjectivity of participants remains a limitation.

6.2.5 Business Success Determination

In the case of business success, our exploratory study adopted the HSC as an extensive reference framework of the possible factors for consideration when assessing success in software development companies. We have also introduced a two-phase business success examination in order to reduce the effect of biased or false recollection; plus a question was retained in the survey instrument to permit the elicitation of success objectives outside the immediate scope of inquiry. We have furthermore developed two different and quite distinct techniques for quantifying business success. However, alternative interpretations of business success could affect our conclusions regarding the correlations presented herein.

The business success determination exhibits a further limitation in that it does not account for changing business success objectives throughout the period of investigation, when in practice business objectives may be subject to change. A further complication arises from the fact that an organisation could be successful even in the absence of stated objectives. Indeed, some influential contributors to the business field have suggested that it is possible to choke a company with strategic planning (including objective setting) leading to rigidity [51]. However, multidimensional performance measurement frameworks (as identified in Section 2) are routinely applied when measuring performance through strategic planning and our exploratory study follows this principle. Nonetheless, it should be stressed that this is just one view of business success, and one that is not without its limitations.

6.2.6 Exploratory Study Duration

An additional limitation stems from the fact that the period under investigation is just one year. Those experienced in process change will be aware that the impact of process changes on business success can take more than one year to fully materialise. As a result, some of the process changes recorded in the year under examination in our study may not carry major benefits until a future year. Similarly, some of the business success recorded could be the result of process improvements enacted in earlier years—and not those improvements implemented during the time frame of our study.

Our exploratory study evaluated software process adaptive capability over the course of a discrete year. However, adaptive capability may vary over time and therefore, a longer study timeframe would permit a more thorough analysis of the relationship between the key phenomena. One way that a company's adaptive capability might be significantly affected over time would be through changes to key company employees. However, in the case of our study, the key individuals that we worked with were all medium/long term fixtures in their respective companies, in situ at study outset and remaining there at study conclusion. It is also the case that software development companies often operate in fast paced environments, wherein it may be more likely that process changes will impact business success in a shorter time frame than more traditional businesses. Nonetheless, factors other than key employees could affect software process adaptive capability and consequently, the timeframe does introduce bounds to the interpretation of the data.

6.2.7 Data Quantification

A number of new quantifications were developed for the purpose of the exploratory study. In the case of *SPChange* and *SituationalChange*, simple sums of the modification to individual process activities and situational factors are employed—in effect applying equal importance to all process activities and all situational factors. While this serves the purpose of providing a barometer for these phenomena, it does not take account of the fact that certain process changes and certain situational changes will inevitably carry greater weight than others.

The primary reason for not weighting the summation components arises from the subjectivity that would be introduced by the researchers (effectively resulting in increased subjective interpretation of underlying subjective data). Gaining consensus on any applied weighting would prove challenging (if not impossible) in the broader community, hence it was decided to apply an equal weighting to all constituents. The result is a less-than-perfect view of the process change and situational change phenomena, but a perfect view of these complex phenomena is not possible under any subjective examination (and from a practical standpoint, only a subjective examination could be adopted to elicit the required information).

It should further be noted that the SP-AC indicator adopted in this research is lacking specificity. By design, the SP-AC does not consider the relationship between individual situational changes and individual process changes. Rather, the SP-AC is a basic ratio of the total software process change to the total situational change. The SP-AC is therefore a bellwether—it is an indicator that assists the general quantification of software process adaptive capability. As such, it is the case that relatively low SP-AC figures may not indicate that there is an issue with the adaptive capability of an organisation, but rather that there is potentially an issue. It should further be noted that there is a reported shortfall in the arsenal of tools available for software process evaluation and that it is "*crucial to push the event horizon of improvement measurement beyond the level of projects*" [88]. The various quantifications introduced in this research may prove useful in addressing the observed gap, however, additional validation steps are advised prior to widespread adoption.

At the present time, there is no other known way of capturing adaptive capability in software companies, and therefore, the SP-AC represents a new contribution to the field. It is not recommended to use the SP-AC in isolation but rather, as a complement to existing business performance metrics. It should also be noted that a complexity issue arises when trying to relate contextual factors with aspects of software development practice [89], such that it may not be advisable to attempt to seek greater specificity in examining adaptive capability; perhaps, therefore, an indicator such as SP-AC is the extent of the focus that we can apply when attempting to quantify software process adaptive capability.

6.3 Future Work

Future valuable work could concentrate on better understanding the relationship between specific situational factors and aspects of the software development process. It has been noted that adjustments to standard software processes are necessary to make them suitable for specific environments [90]. Such adjustments are sometimes referred to as process tailoring, an activity that involves "*adjusting the definitions and/or particularizing the terms of general description to derive a description applicable to an alternate (less general) environment*" [91]. It has further been asserted that "*the diversity of IT projects frustrates any direct attempt to systematize the processes used for their development. One size just won't fit all... All too often, deviation from a standard methodology is seen as an imperfection, as an unwelcome compromise (despite the fact it always happens!)*" [92]. Even in more contemporary approaches, such as agile software development, it has been observed that individual companies may take an *a la carte* approach to practice adoption [93].

Process adaption therefore may be required both to implement a standard approach in an organisation (or project) and to adapt a process to changing situational contexts (which can be considered to be unavoidable). Consequently, it may be beneficial for the software engineering community to have access to a knowledge base that could assist our understanding of software process to situational specific mappings. The development of such a resource would require substantial input from software process experts and would benefit from the incorporation of field data from companies regarding both phenomena. Software practitioners could apply such a resource to support their software process adaptive capability when addressing changing situations, hence producing greater process optimality. However and as already noted, it should be cautioned that considerable complexity exists in the interplay between situational contexts and software processes, such that the focus of attempts to qualify this relationship would require very careful consideration, and a clearly defined scope. The next small step might involve trying to identify the higher priority aspects of the software process and situational factors and to attempt to model their interaction.

7 Conclusion

The role of adaptive capability in supporting business success has long been established [94], [95], [96], and it has been suggested that deriving a competitive advantage is dependent on the ability to continually improve processes [97]. The perspective on the importance of process adaptation in response to changing situational stimuli being taken here has its fundamental roots in evolutionary theory in the natural sciences. Since the late 1970s, such evolutionary perspectives have had a strong and growing influence in the literature on business organisations and organisational economics, in particular in seeking to understand and explain their relative success in coping with rapidly changing environments (as evident for example in [94], [95], [96]). Therefore, the concept of adaptive capability being highlighted and examined in this work can be considered to be strongly aligned with this now well-established evolution tradition in the management and economics areas. It should further be noted that a closely related notion within this broader theoretical literature is the concept of “dynamic capability”, which is intuitively very appealing as a descriptor of the phenomenon that we are highlighting in this paper, but the fields of strategy and economics are still debating what precisely they intend by this term theoretically, so we have decided to remain with the more generic evolutionary term of “adaptive capability”.

Our longitudinal study of 15 companies found that software process adaptive capability and organisational performance are positively associated. Had our study involved a much greater number of companies, had it conducted more extensive inquiries within the participating organisations, and had it examined a broader timeframe, then we could make much stronger statements in relation to our findings. However, our study was of an exploratory nature and it exhibits a number of limitations that should not be casually overlooked. Perhaps paramount among the limitations is the subjectivity of study participants and related quantifications, and the fact that what is actually being measured is their opinion with respect to their changing circumstance, their software process and their business performance. However this limitation itself may highlight an important characteristic of more performant organisations—in that if business performance is informed (at some level) by process adaptation, and process adaptation is catalysed by a capacity to perceive change, then software process optimality and the potentially significant benefits that it can convey are constrained in the first instance by the very ability to recognise changes to the situation. Future examinations of this characteristic may offer a fruitful avenue for advancement in both research and practice.

Despite the limitations that we have identified, our study findings do open up the possibility that software process adaptive capability may be an important differentiator in the context of competitive advantage and organisational performance, and this has implications for researchers and practitioners alike. Those involved in the creation and evolution of software process models and methodologies may want to enhance their support for adaptive capability, those actively engaged in industrial software development may want to focus additional energy on their adaptability, and those involved in the research community may want to further explore this space.

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] SEI, CMMI for Development, Version 1.3. CMU/SEI-2006-TR-008. Pittsburgh, PA, USA: Softw. Eng. Ins., 2010.

[2] Information Technology—Process Assessment, Part 1 to Part 5. Geneva, Switzerland: International Organisation for Standardization, ISO/IEC, IS0/IEC 15504, 2005.

[3] Quality Management Systems—Requirements. Geneva, Switzerland: ISO, ISO, ISO 9001:2000, 2000.

[4] K. Schwaber and M. Beedle, *Agile Software Development with SCRUM*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[5] K. Beck and Extreme *Programming Explained: Embrace Change*. Reading, MA, USA: Addison-Wesley, 1999.

[6] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*. Upper Saddle River, NJ, USA: Addison-Wesley, 2003.

[7] M. Fowler and J. Highsmith, “The Agile Manifesto,” *Softw. Develop.*, vol. 9, no. 8, pp. 28–32, 2001.

[8] L. Constantine. (2012). Methodological agility [Online]. Available: http://www.ddj.com/ architect/184414743

[9] C. Jones. “Development Practices for Small Software Applications,” *CrossTalk, J. Defense Softw. Eng.*, vol. 21, no. 2, pp. 9–13, 2008.

[10] T. DeMarco and B. Boehm, “The agile methods fray,” *IEEE Comput.*, vol. 35, no. 6, pp. 90–92, Jun. 2002.

[11] P. Feiler and W. Humphrey, Software Process Development and Enactment: Concepts and Definitions. CMU/SEI-92-TR-004. Pittsburgh, PA, USA: Software Eng. Institute, Carnegie Mellon Univ., 1992.

[12] B. Boehm and R. Turner, *Balancing Agility and Discipline—A Guide for the Perplexed*. Boston, MA, USA: Pearson Education, 2003.

[13] K. Petersen and C. Wohlin, “Context in industrial software engineering research,” in *Proc. 3rd Int. Symp. Empirical Softw. Eng. Meas.*, 2009, pp. 401–404.

[14] P. Clarke and R. V. O’Connor, “The situational factors that affect the software development process: Towards a comprehensive reference framework,” *J. Inf. Softw. Technol.*, vol. 54, no. 5, pp. 433–447, 2012.

[15] J. Herbsleb, A. Carleton, J. Rozum, J. Siegel, and D. Zubrow, *Benefits of CMM-Based Software Process Improvement: Initial Results*. Pittsburgh, PA, USA: Software Eng. Inst., Carnegie Mellon Univ., 1994.

[16] J. Herbsleb and D. Goldenson, “A systematic survey of CMM experience and results,” in *Proc. 18th Int. Conf. Softw. Eng.*, Mar. 25–29, 1996, pp. 323–330.

[17] P. Lawlis, R. Flowe, and J. Thordahl, “A correlational study of the CMM and software development performance.,” *CrossTalk, J. Defense Softw. Eng.*, vol. 8, no. 9, pp. 21–25, 1995.

[18] K. Hyde and D. Wilson, “Intangible benefits of CMM-based software process improvement,” *Softw. Process: Improvement Practice*. vol. 9, no. 4, pp. 217–228, 2004.

[19] D. Gibson, D. Goldenson and K. Kost, “Performance results of CMMI-Based Process Improvement. CMU/SEI-2006-TR-004. Pittsburgh, PA, USA: Softw. Eng. Inst., Carnegie Mellon Univ., 2006.

[20] S. Cepeda, S. Garcia and J. Langhout, “Is CMMI useful and usable in small settings?,” *CrossTalk, J. Defense Softw. Eng.*, vol. 21 , no. 2, pp. 14–18, 2008.

[21] M. Serrano, C. Montes de Oca, and K. Cedillo, “An experience on implementing the CMMI in a small organisation using the team software process,” in *Proc. 1st Int. Res. Workshop Process Improvement Small Settings*, Oct. 19–20, 2005, pp. 81–92.

[22] M. C. Paulk, “Using the software CMM in small organizations,” in *Proc. Joint 1998 Pacific Northwest Softw. Quality Conf. 8th Int. Conf. Softw. Quality*, Oct. 13–14, 1998, pp. 350–361.

[23] H. K. N. Leung and T. C. F. Yuen, “A process framework for small projects,” *Softw. Process: Improvement Practice*, vol. 6, no. 2, pp. 67–83, 2001.

[24] A. C. Jost, “Small project survival among the CMMI Level 5 big processes,” *Crosstalk: J. Defense Softw. Eng.*, vol. 21, no. 2, pp. 22–26, 2008.

[25] A. Cater-Steel, "Process improvement in four small software companies," in *Proc. 13th Australian Softw. Eng. Conf.*, Aug. 26–28, 2001, p. 262.

[26] A. Anacleto, C. G. Von Wangenheim, C. F. Salviano, and R. Savi, "Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil," in *Proc. 4th Int. Conf. Process Assessment Improvement*, Apr. 27–29, 2004, pp. 33–37.

[27] M. Sanders, *The SPIRE Handbook. Better, Faster, Cheaper Software Development in Small Organisations*. DCU, Dublin, Ireland: Centre for Softw. Eng. Limited, 1998.

[28] C. G.Von Wangenheim, S. Weber, J. C. R. Hauck, and G. Trentin, "Experiences on establishing software processes in small companies," *Inf. Softw. Technol.*, vol. 48, no. 9, pp. 890–900, 2006.

[29] A. Cater-Steel and T. Rout, "SPI long-term benefits: Case studies of five small firms," in *Software Process Improvement for Small and Medium Enterprises—Techniques and Case Studies*, H. Oktaba Ed. Hershey, PA, USA: IGI Global, 2008.

[30] A. I. F. Ferreira, G. Santos, AND R. Cerqueira, "Applying ISO 9001:2000, MPS.BR and CMMI to achieve software process maturity: BL informatica's pathway," in *Proc. 29th Int. Conf. Softw. Eng.*, May 20–26, 2007, pp. 642–651.

[31] M. Biro, J. Ivanyos, and R. Messnarz, "Pioneering process improvement experiment in Hungary," *Softw. Process: Improvement Practice*, vol. 5, no. 4, pp. 213–229, 2000.

[32] L. Scott, L. Carvalho, R. Jeffery, J. D'Ambra, and U. Becker-Kornstaedt, "Understanding the use of an electronic process guide," *Inf. Softw. Technol.*, vol. 44, no. 10, pp. 601–616, 2002.

[33] J. G. Da Silva and P. R. Da Cunha, "Reconciling the irreconcilable: A software development approach that combines agile with formal," in *Proc. 39th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 4–7, 2006, p. 216b.

[34] P. Abrahamsson, O. Salo, J. Ronkainen, Agile Software Development Methods—Review and Analysis. VTT Publications Number 478. Finland: VTT Technical Research Centre of Finland, 2002.

[35] B. Fitzgerald, G. Hartnett, and K. Conboy, "Customising agile methods to software practices at Intel Shannon," *Eur. J. Inf. Syst.*, vol. 15 no. 2, pp. 200–213, 2006.

[36] D. J. Reifer, "How good are agile methods?," *IEEE Softw.*, vol. 19, no. 4, pp. 16–18, Jul./Aug. 2002.

[37] V. Kettunen, J. Kasurinen, O. Taipale, and K. Smolander, "A study on agility and testing processes in software organizations," in *Proc. 19th Int. Symp. Softw. Testing Anal.*, Jun. 12–16, 2010, pp. 231–240.

[38] E. Whitworth and R. Biddle, "The social nature of agile teams," in *Proc. AGILE Conf.*, Aug. 13–17, 2007, 2007, pp. 26–36.

[39] S. Syed-Abdullah, M. Holcombe, and M. Gheorge, "The impact of an agile methodology on the well being of development teams," *Empirical Softw. Eng.*,vol. 11, no. 1, pp. 143–167, 2006.

[40] L. A. Williams and R. R. Kessler, "The effects of 'pair-pressure' and 'pair learning' on software engineering education," in *Proc. 13th Conf. Softw. Eng. Edu. Training*, Mar. 6–8, 2000, p. 69.

[41] T. Van de Grift, "Coupling pair programming and writing: Learning about students' perceptions and processes," *ACM SIGCSE Bull.*, vol. 36, no. 1, pp. 2–6, 2004.

[42] R. Van Solingen, "Measuring the ROI of software process improvement," *IEEE Softw.*. vol. 21, no. 3, pp. 32–38, May/Jun. 2004.

[43] R. Lal. (2011). Strategic factors in agile software development method adaptation: A study of market-driven organisations. Ph.D. dissertation, AUT University, Auckland, New Zealand [Online]. Available: http://mro.massey.ac.nz/bitstream/handle/10179/2496/02_whole.pdf

[44] N. Baddoo and T. Hall, "De-motivators for software process improvement: an analysis of practitioners' views," *J. Syst. Softw.*, vol. 66, no. 1, pp. 23–33, 2003.

[45] G. Coleman and R. O'Connor, "Investigating software process in practice: A grounded theory perspective," *J. Syst. Softw.*, vol. 81 no. 5, pp. 772–784, 2008.

[46] P. Clarke and R. V. O'Connor, "An empirical examination of the extent of software process improvement in software SMEs," *J. Softw.: Evolution Process*, vol. 25, no. 9, pp. 981–998, 2013.

[47] P. Clarke and R. V. O'Connor, "The influence of SPI on business success in software SMEs: An empirical study," *J. Syst. Softw.*, vol. 85, no. 10, pp. 2356–2367, 2012.

[48] M. Niazi and M. A. Babar. "Identifying high perceived value practices of CMMI level 2: An empirical study," *Inf. Softw. Technol.*, vol. 51, no. 8, pp. 1231–1243, 2009.

[49] F. J. Pino, F. Garcia, and M. Piattini. "Software process improvement in small and medium software enterprises: A systematic review," *Softw. Quality J.*, vol. 16, no. 2, pp. 237–261, 2008.

[50] T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimates*. Englewood Cliffs, NJ, USA: Yourdon Press, 1982.

[51] H. Mintzberg, B. Ahlstrand, and J. Lampel, "*Strategy Safari: A Guided Tour Through the Wilds of Strategic Management*. New York, NY, USA: Free Press, 1988.

[52] S. C. Petter and M. J. Gallivan, "Toward a framework for classifying and guiding mixed method research in information systems," in *Proc. 37th Hawaii Int. Conf. Syst. Sci.*, Jan. 5–8, 2004, pp. 1–10.

[53] Systems and Software Engineering—Software Life Cycle Processes. ISO/IEC, ISO/IEC 12207–2008—Geneva, Switzerland: ISO, 2008.

[54] T. Dyba, "Contextualizing empirical evidence," *IEEE Softw.*, vol. 30, no. 1, pp. 81–83, Jan./Feb. 2013.

[55] T. Dyba, D. I. K. Sjoberg, and D. S. Cruzes, "What works for whom, where, when, and why?: On the role of context in empirical software engineering," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, 2012, pp. 19–28.

[56] K. Smolander and T. Päivärinta, "Forming theories of practices for software engineering," in *Proc. 2nd SEMAT Workshop General Theory Softw. Eng. (Co-Located Int. Conf. Softw. Eng.)*, 2013, pp. 27–34.

[57] S. A. Fricker, M. Persson, and M. Larsson, "Tailoring the software product management framework for use in a healthcare organization: Case study," in *Proc. 20th Eur. Conf. Syst., Softw. Services Process Improvement*, 2013, pp. 155–166.

[58] G. A. García-Mireles, M. A. Moraga, F. García, and M. Piattini, "A framework to support software quality trade-offs from a process-based perspective," in *Proc. 20th Eur. Conf. Syst., Softw. Services Process Improvement*, 2013, pp. 96–107.

[59] S. Jeners, P. Clarke, R. V. O'Connor, L. Buglione, and M. Lepmets, "Harmonizing software development processes with software development settings—A systematic approach," *Commun. Comput. Inf. Sci.*, vol. 364, pp. 167–178, 2013.

[60] M. A. Maidique and B. J. Zirger, "The new product learning cycle," *Res. Policy*, vol. 14, no. 6, pp. 299–313, 1985.

[61] R. E. Morgan and C. A. Strong, "Business performance and dimensions of strategic orientation," *J. Bus. Res.*, vol. 56, no. 3, pp. 163–176, 2003.

[62] C. Parker, "Performance measurement," *Work Study*, vol. 49, no. 2, pp. 63–66, 2000.

[63] S. Hart, "Dimensions of success in new product development: An exploratory investigation," *J. Marketing Manage.*, vol. 9, no. 1, pp. 23–41, 1993.

[64] D. F. Jennings and S. L. Seaman, "High and low levels of organizational adaptation: An empirical analysis of strategy, structure, and performance," *Strategic Manage. J.*, vol. 15, no. 6, pp. 459–475, 1994.

[65] A. M. Ghalayini and J. S. Noble, "The changing basis of performance measurement," *Int. J. Oper. Prod. Manage.*, vol. 16, no. 8, pp. 63–80, 1996.

[66] G. C. Reid and J. A. Smith, "What makes a new business start-up successful?" *Small Bus. Econ.*, vol. 14, no. 3, pp. 165–182, 2000.

[67] H. I. Ansoff, *Corporate Strategy*. New York, NY, USA: McGraw-Hill, 1965.

[68] R. H. Hayes and W. J. Abernathy, "Managing our way to economic decline," *Harvard Bus. Rev.*, vol. 58, no. 4, pp. 67–77, 1980.

[69] S. G. Bharadwaj, P. R. Varadarajan, and J. Fahy, "Sustainable competitive advantage in service industries: A conceptual model and research propositions," *J. Marketing*, vol. 57, no. 4, pp. 83–99, 1993.

[70] R. G. Eccles, "The performance measurement manifesto," *Harvard Bus. Rev.*, vol. 69, no. 1, pp. 131–137, 1991.

[71] M. G. Brown, *Keeping Score: Using the Right Metrics to Drive World-Class Performance*. New York, NY, USA: Quality Resources, 1996.

[72] R. L. Lynch and K. F. Cross, *Measure Up! Yardstick for Continuous Improvement*. Cambridge, MA, USA: Basil Blackwell, 1990.

[73] A. D. Neely, C. Adams, and M. Kennerley, *The Performance Prism: The Scorecard for Measuring and Managing Business Suc-cess*. London, U.K.: Prentice-Hall, 2002.

[74] R. S. Kaplan and D. P. Norton, "The balanced scorecard: Measures that drive performance," *Harv. Bus. Rev.*, vol. 70, no. 1, pp. 71–79, 1992.

[75] M. Kennerley and A. Neely, "Performance measurement frameworks: A review," in *Business Performance Measurement—Theory and Practice*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[76] A. A. De Waal, "Behavioural factors important for the successful implementation and use of performance management systems," *Manage. Decision*, vol. 41, no. 8, pp. 688–699, 2003.

[77] M. Hudson, A. Smart, and M. Bourne, "Theory and practice in SME performance measurement systems," *Int. J. Operations Prod. Manage.*, vol. 21, no. 8, pp. 1096–1115, 2001.

[78] G. S. Sureshchandar and R. Leisten, "Holistic scorecard: Strategic performance measurement and management in the software industry," *Measuring Bus. Excellence*, vol. 9, no. 2, pp. 12–29, 2005.

[79] H. Andersen, I. Cobbold and G. Lawrie, "Balanced scorecard implementation in SMEs: Reflection in literature and practice," in *Proc. 4th SMESME Conf.*, 2001, pp. 103–112.

[80] P. Clarke and R. V. O'Connor, "The meaning of success for software SMEs: An holistic scorecard based approach," in *Proc. 18th Eur. Conf. Syst., Softw. Services Process Improvement*, 2011, pp. 27–29.

[81] Eur. Commission, "Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises. 2003/361/EC," *Official J. Eur. Union*, vol. L, no. 124, pp. 36–41, 2003.

[82] P. Clarke and R. V. O'Connor, "Business success in software SMEs: Recommendations for future SPI studies," in *Proc. 19th Int. Conf. Syst., Softw. Services Process Improvement*, 2012, pp. 1–12.

[83] D. Truex, R. Baskerville, and J. Travis, "Amethodical systems development: The deferred meaning of systems development methods," *Accounting, Manage. Inf. Technol.*, vol. 10, no. 1, pp. 53–79, 2000.

[84] P. Hinton, *Statistics Explained: A Guide for Social Science Students*. London, U.K.: Routledge, 1995.

[85] R. O'Brien, "The use of Pearson's with ordinal data," *Amer. Sociol. Rev.*, vol. 44, no. 5, pp. 851–857, 1979.

[86] W. Harper, *Statistics*. London, U.K.: Pitman Publishing, 1991.

[87] The R Project. (2011) [Online] Available: www.r-project.org. v2.08

[88] M. Unterkalmsteiner, T. Gorschek, A. K. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt, "Evaluation and measurement of software process improvement: A systematic literature review," *IEEE Trans. Softw. Eng.*, vol. 38, no. 2, pp. 398–424, Mar./Apr. 2012.

[89] T. Dyba, "Contextualizing empirical evidence," *IEEE Softw.*, vol. 30, no. 1, pp. 81–83, Jan./Feb. 2013.

[90] R. Baskerville and J. Stage, "Accommodating emergent work practices: Ethnographic choice of method fragments," in *Proc. Realigning Res. Practice IS Develop.: The Social Organisational Perspective*, 2001, pp. 12–28.

[91] M. P. Ginsberg and L. H. Quinn, "Process tailoring and the software capability maturity model," Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-94-TR-024, 1995.

[92] J. Cameron, "Configurable development processes," *Commun. ACM*, vol. 45, no. 3, pp. 72–77, 2002.

[93] A. N. Bowers, R. S. Sangwan, and C. J. Neill, "Adoption of XP practices in the industry: A survey," *Softw. Process: Improvement Practice*, vol. 12, no. 3, pp. 283–294, 2007.

[94] R. R. Nelson and S. Winter, *An Evolutionary Theory of Economic Change*. Cambridge, MA, USA: The Balknap Press of Harvard Univ. Press, 1982.

[95] H. A. Aldrich, *Organizations Evolving*. London, U.K.: Sage, 1999.

[96] H. A. Aldrich, *Organizations and Environments*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1979.

[97] D. J. Teece, "Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance," *Strategic Manage. J.*, vol. 28, no. 11, pp. 1319–1350, 2007.

Paul Clarke received the PhD degree in 2012 from Dublin City University, where he is currently a lecturer in software engineering. He is also a research fellow with Lero—the Irish Software Research Centre and a nominated national delegate to the ISO/IEC JCT1/SC7 standardization body, where he is actively involved in the development and evolution of best practice standards for software engineering. Previously, he held the post of research manager in the Regulated Software Research Centre based at Dundalk Institute of Technology, while earlier he also practiced software engineering in various industrial based organizations for 12 years. His research interests include software processes, software development methodologies and standards, safety critical software development, software process adaptation and improvement, economics of software engineering, and team dynamics.

Rory V. O'Connor is an associate professor of software engineering at Dublin City University, where he is currently serving as the head in the School of Computing. He is also a senior researcher with Lero, the Irish Software Research Centre and is Ireland's head of delegation to the ISO/IEC JCT1/SC7 standardization body. His research interests are centered on the processes and standards whereby software-intensive systems are designed, implemented, and managed. He is currently the editor in chief of the journal *Computer Standards and Interfaces*.

Brian Leavy received the BSc (physics and maths) and MBA degrees from National University of Ireland (NUIG), Galway, and the PhD degree from the University of Warwick. He spent the first eight years of his career as a technical trainer and manufacturing engineer in the IT sector with Digital Equipment Corporation, now part of HP, where one of his responsibilities was for manufacturing test systems diagnostic software support. He has been a business academic in Dublin City University since 1981, and his main research interests are in strategy, leadership, and innovation.

Murat Yilmaz received the master's degree in software engineering from the University of Minnesota, in 2009, and the PhD degree in computer science from Dublin City University in 2013. With more than a decade of experience in software development and project management, he worked for several IT companies as a software developer, software architect, technical lead, systems engineer, and project coordinator. He is currently an assistant professor with the Department of Computer Engineering, Cankaya University, and is also an adjunct faculty member of the Informatics Institute, Middle East Technical University. His research interests include empirical software engineering, software project management, game theory and design, serious games, team dynamics, and gamification.