Software Process Reflexivity and Business Performance: Initial Results from an Empirical Study

Rory V. O'Connor
Lero, Irish Software Research Centre
Dublin City University
Dublin
Ireland
Rory.OConnor@dcu.ie

Paul Clarke
Dublin City University
Dublin
Ireland
Paul.Clarke@computing.dcu.ie

ABSTRACT

Commercial software development organisations routinely operate in dynamic environments, with various situational factors that affect the software development approach undergoing recurring change. We therefore suggest that process reflexivity - the ability to reflect upon the suitability of a software process for a given context and to adapt the process according to changing situational circumstances – is an important capability for software development organizations.. In support of this position we conducted an exploratory industrial study of software development in practice. An initial analysis of our data suggests that software process reflexivity may exercise a strong influence over business success. Further work is required to fully examine our data, however, initial findings indicate that software process reflexivity is worthy of greater attention.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management - Software process models.

Keywords

Software process improvement, process reflexivity, process evolution.

1. INTRODUCTION

In his landmark 1859 book, The Origin of Species, Charles Darwin showed that those species that adapt best to their changing environment have the best chance of surviving, while those who do not adapt do not make it. Similarly human beings adapt to changes in their environment by making changes in their behaviour and actions, such as wearing a hat and gloves on a winter's day. In the world of software development, organisations routinely operate in dynamic environments, with various situational factors that affect the software development approach undergoing recurring change. It therefore seems reasonable that such an evolutionary concept of 'adapt or die' is relevant for software development organisations. In a recent industrial study of software development in practice, we found that data collected from a variety of different software development companies supports the view that software process reflexivity may have an important role to play in supporting business success, thus advancing the case for further examination of the role of software process reflexivity in practice.

ICSSP'15, August 24–26, 2015, Tallinn, Estonia
ACM. 978-1-4503-3346-7/15/08...$15.00
http://dx.doi.org/10.1145/2785592.2785607

Strongly related to this concept of process reflexivity, the literature on managerial and organizational learning has frequently drawn on the concept of reflection, indicating a desirable practice of inquiry that is characterized by engaging in comparison, pondering alternatives, taking diverse perspectives and drawing inferences (e.g. [1]). Reflection is often emphasized as a critical ingredient of practice in certain professions such as the medical domain, where complex and new situations require high levels of situational awareness. However, relatively little is known, however, about how organizations can actively foster such a reflective attitude and about the challenges that may be encountered when trying to do so.

This is particularly the case in complex and novel situations that call for high situational awareness, where the concept of reflective processes has been granted a major role in management literature [2]. Schon [2] characterizes 'reflective practitioners' by their tacit knowing-in-action and their ability to question their knowing-in-action and the underlying 'framing' of the situation, when confronted with complex, novel tasks and unprecedented events. This view of professional practice stands in opposition to technical rationality, according to which professions such as medicine or civil engineering which comprise a standardized body of knowledge that has to be acquired by novices and which is applied in order to solve predefined problems in practice. In contrast, a reflective practitioner is characterized by the ability and willingness to question routinized ways of thinking and acting. When the volatility of software development is taken into consideration, a good example of which is the varying degree of requirements uncertainty, one can intuitively sense that routinized thinking alone may not be an ideal solution to the software development challenge. Thus, we look to other modes of thinking to help supplement software development process management, with evolutionary theory presenting as a potentially valuable concept.

1.1 Evolutionary Theory of the Firm

The idea that organisations need to continually reflect on and adapt their processes in response to changing situational contexts has received much attention in the field of economics, for example, the neoclassical theory of the firm presents profit maximisation as the reason for existence [3]. However, other theories of the firm (to be used in conjunction with the neoclassical theory) have emerged over time and one of these, the Evolutionary Theory of the Firm [4], appears to describe many of the characteristics of software development organisations. According to the evolutionary theory of the firm, a company is constituted by both physical and human assets, and there is a strong focus on organizational capability [5] where the ability to reflect, respond and adapt in view of changing circumstances and information is considered central to continual organisational success. In essence, the evolutionary theory of the firm suggests

that organisations that are better at learning, that can utilise their spare capacity to promote adaptation and innovation, are more likely to be successful and to endure. This indicates that companies should seek to gain a strategic advantage through continuous product, process and organisational innovation and that this is achieved by ensuring that flexibility exists at all levels in the company. The fundamental perspective is that "improvement is always possible and ideas for improvement can come from everyone", with the premise that firms "depend upon learning to maintain a competitive advantage" [6].

Since they continually create new software and often depend on innovation as a cornerstone of their business, software development companies would appear to offer a particularly good fit for the evolutionary theory of the firm. Software companies exist in a fast moving, innovative environment and don't have self-contained, slowly evolving products that are pushed off the manufacturing line but rather, the product is continually evolving. Therefore, there is a compelling need for software development organisations to continually apply accumulated experience and knowledge to work practices. While not necessarily grounding his ideas in the evolutionary theory of the firm, Watts Humphrey recognises the importance of learning and process improvement stating that "reactive changes generally make things worse... [and] crisis prevention is more important that crisis recovery" [7]. Evolutionary theory can offer similar wisdom, promoting the importance of metamorphosis via learning within an organisation, for both the products and the production process.

1.2 Software Process Evolution

The software process constitutes a significant and complex component of a software development business, and therefore, when viewed through the evolutionary theory of the firm lens, the success of a software business will be affected by the degree to which the organisation is capable of adapting its software process. It would therefore appear beneficial for software development companies to measure their software process reflexivity– so as to have insight into the extent to which they are learning and evolving.

The process utilised when developing software is an important consideration for software development endeavour. It is accepted by many that software organizations should routinely reflect on their development process in order to evolve or adapt to best support the changes that are inevitable in any given real world setting. However, it remains a challenge to observe the extent to which a software development organisation has been able to reflect and adapt – since the facets of both the software process and the environmental factors that affect it are complex in nature. It is nonetheless reasonably clear that the extent of desirable or optimal software process adaptation is dependent on the degree of situational change (the change that has manifested in the real world setting) – even if these two phenomena are complex in nature and challenging to interrelate. Therefore, focused reflection on current development processes and an analysis of the extent of process adaptation should ideally incorporate a measure of situational change.

It may perhaps be the case today that much of our efforts in the software process space are inhibited due to an absence of techniques for relating situational demand for process with actual process enactment – but this should come as no surprise as there is a daunting (and quite possibly intractable) complexity challenge in trying to fully harmonise each and every situational nuance with corresponding process solutions [15]. However, this challenge can be partially overcome through the adoption of techniques that simplify the problem into its larger component parts, a good example of which is the five dimensional Boehm Turner model [16] for determining the extent to which a software process should be agile. It is in this spirit that we undertook an exploratory study that aimed to determine broad (but useful) approximations of process change and situational change. An initial analysis of our data is presented herein, and it would suggest that this simplification of the complexity challenge to broad approximations of the central phenomena of interest could be a useful technique for examining software process performance in an organisation.

1.3 Importance of Process Reflexivity

While it seems unlikely that we will be able to design a single software process that will optimally address all software development settings, it is desirable to examine individual software development settings with a view to architecting an appropriate, tailored development process. Since aspects of the world are inevitably subject to change, it is reasonable to assume that an optimal software development process should require regular adaptation - ranging from minor improvements to major overhauls 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 reflexivity than other business sectors. We therefore suggest that it is desirable to be able to measure the extent of software process reflexivity – even if only in an indicative fashion.

In traditional capability maturity frameworks, such as ISO/IEC 15504 and CMMI, process improvement is supported by the definition of maturity levels, where at the highest level, continuous process improvement is promoted as the key to process improvement. However, few organisations ever attain the highest process maturity level and the transition between the different maturity levels is fixed in nature. As such, capability maturity frameworks do not necessarily offer an effective mechanism for adapting the software development process in consideration of specific changing situational contexts. Furthermore any given process capability can only be optimal for a brief period, since the nature of the world is one of change rather than permanence.

Certain agile software development approaches have advanced the role of process reflexivity. For example, a Scrum retrospective is designed to review the effectiveness of a *sprint* (an implementation cycle). However, retrospectives tend to be conducted within individual teams and although they are designed to promote process reflection and consequent improvement, retrospectives are geared towards subtle improvements within the general Scrum technique. In common with capability maturity frameworks, Scrum does not provide a mechanism for measuring the extent to which an organisation has adapted the software development process over a period of time – even if it may offer improvements over traditional approaches with respect to supporting process adaptation..

Due to the relationship between software process change and situational or environmental change, any analysis of process reflexivity performance should ideally incorporate a measure of situational change. If both the amount of process change and the amount of situational change are taken into consideration, it is possible to get an indication of the performance of a software development organisation in terms of process reflexivity. Given that the existing software development approaches discussed in

the previous paragraphs offer no support for measuring such process reflexivity performance, we have developed a new technique for deriving an indication of a software firm's software process reflexivity.

2. FIELD STUDY

As already noted, to obtain a meaningful indication of the extent to which an organisation has adapted its software process to changing circumstances requires that we examine two distinct phenomena: firstly, the amount of software process change, and secondly the degree of change in the factors that affect the software process. Consequently, we developed survey instruments to measure process change and situational change – and discharged these instruments in 15 software companies. In addition, a third survey instrument was established to examine the extent of business success, allowing for a broad evaluation of the relationship between software process change, situational change, and business success in software companies.

The amount of software process change in an organisation could be indirectly determined through the use of two separate process assessments (using methods such as CMMI SCAMPI or similar) spread over a given time period and conducting a differential analysis of the two capability profiles. This however represents a convoluted and inefficient approach to determining the amount of software process change in an organisation – as it is concerned with process capability (from which process change can be derived) as opposed to directly measuring process change. Therefore, we developed a new survey instrument to examine process change that requires just a single engagement with an organisation [8] – an approach based on the ISO/IEC 12207 systems and software lifecycle processes standard [17].Our software process change survey instrument, which contains 63 individual questions, was subjected to rigorous independent review by external experts including ISO/IEC 12207 authors and editors (ISO SC7 Working Group 7) and was also piloted to get feedback from an industrial perspective.

In order to create a situational change survey instrument, it was necessary to adopt the most extensive, available and relevant reference framework [9].The primary purpose of this instrument is to provide a profile of the extent of change that occurred over a period of time in the situational factors that are known to affect the software process. In deriving the survey instrument from the underlying situational factors reference framework, the guiding principle was that all of the individual factors from the reference framework should be addressed in individual questions in the survey instrument (and where appropriate, multiple questions should be developed for individual factors; for example, where a large number of sub-factors are available). Once produced, the situational factors survey instrument, which contains 49 individual questions, was also subjected to industrial piloting.

In the creation of a business success survey instrument it is necessary to consider the various dimensions of business success for software development organisations. In the business literature, the term success is often used interchangeably with the term performance, with both terms representing the achievement of something desired, planned or attempted. However, beyond this general description, there is some ambiguity in terms of what constitutes business performance. In recent decades, there has been a shift away from treating financial measures as the sole foundation for performance measurement to treating them as just one pillar supporting business success - and this has given rise to multidimensional performance measurement frameworks [10]. Of the numerous multidimensional performance measurement frameworks that have been developed, the Balanced Scorecard [11] is the most popular and has been rendered into a strategic performance measurement and management framework for the software development industry known as the Holistic Scorecard (HSC) [12]. Using the HSC as a foundation, we systematically developed a business success survey instrument comprising 51 individual questions (applying a combination of internal formal review and external industrial piloting) [13]. Since not all organisations are 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, our business success investigation consists of two distinct phases. The initial phase formally identifies the business goals in each of the HSC dimensions for the forthcoming year. The second phase involves revisiting participating organisations at the end of the year under investigation, at which point each of the previously elicited objectives is evaluated in terms of achievement..

2.1 Data Evaluation

Over a 16 month period and using direct interviewing, we applied our survey instruments to the task of examining the amount of software process change, situational change and business success in software SMEs (small to medium sized software companies with less than 250 employees). Measuring the extent of business success in tandem with software process change and situational change permits assessment of possible relationships between these three phenomena.

In general, the business success questions were addressed by individuals such as the Chief Executive Officer, Chief Operating Officer, Managing Director, or Director of Finance. Questions in relation to the software process and situational change were generally addressed to the Director of Engineering, Chief Technical Officer, Engineering Manager or Development Manager. Since job titles can vary from organisation to organisation (as does the remit and extent of knowledge of the individual undertaking any particular role), one of the initial discussions with each participating organisation was aimed at identifying the most suitable person(s) to participate in different aspects of the investigation. A total of four separate engagements were required with each company: one each for software process change and situational change, and a further two for business success (the first to establish the business objectives for the forthcoming year, and the second (one year later) to determine the extent to which the objectives were achieved).

Using the data collected from the survey instruments, approximate measurements were produced for the three phenomena under investigation: software process change, situational change, and business success. These approximations were subject to rigorous statistical analysis using appropriately selected statistical methods, techniques and tools. Due to space limitations a detailed explanation of the data analysis is not presented here, rather summary results are discussed.

Perhaps the most striking feature of the data is that those organisations with levels of process change that outstrip reported levels of situational change are tending to also report higher levels of business success. It should be highlighted that there is a risk here of treating apples as being the same as oranges as all three phenomena are examined using different base frameworks. However, we are interested in relative levels for each of the phenomena, and just as we might expect to see high levels of vegetation growth resulting from relatively high levels of rainfall

coupled with relatively high levels of sunshine, so too are relatively high levels of situational change coupled with relatively high levels of process change likely to support relatively higher levels of business performance. This is one of the tenets of the evolutionary theory of the firm. Figure 1 shows the values recorded for these three distinct phenomena for the 15 participating companies.

In briefly examining this data, we have chosen to highlight 2 clusters of companies, which appear worthy of discussion. Cluster 1 shows companies with the lowest business success, revealing that in a general (and visual) sense, lower business performance tends to be associated with lower levels of software process change. The second cluster illustrates that companies with higher levels of situational change, coupled with higher levels of software process change are those with the higher scores in terms of business success. Further evidence that business success interacts with both process and situational change may be found in the case of Company 9. This company records the single highest amount of both software process change and situational change, yet it is not the most successful company. One possible explanation for this is that the amount of software process change is not particularly large relative to the degree of situational change.

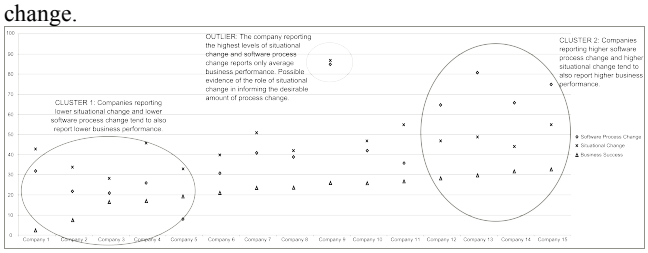

Figure 1. Recorded Software Process Change, Situational Change and Business Success values.

Our study was limited to just 15 small to medium sized companies which (although it required significant time investment and effort) is a very small sample set. Nonetheless, the initial review of the data suggests that there is a trend to be observed here – and it is a trend that can be considered to be largely consistent with the evolutionary theory of the firm: companies that are better at adapting their software process vis-a-vis their changing situational context are tending to witness relatively greater levels of 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 and can produce an equivalent product for less, it may be difficult to develop a viable survival or recovery strategy. And no amount of process innovation may be capable of overcoming such a scenario.

Another possible explanation for the relationship between the phenomena is that software process change results in an improved process which supports greater business success. From the statistical analysis of our data, software process change and business success were highly positively correlated. It could be further proposed that an increased awareness of situational change is a prerequisite for software process change and therefore the reported situational change is the key ingredient underlying the apparent relationship. Our data supports such a viewpoint, as the statistical correlations for situational change and business success are also positively correlated.

Rather than viewing the possible explanations for the relationships presented above as being mutually exclusive, it is the view of the authors that the interrelationships are in fact amethodical. An amethodical relationship is one in which a change to one phenomenon can result in a change in a second phenomenon, with the consequent change in the second phenomenon possibly giving rise to a further change to the phenomenon that experienced the change in the first instance (i.e. something that can be considered similar to symbiosis is at play). Furthermore, multiple phenomena can constitute a system, with all phenomena influencing each other in a series of reflexive type interrelationships.

Our initial analysis of the data gathered from the study supports the view that the relationships between the phenomena are amethodical. For example, increases in business success are positively associated with increases in software process change. Similarly, increases in situational change appear to be positively associated with increases in business success. 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 reflexivity. 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. An outcome of this observation is that all three phenomena are important considerations that can benefit from measurement – if only approximate measurement.

It is therefore the case that the absolute values recorded for the various phenomena may themselves be important quantifications for consideration. For example, if an organisation reports relatively small amounts of process change, one can reasonably be apprehensive – since it is not likely that a company continually has a perfectly suitable set of processes (over a period of time such as the 12 months under examination in the exploratory study), and it is generally not the case that an environment is completely unchanging. Therefore, if an organisation reports relatively low levels of situational change, it could be the case that there may be a shortcoming in the organisation in terms of perceiving change (something that is not desirable from an evolutionary perspective).

Taken together, the evaluations would appear to support the role of evolution in supporting business success. We therefore believe that a combination of absolute values for the phenomena, considered together with the relationship between software process change and situational change, represents useful new measurements for software development companies. In later works we intend to undertake a detailed and formal statistical analysis of the study data but for now, the initial observations suggest that better business performance may be achieved where process adaptation is relative to situational change. The evidence also demonstrates that process reflexivity is a complicated business – it requires not just sensitivity to situational change but also a corresponding capacity to implement process change. However, and despite the results supporting the theoretical role of the evolutionary theory firm when considering software processes, perhaps the most limiting aspect of the exploratory study discussed herein is the sample size, which is certainly too small from which to claim a generalizability in the findings – but it is sufficient to suggest that the ability of organisations to adapt may

be strongly related to overall success and therefore an important consideration for software companies.

3. DISCUSSION

It is a basic requirement of a software development process that it should fit the needs of the organisation that it serves. And these needs are not static but rather they are subject to change – and therefore software process evolution should be an important consideration. The role of evolution as an ingredient for business success in general has long been established and it has been suggested that companies no longer compete on processes but on the ability to continually improve processes [14]. If this is the case, then software development companies should continually reflect on the appropriateness of their methods of work, modulating their software processes according to their circumstances.

In an exploratory study, we have empirically examined the role of process evolution in software development companies in practice by collecting data on the extent of change to the factors affecting the software process over an extended period, while in tandem also examining the extent of software process change and business success. The analysis of the collected data demonstrates that business success tends to be greater in companies with better support for process evolution. Caution should be applied when interpreting the results of the data analysis as the exploratory study (and corresponding initial analysis) exhibits some limitations, such as the number of participating organisations and the restriction to small and medium sized companies. Nonetheless, over 80 hours of direct interviewing time was invested in reliably extracting the underlying data, and a substantially larger investment again was required for the data analysis and evaluation.

Despite the noted limitations, the present evaluation of the data raises the possibility that process evolution may be of value to the software development community, particularly practitioners. Adapting software development processes to changing situations is a task that most software development organisations are challenged with. Given the large and complex nature of both software development processes and situational factors that can affect the software development process, the approximations outlined in this article are potentially of significant assistance to software development companies as an indicator of their performance in adapting their software processes.

In essence we have discovered that where software process change is relatively greater than situational change, businesses are reporting increased levels of business success. This finding supports the idea that organisations need to continually reflect on and adapt their processes in response to changing situational contexts. This would appear to lend credence to the proposition that the evolutionary concept of 'adapt or die' is as relevant for software development organisations as it is in the natural world. Therefore we suggest that software process reflexivity should be considered as an important and independent software development process concept. And it is perhaps the case that in adopting a process reflexive viewpoint, the debates that have raged over which software development process is best can be quelled – since no individual approach is best as all settings are different and subject to continuous change.

4. 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).

5. REFERENCES

[1] Kolb, D. A. (1984) Experiential Learning: Experience as the Source of Learning and Development. Prentice-Hall.

[2] Schon, D. A. (1983) The Reflective Practitioner: How Professionals Think in Action . New York: Basic Books

[3] Cyert, R. M. and Hedrick, C. L. 1972. Theory of the Firm: Past, Present, and Future; An Interpretation. Journal of Economic Literature. 10 (2), pp398-412.

[4] R. R. Nelson and S. Winter, An Evolutionary Theory of Economic Change. Cambridge, Massachusetts, USA: The Balknap Press of Harvard University Press, 1982.

[5] Chandler, A. D. 1992. Organizational Capabilities and the Economic History of the Industrial Enterprise. The Journal of Economic Perspectives. 6 (3), pp79-100.

[6] Best, M.H. 1990. The New Competition, Polity Press.

[7] Humphrey, W.S. 1989. Managing the Software Process. Reading, Massachusetts, USA: Addison-Wesley.

[8] P. Clarke, and R.V. O'Connor, An empirical examination of the extent of software process improvement in software SMEs, Journal of Software: Evolution and Process, 25 (9), pp. 981-998, 2013.

[9] P. Clarke, and R.V. O'Connor, The situational factors that affect the software development process: Towards a comprehensive reference framework, Information and Software Technology, 54 (5), May 2012. pp. 433-447.

[10] R. G. Eccles, "The Performance Measurement Manifesto," Harvard Business Review, 69 (1), pp. 131-137, 1991.

[11] R. S. Kaplan and D. P. Norton. "The Balanced Scorecard - Measures That Drive Performance," Harv. Bus. Rev., 70 (1), pp. 71-79, 1992

[12] G. S. Sureshchandar and R. Leisten. "Holistic Scorecard: strategic performance measurement and management in the software industry," Measuring Business Excellence, 9 (2), pp. 12-29, 2005

[13] P. Clarke, and R.V. O'Connor, The Meaning of Success for Software SMEs: An Holistic Scorecard Based Approach, In O'Connor, R.V., Pries-Heje, J. and Messnarz R. (Eds), Systems, Software and Services Process Improvement, CCIS Vol. 172, Springer-Verlag, June 2011

[14] D. J. Teece, "Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance," Strategic Management Journal, 28 (11), pp. 1319-1350, 2007.

[15] T. Dyba. "Contextualizing Empirical Evidence," IEEE Software, 30 (1), pp. 81-83, 2013.

[16] B. Boehm and R. Turner, Balancing Agility and Discipline - A Guide for the Perplexed. USA: Pearson, 2003.

[17] ISO/IEC, ISO/IEC 12207-2008 - Systems and Software Engi-neering – Software Life Cycle Processes. Geneva, Switzerland: ISO, 2008.