Expanding on the Process Perspective in Software Process Improvement Practices

HARRI KETO, JARI PALOMÄKI and HANNU JAAKKOLA, Tampere University of Technology

Process modeling is an essential part of software process improvement. This article aims at expanding the dominant paradigm of processes modeling where the organization's operations is seen as an idealistic mechanistic process. Our view is that enhancement of the process improvement practices requires re-evaluation of the main concept, that is the process view. Process philosophy and system thinking is offered to expand the perspective. Modeling involves essentially the ability of the modeler to abstract, i.e., the ability to look at the modeling object from a chosen point of view, and thereby distinguish and conceptualize the properties considered relevant to the purpose of the modeling.

1. INTRODUCTION

Looking at the development of software engineering (SE) practices from the beginning of the discipline to this day, a significant change in generic processes and practices in the field can be seen. In the last two decades, the agile methods have become part of the official practices of software organizations. The software industry has developed instead of the traditional linear predetermined process to adhere more and more adaptable and iterative process. This change seems to have contributed to the software process improvement (SPI) practices as well. The change of perspective is also seen as an interest in the research papers and development of standards [Henriques and Tanner, 2017] [Keto and Jaakkola, 2017]] [Kuhrmann, Diebold and Münch, 2016].

The concept of the process is in many ways central. SE processes are modeled to provide a unified concept as a foundation for process improvement. The process is usually modeled as a system that, by means of related functions and resources, converts inputs into meaningful products. This provides a structured and easily understandable perspective. At the same time, however, it must be remembered that this is only an abstraction involving certain features. We call this kind of systematic structural partitioning approach a *mechanistic process conception* because it easily creates an illusion system which behavior is wrongly assumed to be predictable by understanding the behavior of its components [Keto, Palomäki and Jaakkola, 2010] [Lindsay, Downs and Lunn, 2003].

This article aims at expanding the dominant paradigm of processes modeling where the organization's operations simply become an idealistic mechanistic process. Our view is that enhancement of the process improvement practices requires re-evaluation of the main concept, that is the process view. The writing relies partly on the historical perspective, as we want to highlight the background of the prevailing paradigm's development over time. Chapter 2 gives

Author's address: Harri Keto, Tampere University of Technology, PL 300, 28101 Pori, Finland; email: harri.keto@tut.fi; Jari Palomäki, Tampere University of Technology, PL 300, 28101 Pori, Finland; email: jari.palomaki@tut.fi; Hannu Jaakkola, Tampere University of Technology, PL 300, 28101 Pori, Finland; email: hannu.jaakkola@tut.fi.

In: Z. Budimac (ed.): Proceedings of the SQAMIA 2018: 7th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications, Novi Sad, Serbia, 27–30.8.2018. Also published online by CEUR Workshop Proceedings (http://ceur-ws.org, ISSN 1613-0073)

an overview of changes in the process improvement. We look at the evolution of the overall process of software development and the appropriateness of SPI from a process management perspective. In Chapter 3, we point out a basis for critical review by making a brief overview on process philosophy and system thinking. One motivation behind our view is the criticism of the system thinking and the process philosophy to the reductive thinking of classical science. In Chapter 4 we open the discussion about the subject.

2. PROCESS PERSPECTIVE CHANGE IN SPI

Many process management practices in SE have been adapted through practical experience from closely related sciences and industry. SE has been characterized by the development of process innovations in practical projects. These lead to good practices that get gradually gaining foothold in the industry. At some point, researchers summarize these features as theories that get the official label in the industry [Shaw, 1990]. In this section, we first propose how the change in the SE process enables new kinds of practices to improve processes, but on the other hand, it also poses challenges for comprehensive development. Then we look at the development of process thinking in process management.

2.1 Evolution of the software engineering process and implications on SPI

In the early days of information technology, SE was seen as a straightforward *linear process*, like a manufacturing industry, where the product is first defined and planned accurately and the specification is fixed before implementation and test. This sounds a well-controlled process, but there were also early warnings about making the process too linear. The famous writing of Winston Royce demonstrated the nature of the SE's distinction with the traditional manufacturing industry [Royce, 1970]. Instead of using a linear top-down process, Royce suggested to use early simulation, which can be interpreted as prototyping. The use of a prototype in producing computer programs was reported already in 1956 [Benington, 1983]. Since the 1970s, the linear model was commonly referred to as the *Waterfall Model*, and various variations were developed based on it. The SE was aimed to pursue as a *plan-based* process.

The Spiral Model released by Barry Boehm highlighted the features of modern SE such as *iteration*, a *gradual refinement* of the product, utilization of *prototypes* and *risk management, see* [Boehm, 1988]. To the model was embedded a notable feature that is nowadays apparent in the applications of all *agile methods*: today the sequential steps of the waterfall model have been translated into assignments that repeat in each cycle of the iteration. The purpose is to produce a ready-to-evaluate outcome as an output of each cycle. Balancing between a plan-based process and the agility has led to iterative and *incremental* ways to implement software. The incremental refers to a progressive development process in which the customer gets access to some of the software while at the same time new increments are planned. A process is called *evolutionary* development, when the specification of the software may not necessarily be entirely attached at the beginning of the project but it may change during the process, as the design is refined.

In agile SE, short iterations are intended to produce a concrete result to be evaluated and which adds value to the system. A concrete process example can be found in the *Scrum* method, which process is based on sprints of 1-4 weeks, of which the *self-managed team* is responsible [Schwaber and Beedle, 2001]. The basis for method was described by Takeuchi and Nonaka, see [Takeuchi and Nonaka, 1986]. Self-managed teams are characterized by features such as a common task, versatile members, the ability to decide on their work patterns, schedule and distribution of tasks, and it gets feedback as a team [Cummings, 1978]. Self-management means that the team has the prerequisites for acting as freely within the work unit and that it can change its performance strategy in a changing operating environment to solve problems [Wageman, 1997].

The agile process is based not only on iterativity and incrementality, but also the most *straightforward and rapid communication* between the different stakeholders and within the development team. Process performance relies more on *team expertise* than on the predefined and external controlled process, though many of the practices to be followed are highly *disciplined* and *automated*. For example, in XP programming, a maximum automation is sought for configuration and version control, and this requires compliance with common rules [Beck, 2004]. Development seems to have led to the popularity of agile *process frameworks* which include a *knowledge-base* and *workflow patterns*, see for example [Scaled Agile, 2018]. It is common for them, that they are not content with just describing the steps of the process, but they provide tangible and experimental information on the best practices and tools to support the project resources, roles and tasks. The aim is to provide work models that can be *tailored* for an individual organization or for a project.

The Scrum process possesses an important feature for improvement of operations, namely *Sprint Retrospective*. Instead of external activity of an outside expert, the retrospective is practically embedded into the process as one of the required process event. During the retrospective, the team discusses what went well in the last iteration (sprint) and how to improve work processes. By the end of the retrospective, the team should have identified improvements that it will implement in the next iteration [Schwaber and Sutherland, 2017]. The popularity of agile methods, and in particular the Scrum, poses challenges on the one hand, but also provides a practical platform for improving processes. Emphasizing the team's expertise and communication creates a favorable soil for commitment. The project specific retrospective cannot replace a wider development program, but on the other hand, agile projects may provide them with a good platform to implement and experiment with new practices.

2.2 Considerations of process thinking in process management

Philosopher and economist Adam Smith presented the idea that the work can be parsed into basic tasks. The work was organized in a new way to mass production. From the point of view of systemic thinking, the production process was seen as a machine controlled by instructions and rules, all of which could be controlled by the "owner" [Ackoff, 1994]. At the beginning of the 20th century, Henry Ford designed a new kind of way to produce high-quality cars to their competitors much more efficiently and more economically in a well-defined serial production process. This and Frederick Taylor's "Principles of Scientific Management" have a remarkable impact as a model for modern production process. Taylor emphasized, among other things, the importance of simplifying work, evaluating and improving performance, and the importance of controlling work results that can be clearly seen as an application of the open system feedback loop. The idea of a company's organization changed substantially. Ackoff notes that it was inevitable for management to see an organization as a social system, some of which could have goals of their own [Ackoff, 1994]. This can be seen as an attempt to move from a mechanistic process to a wider perspective.

Process Management (PM) is a management discipline where the basic unit of management and organization is a *business process* that crosses horizontal organizational boundaries. PM includes several trends. One of the key drivers of PM can be seen in Michael Porter's *value chain* concept, where the *core processes* of a company constitute a level, in where each step of the action chain adds value to the product, see [Porter, 1985]. In addition, Geary Rummler emphasized the importance of organizational performance enhancement and process re-design, see [Rummler and Brache, 1990]. *Process improvement* practices include modeling, evaluating, and measuring operational processes as well as making improvement actions based on these. In most Total Quality Management (TQM) approach quality improvements are achieved through using the "Plan-Do-Check-Act" cycle, an approach associated originally with the ideas of Schewhart and Deming [Houston and Dockstader, 1988].

In the early 1990s, the position of information technology (IT) in relation to organizations' business processes changed substantially. The role of IT, previously the automation of work, became one of the key business transformation factors that would allow a radical re-engineering of business processes, see [Hammer, 1990], [Davenport and Short, 1990] and [Harrington, 1991]. Business Process Reengineering (BPR) did not finally prove successful. Instead of gradually improving the process, the idea was to create a completely new kind of process that would radically differ from old practices. After initial interest, however, the favor of BPR gradually faded. One reason was the misuse of the methodology [Harrington, 1998]. Both Hammer and Davenport have noted that the approach did not take into account the human dimension and Davenport wrote: "*Reengineering treated people as interchangeable parts to be reengineered"* [Wang, 2008, 27]. It can be concluded that the transfer of the process control mechanism from the manufacturing industry to design and service-oriented business seems to have led to deadlock because the perception of the process was highly mechanistic. The rise and fall of BPR is a good example of the risks of a mechanistic process conception.

In the mid-1980s, the trend to develop the productivity and quality of SE from a process point of view was strengthened. At the same time, ISO 9000-based quality development programs became increasingly popular also in the development of software organizations. The roots of the quality movement are strongly in PM and TQM practices, which both share process thinking as an important aspect of the organization management. Particularly in the context of TQM, process improvement is seen as iterative and controlled change practice [Houston and Dockstader, 1988] [Hackman and Wageman, 1995]. Software Engineering Institute (SEI) research can be considered as one of the first impulses in Software Process Improvement (SPI) [Curtis, Kellner and Over, 1992]. Today the most famous models are SEI's CMM (Capability Maturity Model) and ISO standard ISO 15504, which is to be renewed into the ISO 33000 series. The use of large standard models has been debated by researchers, especially in the case of small and medium-sized software organizations. [Richardson and Cresse von Wangenheim 2007] [Raninen 2014]. Despite criticism, in any case, standardization has brought together a lot of significant information about the SE.

3. A GROUND TO EXPAND THE PERSPECTIVE

In connection with the criticism of reductive thinking, we present two significant philosophical approaches. The first, Process Philosophy can be used to explain the object as a dynamic phenomenon. The second, System Thinking has had a major impact on process modeling. They have affected many methods and practices of SE, but their impact is not necessarily known or their interpretation is superficial.

3.1 Process philosophy

Process philosophy is based on the claim that the existence is of a procedural nature and that change is the dominant feature of reality everywhere [Rescher, 1996, 7-8]. Process philosophy is characterized by emphasizing the primacy of activity and related concepts such as time, change and evolution [Rescher, 1996, 27-28]. In process philosophy, reality is perceived to be dynamic and is generally presented as a contrast to the more static substance-based philosophy. Substance is a permanent independent object that may exist irrespective of other objects. In the ancient era, Heracleus emphasized the process and the constant movement, as opposed to Parmenide's idea that nothing moves. Platon later combined these two opposite ways of thinking. According to him, the sensible world is in motion and changing, and we only have "*doxa*", that is, common belief. However, Plato's pupil, Aristotle established substance thinking on philosophical and scientific thinking so that a more dynamic understanding of reality began only in the late 18th century to come back to philosophical consideration.

According to Alfred Whitehead, the world is best understood as systems of large and small finite four-dimensional space-time events, some of which are relatively stable [Whitehead, 1929]. Space-time events are linked either on within other or overlap and cut together so that chains of events can be formed. Events are in constant change. The change represents the realization of certain opportunities and the disappearance of others. The world simply does not exist as such but is in a state of continuous becoming. According to Whitehead, reality is the process of becoming of actual entities (or actual occasions) [Whitehead 1929, 22]. They endure only a short time, and they are processes of their own self-creation. There are also eternal objects to be understood as conceptual objects. They express a concrete reality without actually being real.

Following Whitehead's idea that everything is made up of processes, for every phenomenon there is a modeling point of view, by which it can be seen as a process or group of processes. These processes can be divided into eternal processes interpreted as concepts and into real processes interpreted as finite four-dimensional time-space events. When looking at process modeling and the processes involving people, we also need to take into account the different subjective perceptions of the process. Thus, we can distinguish three interrelated process abstraction levels: *the ideal conceptual level*, *the subjective conceptual level* and the *level of physical real time events*, see more in [Keto, Mäkinen and Linna, 2011]. The ideal level consists of the concepts of the process. The subjective level is the individual actor's point of view to the actual physical process.

3.2 System thinking

By systemic thinking (ST), we refer to a broad range of interdisciplinary approaches that are bound together by the concept of the system and the holistic approach to analysis [Checkland, 1999] [Pyster and Olwell, 2013] [Mingers and White, 2010] [Lane, 2016]. According to the classical concept of science, the system can be extracted from the constituents and their properties. Reductionist analysis, through which a complete understanding of the world was achieved, was however criticized at the beginning of the 20th century. This ultimately led to the birth of the modern system thinking. Checkland examines the evolution of ST as part of the general evolution of science, see [Checkland, 1999, 23-58]. He presents, that the method of traditional science is still evolving methodologically to reductive direction, and that the diversity of the world leads to a situation that the phenomenon under examination has to be considered as a simplification of reality [Checkland, 1999, 51]. Checkland perceives the evolution of system thinking using a comparison to a machine [Checkland, 1999, 97]. At the time of the industrial revolution, the view about the world was based on the classic science looking the world a deterministic and "programmable" clock-like machine. System thinking was further developed through cybernetics and general system theory to look at the system as a self-regulating machine that controls its own behavior. During the industrial revolution, looking by reductive thinking, an organization was structured like a closed system. In the 20th century, the perception of the organization was based on the concept of the open system of general system theory. The organization was now seen as a changing system, which communicates with its environment and strives towards balance.

Checkland presents the concepts of hierarchy, emergence, communication, control, and especially positive and negative feedback as a basis for system thinking [Checkland, 1999, 74-92]. Mingers and White recognize three phases in the theoretical development of system thinking: the first early years from the 1920s to the 1960s, when the basic concepts were developed; the second period from the 1970s to the 1990s, when different system-based approaches were developed; as well as third, later development of Chaos and Complexity theory [Mingers and White, 2010]. They recognize the following theories and methodologies in the development of system thinking in the 21st century: general system theory and complexity theory; cybernetics; system dynamics (including learning organization); as well as the methodology of soft systems and problem solving. Pirjo Ståhle has drawn attention from the 1960s to the complexity of the system and the intrinsic ability to change [Ståhle, 2009]. She argues that systemic thinking has proceeded to investigate instead of control and stability, the chaotic and unpredictable behavior of the system, and the internal dynamics of the system.

4. DISCUSSION

We point out the idea that the mechanistic (reductive) process conception behind the modeling practice can guide the outcome and will circumscribe the essential aspects of the object being modeled. In process development, it is not enough to find out the mechanical structure of the process, as the process always involves the subjective and mental aspects of the actors. To make the model adaptive and flexible we can leave room for interpretation. For example, in standards, this has often been done so that the standard tells what should be done, but it does not tell how. The real behavior of the subjects in the process is determined by the combination of the observed process and the process being interpreted, as well as by the actors' stimuli, errors and personal preferences. Subjectivity causes uncertainty in the system of feedback chains that can be influenced by reducing deviations between the desired, official, interpreted real and observed processes [Bandinelli et al., 1995].

The concept of process and system thinking have had a strong impact on modeling practices and methods. Different formal languages and notations have been developed to describe the system, see for example [OMG, 2011]. They have evolved from the interaction between system thinking and software technology. In the 20th century, process thinking and process modeling became a tangible part of the company's strategic planning tool and the importance of modeling was emphasized. However, one can ask, whether the methods of modeling the system and in particular the practices have developed in the same way that system thinking and process philosophy have progressed? Especially when looking at pragmatic guides and literature on the process re-engineering and improvement, it seems that the approach of reductionism is very prevalent. One of the reasons might be that reductive thinking provides strong visual illustrative technology for Western thinking, and that systemic thinking based on the ideas of holism and emergence does not provide a sufficiently clear philosophy [Checkland, 1999, 97]. This raises the question of whether other disciplines offer theories that could be utilized in SE. Some examples can be shown. For example, the qualitative action research contains similar elements (interviews and data analysis, for example), which are followed in process assessment. Robert Flood notes that soft system theory has provided the intellectual basis for many of the methods used in the action research [Flood, 2010]. Another similar example is explaining the change and evolution of a research subject through process theory. The qualitative process research, which collects data by concentrating on the events of the subject and their temporal order, has emerged in many fields as a valuable research method [Van de Ven and Poole, 1995] [Van de Ven, 2007]. As a third example, we mention the evolution of the systems engineering as an interdisciplinary field of engineering science. It uses the system thinking principles to study development and management of complex systems over their life cycle. The historical background of the system engineering is partly in the soft system methodology and it has a strong process-based connection to SE [Pyster and Olwell, 2013] [Honour, 2018].

5. CONCLUSION

Process modelling is an important part of the process improvement. In this article, we wanted to emphasize the importance of system thinking and process thinking in utilizing processes. There is a risk that practical improvement will not be able to take advantage of a holistic perspective. The modeling approach is easily reductive, providing a simplified mechanistic view of the organization's processes. There is a risk to see and utilize the process description of an organization as a nearly closed system. The model can be very detailed, but it does not necessarily take into account all aspects of the activity and changing connections to the surrounding reality. The modeling method and modeler's own theoretical process concept puts the constraints on the modeling situation. Modeling involves essentially the ability of the modeler to abstract, i.e., the ability to look at the modeling object from a chosen point of view, and thereby distinguish and conceptualize the properties considered relevant to the purpose of the modeling. We suggest three interrelated process abstraction levels: the ideal conceptual level, the subjective conceptual level and the level of physical real time events. Modeling is an epistemological process that describes the aspects considered relevant to objects in the target area. In addition to abstraction, modeling requires the categorization, generalization and axiomatization of objects, as well as the choice of presentation format of the model.

REFERENCES

Ackoff, R. L. (1994) 'Systems thinking and thinking systems', *System Dynamics Review*, 10(2–3), pp. 175–188. doi: 10.1002/sdr.4260100206.

Bandinelli, S. *et al.* (1995) 'Modeling and Improving an Industrial Software Process', *IEEE Transactions on Software Engineering*, 21(5), pp. 440–454. doi: 10.1109/32.387473.

Beck, K. (2004) *Extreme programming explained - embrance change*. Boston, MA: Addison-Wesley.

Benington, H. D. (1983) 'Production of Large Computer Programs', *IEEE Annals of the History of Computing*, 5(4), pp. 350–361. doi: 10.1109/MAHC.1983.10102.

Boehm, B. (1988) 'A Spiral Model of Software Development and Enhancement', *Computer*, 21(May), pp. 61–72. doi: 10.1109/2.59.

Checkland, P. (1999) *Systems Thinking, Systems Practice: Includes a 30-Year Retrospective*. Chichester: John Wiley and Sons Ltd.

Cummings, T. G. (1978) 'Self-Regulating Work Groups: A Socio-Technical Synthesis', *The Academy of Management Review*, 3(3), pp. 625–634. doi: 10.2307/257551.

Curtis, B., Kellner, M. I. and Over, J. (1992) 'Process modeling', *Communications of the ACM*, 35(9), pp. 75–90. doi: 10.1145/130994.130998.

Davenport, T. H. and Short, J. E. (1990) 'The new industrial engineering: information technology and business process redesign', *Sloan Management Review*, (Summer 1990), pp. 11–27.

Flood, R. L. (2010) 'The Relationship of "Systems Thinking" to Action Research', *Systemic Practice and Action Research*, 23(4), pp. 269–284. doi: 10.1007/s11213-010-9169-1.

Hackman, J. R. and Wageman, R. (1995) 'Total Quality Management□: Empirical , Conceptual , and Practical Issues', *Administrative Science Quarterly*, 40(2), pp. 309–342.

Hammer, M. (1990) 'Reengineering Work: Don't Automate, Obliterate', *Harvard Business Review*, 68(4), pp. 104–112.

Harrington, J. H. (1991) 'Improving business processes', *The TQM Magazine*, 3(1), pp. 39–44. doi: http://dx.doi.org/10.1108/eb059514.

Harrington, J. H. (1998) 'Performance improvement□: the rise and fall of reengineering', *The TQM Magazine*, 10(2), pp. 69–71. doi: 10.1108/09544789810211353.

Henriques, V. and Tanner, M. (2017) 'A systematic literature review of agile and maturity model research', *Interdisciplinary Journal of Information, Knowledge, and Management*, 12, pp. 53–73. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85016155701&partnerID=40&md5=85c3683faf037fd2f2c54799d1b5bb75.

Honour, E. C. (2018) 'A historical perspective on systems engineering', *Systems Engineering*, 21(3). doi: 10.1002/sys.21432.

Houston, A. and Dockstader, S. L. (1988) *A Total Quality Management Process Improvement Model*. Available at: http://www.dtic.mil/dtic/tr/fulltext/u2/a202154.pdf (Accessed: 28 March 2017).

Keto, H. and Jaakkola, H. (2017) 'Identifying up-to-the-minute topics in software process improvement research', in Budimac, Z. (ed.) *SQAMIA 2017 - Proceedings of the 6th Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications*. CEUR-WS.org, p. 7:1-8. Available at: http://ceur-ws.org/Vol-1938/paper-ket.pdf.

Keto, H., Mäkinen, T. and Linna, P. (2011) 'Eliciting process conception contradictions in software process improvement', in *PICMET: Portland International Center for Management of Engineering and Technology, Proceedings*.

Keto, H., Palomäki, J. and Jaakkola, H. (2010) 'Towards the process-ontological modelling', in Welzer, D. T. et al. (eds) *Information Modelling and Knowledge Bases XXI. Frontiers in Artificial Intelligence and Applications*. Amsterdam: IOS Press. doi: 10.3233/978-1-60750-477-1-299.

Kuhrmann, M., Diebold, P. and Münch, J. (2016) 'Software process improvement: a systematic mapping study on the state of the art', *PeerJ Computer Science*, 2, p. e62. doi: 10.7717/peerj-cs.62.

Lane, D. C. (2016) 'What we talk about when we talk about "systems thinking"', *Journal of the Operational Research Society*, 67(3), pp. 527–528. doi: 10.1057/jors.2015.10.

Lindsay, A., Downs, D. and Lunn, K. (2003) 'Business processes - attempts to find a definition.pdf', *Information and Software Technology*, 45(15), pp. 1015–1019. doi: 10.1016/S0950-5849(03)00129-0.

Mingers, J. and White, L. (2010) 'A review of the recent contribution of systems thinking to operational research and management science', *European Journal of Operational Research*, 207(3), pp. 1147–1161. doi: 10.1016/j.ejor.2009.12.019.

OMG (2011) *Business Process Model and Notation (BPMN) Version 2.0*. Available at: http://de.wikipedia.org/wiki/Business_Process_Modeling_Notation.

Porter, M. (1985) *Competitive Advantage: Creating and sustaining superior performance*. (Republish. New York: FREE PRESS.

Pyster, A. and Olwell, D. (eds) (2013) *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.1.2.* 1.1.2. Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Available at: www.sebokwiki.org/.

Rescher, N. (1996) *Process Metaphysics - An Introduction to Process Philosophy*. New York: State University of New York.

Royce, W. W. (1970) 'Managing the development of large software systems', in *IEEE WESCON Vol. 26, No. 8*.

Rummler, G. A. and Brache, A. P. (1990) *Improving Performance - How to Manage the White Space on the Organization Chart*. 1nd edn. San Francisco: Jossey-Bass.

Scaled Agile (2018) *Scaled Agile Framework*. Available at: https://www.scaledagileframework.com/ (Accessed: 25 May 2018).

Schwaber, K. and Beedle, M. (2001) *Agile software development with Scrum*. Prentice Hall.

Schwaber, K. and Sutherland, J. (2017) *The Scrum Guide™*. Available at: https://www.scrumguides.org/scrum-guide.html (Accessed: 31 May 2018).

Shaw, M. (1990) 'Prospects for an Engineering Discipline of Software', *IEEE Software*, (November), pp. 15–24.

Ståhle, P. (2009) 'The dynamics of self-renewal: a systems-thinking to understanding organizational challenges in dynamic environments', in Bonfour, A. (ed.) *Organizational Capital: Modelling, Measuring and Contextualizing*. London: Routledge, pp. 119–146.

Takeuchi, H. and Nonaka, I. (1986) 'The new new product development game', *Harvard Bus. Rev*, vol 64(1), pp. 137–146.

Van de Ven, A. H. (2007) *Engaged scholarship: A Guide for Organizational and Social Research*. Oxford University Press.

Van de Ven, A. H. and Poole, M. S. (1995) 'Explaining development and change', *The Academy of Management Review*, 20(3), pp. 510–540. doi: 10.5465/AMR.1995.9508080329.

Wageman, R. (1997) 'Critical success factors for creating superb self-managing teams', *Organizational Dynamics*, 26(1), pp. 49–61. doi: 10.1016/S0090-2616(97)90027-9.

Wang, P. (2008) 'Whatever Happened to BPRP The Rise, Fall, and Possible Revival of Business Process Reengineering: From the Organizing Vision Perspective', in Grover, V. and Markus, M. L. (eds) *Business Process Transformation*, pp. 23–40.

Whitehead, A. N. (1929) *Process and Reality: An Essay in Cosmology*. New York: The Macmillan Co.