O'Connor, R.V. and Yilmaz, M., Exploring the Belief Systems of Software Development Professionals, journal Cybernetics and Systems, 46 (6-7), pp. 528-542, 2015.

Exploring the Belief Systems of Software Development Professionals

Rory V. O'Connor
School of Computing, Dublin City University, Ireland
roconnor@computing.dcu.ie

Murat Yilmaz
Cankaya University, Turkey
myilmaz@cankaya.edu.tr

Abstract

It is commonly accepted that an individual's beliefs and actions are based on their assessment and perceptions of the world they inhabit. In order to determine what practices an individual is likely to follow at any given time, it is necessary to understand the individual's behavioural intention in a given circumstance. From an Information Technology perspective, software development professional's belief systems are potentially the basis for the adoption and implementation of new and innovative work practices and processes. In this paper we explore the belief systems of software development professionals to understand the beliefs underlying intention and practice, and seek answers about how they adopt or reject new and innovative software development processes and practices. The results point out a strong influence of past experiences, personality types and repeated behaviour on current software development processes and practices in industrial settings.

INTRODUCTION

A significant step towards the introduction of any new and innovative work practices and processes in an organization is to understand the belief systems of the team members that can guide behaviour and inform important decisions, which may help us to explore and understand how decisions are ultimately made. In the context of managing the change in Information Technology (IT) environments, a critical objective is to have software developers use processes and techniques that are new (and, hence, innovations) for them, and which involve a change in behaviour. One option in bringing about a behavioural change is to simply acquire a new technology or process, and mandate that it be used. While such a course of action may have some value in the short term, the negative longer-term effects of the forced use of innovations have been well documented (Ram and Jung, 1996). In fact, some research reveals that managers recognize the value of influencing their staff attitude toward innovations over compelling their use (Agarwal and Prasad, 2000). The problem then becomes one of understanding how to influence innovation usage behaviour proactively without resorting to coercion. Research in social psychology as well as information systems suggests that attitude, which is an affective response to the proposed behaviour, is a determinant of intentions. If this is true, then information system managers have an alternative to compelling the use of new and innovative work practices.

Prior research suggests that individual's beliefs about using the innovation affect their attitude toward using it, although there is divergence in how this research approaches the identification of relevant beliefs. In the Theory of Reasoned Action (TRA) (Fishbein and Ajzen, 1975), the identification and measurement of relevant beliefs is a two-step process. Salient beliefs about the behaviour in question have to be elicited first from target users for each specific context to which the theory is applied. Then the strength of each of these beliefs about the behaviour is assessed and is weighted multiplicatively by the value assigned by the subject to that attribute of the behaviour. For example, a behavioural belief in the context of the adoption of specific technology might be: "*using software tools improves my productivity in software development*" while an example of the associated evaluation is: "*it is important for me to be more productive in software development*".

IT acceptance and usage represent central concerns in recent information systems research. Although several theoretical models have been proposed to describe the phenomenon associated with IT acceptance, the Technology Acceptance Model (TAM) is increasingly recognized as a robust yet parsimonious conceptualization. Drawing upon the TRA, TAM argues that IT acceptance behaviours can be explained by individual beliefs about the usefulness and ease of use of the IT. Empirical studies based on TAM have indicated that usefulness and ease of use beliefs do explain significant variance in attitude, intentions, and usage. Other work in technology acceptance, notably innovation diffusion studies, however argue for a more comprehensive set of beliefs.

There has been considerable research on the factors that predict whether individuals will accept and voluntarily use information systems. The TAM has a base in psychological research, is parsimonious, explains usage behaviour quite well and can be operationalized with valid and reliable instruments. A limitation of TAM is that it assumes usage is volitional, that is, there are no barriers that would prevent an individual from using an IS if he or she chose to do so.

Other research (such as (Rogers, 1995)) has studied the relevant literature in the adoption and diffusion of innovations and has tried to identify more comprehensive sets of beliefs or attributes of an innovation that influence user acceptance. A relevant belief identified by Tornatzky and Klein (1982) is compatibility, which is defined as "t*he degree to which an innovation is perceived as being consistent with the existing values, needs, and past experiences of potential adopters*" [p. 195].

Theoretical and empirical research on technology acceptance, while acknowledging the importance of individual beliefs about the compatibility of a technology, has produced equivocal results.

It is unclear what relationship creative style may have to software developers' beliefs and attitudes regarding an innovation. Although one could argue that less innovative employees may judge the innovation to be more difficult to use and they may offer this as an excuse for their reluctance to adopt or accept the innovation, the precise nature of the relationship between innovativeness and ease of use may not be so straightforward. As one example, Agarwal and Prasad (2000) in their examination of a domain-specific form of innovativeness - which they label personal innovativeness with IT - do not anticipate a direct relationship between personal innovativeness with IT and employees' attitudes to a technological innovation, nor do they find a direct relationship in their study.

SOFTWARE PROCESS

A software process essentially describes the way an organization develops its software products and supporting services, such as documentation. Processes define what steps the development organizations should take at each stage of production and provide assistance in making estimates, developing plans and measuring quality. The process and associated activities are often documented as sets of procedures to be followed during development. However, the documentation is not the process but should clearly represent the process as it is implemented within an organization. To simplify understanding and to create a generic framework, which can be adapted by organizations, software processes are represented in an abstract form as software process models. A number of different models exist as instantiations of how software development can be undertaken. Some of the best-known process models include: Waterfall Development, Evolutionary Development and Component-based Development (Sommerville, 2007).

There is a widely held belief that a better software process results in a better software product, with authors such as Humphrey (1995) stating that "*to improve your product, you must improve your process quality*". These ideas have led to a focus on Software Process Improvement (SPI) which can be traced back to the 1970s and 1980s and the work of Crosby (1979) and Juran (1988) who demonstrated that, in the area of production management, product quality could be improved through a better production process.

SPI aims to understand the software process as it is used within an organisation and thus drive the implementation of changes to that process to achieve specific goals such as increasing development speed, achieving higher product quality or reducing costs. The reason for this focus on SPI is encapsulated in the belief that there is an intrinsic link between the quality of the software process and the quality of the outputs emanating from that process and this belief is shared by a number of authors. Zahran (1998) claims "*it is a widely accepted fact that the quality of a software product is largely determined by the quality of the process used to maintain and develop it*". SPI models have been developed to assist companies in this regard and purport to represent beacons of 'best practice'. Contained within international standard quality models such as CMMI, ISO 9000 and ISO/IEC 29110 (Laporte et al 2014), according to their supporters, lies the road to budgetary and schedule adherence, better product quality and improved customer satisfaction.

However, the domain of SPI is not without its detractors. For example, according to the SPI manifesto (Pries-Heje and Johansen, 2010) many people believe that they don't need processes in order to build and ship software products. This belief may be the source of most resistance to change met by SPI professionals. But the fact is that you cannot create software without process. Another problem is when 'process' is seen as 'somebody else's process description.' This again leads to the misconception that one can do without process. Software should not be created without process; however, what is important is that you have process that fits the need of your projects and your business. Another belief of the practitioners is that following a process as it is defined is 'overkill'. A widespread opinion prevailed that there was an easier or less timeconsuming way to achieve an objective and many were happy to ignore their own processes to do so.

While many software process researchers understand the importance of software process management via SPI, it has been reported that in practice, managers can lack a commitment to SPI initiatives, resulting in a low process priority (Clarke & O'Connor, 2013). Furthermore, it has been reported that software SMEs tend to only implement SPI in response to negative business events (Coleman et al 2008a). These earlier studies were focused on examining the motivations for and perceptions of SPI and the findings are not particularly encouraging from a software SME perspective.

With the effective management of business processes being recognized as important for business success (Clarke et al, 2012) it follows that the effective management of the software development process is also important for business success. The findings from earlier studies suggest that software SMEs do not adopt a proactive or highly prioritized approach to SPI. This raises the problem that software SMEs may be limiting their competitive advantage by not adopting a stronger SPI priority.

The related research identified above provides an interesting and valuable insight into the role of SPI in software SMEs. However, none of this earlier research was designed to capture the full extent of the influence of the beliefs that individual key players in organizations have and its associate impact on the decision to introduce any new and innovative work practices and processes into an organization. We propose that in order to better understand this problem domain, we explore the underlying belief systems of software organization team members to better understand the factors that can guide behaviour and inform these important decisions with respect to adopting new and innovative work practices and processes, specifically SPI practices, into the software development process.

STUDY OVERVIEW

The investigation of actual software process in practice and the underlying beliefs of the key stakeholders relies heavily on eliciting and understanding the opinions of those who use the software processes in situ and the interpretation of these experiences and the reality of the situation under study. The study, therefore, naturally lends itself to the application of qualitative research methods, as they are orientated towards how individuals and groups view and understand the world and construct meaning out of their reality.

The use of qualitative research in software development studies has not been widely embraced with the Software Engineering (SE) domain. The focus on technological issues in SE studies and the associated extensive use of quantitative methods has been criticised by Bertelsen (1997) who argues for the use of qualitative research in SE. He contends that as SE is a "socio-culturally, not a technically, constituted phenomenon" any research conducted "cannot be based exclusively on natural science approaches but must include a way to understand psychological, social, and cultural phenomena". We agree with Bertelsen in believing that, to get an accurate picture of SPI in practice, one must investigate beyond purely technological factors.

Of the qualitative methodologies available, we suggest that grounded theory offers the best mechanism for achieving our objectives. Because of its interpretivist emphasis, and its ability to explain socio-cultural phenomena, grounded theory has been primarily used in the fields of sociology, nursing and psychology from the time of its establishment in the late 1960s. Since then, however, it has widened its reach into the business sector and latterly into the IS field, where it has been used to explain intentions, actions and opinions regarding management, change and professional interactions (Coleman, 2007). Silva and Backhouse (1997) support its use arguing that, "*qualitative research in information systems should be led by theories grounded in interpretive and phenomenological premises to make sense and to be consistent*". Myers (1997) believes that grounded theory has gained growing acceptance in IS research because it is a very effective way of developing context-based, process-oriented explanations of the phenomena being studied. The strengths of the method include facilitating the gathering and analysis of those human experiences and the associated interrelationships with other human actors, coupled with situational and contextual factors (O'Connor, 2012).

In this paper, we document the combined efforts of the researchers in multiple detailed studies into software process in actual practice in software development companies over an extended period. These studies have all been conducted to explore specific aspects of software process such as process formation (Coleman et al 2008b), attitudes to adoption of software standards (O'Connor & Laporte, 2014), software process knowledge management (Ryan et al, 2013), software team dynamics (Basri et al, 2011) software productivity (Yilmaz and O'Connor, 2012a), and personality types (Yilmaz and O'Connor, 2012b). Despite the varying specific emphasis in these studies, in all instances the participants were asked key foundational questions on the formation and evolution of software process within their organization and their attitudes, beliefs and opinion on both software process in general, process models and SPI activities. Accordingly common themes and patterns of behaviour and opinions can be observed and compared to provide a wider analysis of the belief systems of software organization team members to better understand the factors that can guide behaviour key decisions in relation to the adoption of innovative SPI practices.

These studies have all involved detailed interviews with senior members of company management, with job titles such as CTO, CEO and company founder, and also senior project level staff, with job titles such as project manager and software development manager. In total over 30 companies located in Ireland, UK and Turkey have participated in these studies. All companies are heavily involved in the software development sector and represent primarily software product companies and bespoke software providers. In all cases, interviews were recorded and subsequently analysed using grounded theory coding techniques (the process of breaking down interviews and observations into distinct units of meaning which are labelled to generate concepts) and memoing techniques (the on-going process of making notes and ideas and questions that occur to the analyst during the process of data collection and analysis). The resultant concepts are initially clustered into descriptive categories, which suggests an emergent theory.

STUDY RESULTS

Based on the analysis process, we have identified a series of related categories around '*Stakeholder Beliefs'* that explain the phenomena under study. Figure 1 illustrates this, showing the main categories *Stakeholder Beliefs* and the main categories that influence these beliefs: *Background* of the people involved and their *Prior Experience* in the software business and their *Personality Type*. These in turn influence the *Process Formation* in a company, the *Process Evolution* in the company and the adoption or otherwise of *Process Standards.*

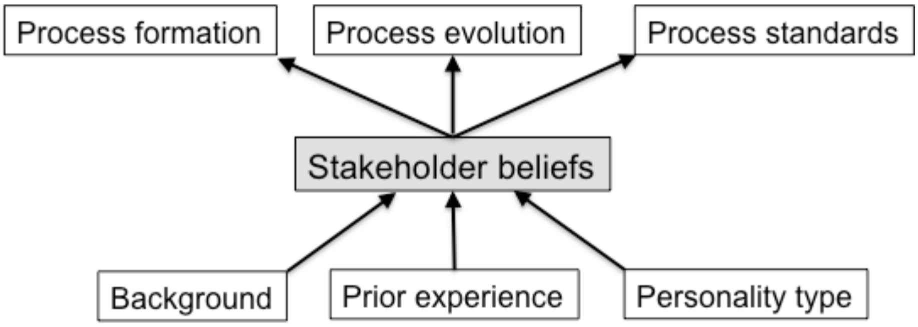

Figure 1. The overall main category diagram

One of the key theoretical themes of interest is *Process Formation*, i.e. how the software process in the company was originally formed. The findings show that how process is formed depends primarily on the beliefs of those persons who are charged with creating the initial process. In the study companies the title of the key person (or *Stakeholder*) with overall responsibility for the initial software process differed, it can be generally referred to as either 'company founder' or 'software development manager'.

Background

A key determinant in shaping the *Beliefs* of the stakeholders was their *Background*. The majority of those interviewed had previously operated as a software development manager, or similar role prior to joining (or founding) their current company. By the term *Background* we essentially mean the expertise that person has accumulated over their working and educational lives and the impact this has on the management style they cultivated in their current company. Where the software development manager had worked before and what process / process improvement model they used shaped their beliefs about the entire concept of process and drove the adoption of process used in their current company. The following extract from one company is typical of the company responses as to why a particular process model was used: "*For software development we have used the Rational Unified Process (RUP) model. The reason is that the guy we took in to head up our technology area brought that [RUP] with him*". The Chief Technology Officer (CTO) of another company also provides a representative comment on the influence of the *Background* by stating: "*In terms of technology, I'm the CTO, I was hired [in week 2 of company's existence] to build the team, build the vision and build the products... I've been involved in SPI wherever I have gone and here I make sure that the processes from day 1 are reasonable if not great*".

Closely related to *Background* is the influence it has on the management style in the company. There was a sharp diversity between the management styles adopted within the different study companies, operating at both ends of the management style spectrum. Some companies tend to be more enforcing of process allowing little deviation which with strong similarities to McGregor's (1985) 'Theory X' style. Examples of this Management Style can be seen in one company who directed their staff on why they needed to follow SPI: "*So we were telling people this [SPI] is for the growth of the company so it's for everybody's good to go along with it and embrace it*". One of the smallest companies interviewed, has a very 'hands-on' CEO who also adopts a 'command and control' management style as exemplified by his comment "*If a guy isn't delivering, we just don't want him in the company. You encourage him to leave or structure an exit for him*".

In opposition to such a Theory X style, many company managers operate a much more open regime, similar to McGregor's (1985) 'Theory Y' style. In this context, the opinions of subordinates are valued and included as part of software development policy and there is greater evidence of trust in development staff and their ability to carry out tasks with less direct supervision. Self-empowered teams and shared ownership is more associated with this style of management and was more widely deployed in companies exhibiting this style of management, as exemplified by one company who said "*If you have 1 guy working on a piece of software with 15 years experience he understands the principles of how we work. He knows what he's doing and doesn't need me interfering*". Another company typified such an open approach by saying that it was normal for "*the manager to send an email out to all his department saying we are thinking of going this way, what do you think?".*

Prior Experience

A key determinant in shaping the *Beliefs* of the stakeholders was their *Prior Experience*. All managers brought with them something less tangible, namely 'experience'. This is simply defined within this study as 'knowing what to do in a given situation'. One manager when asked about how he managed to grow the software development activity in his current organisation stated, "*I guess a lot of it is our [previous company] experience because we understood what we needed to do when we got to a certain level*". This factor was widespread across the interviews. The managers' prior knowledge, and the fact that they had encountered similar situations before, gave them a belief that they were equipped to deal with the situations they found when joining their current employers. One company appointed a number of senior development staff simultaneously. They then used the backgrounds of all of these individuals to determine their initial process: "*We sat down at the beginning and looked at what sort of environments have people worked in before, what sort of process did they have there and we tried to import them and tried to adapt them*". Essentially in almost all cases the practices used by the senior developer, created by their background experience, become the de facto initial process.

If the managers had a positive *Prior Experience* with a particular process model and they understood it particularly well, then they opted for familiarity rather than something novel. This concept of bringing a particular model or tool with them was a common feature of the managers interviewed. One company manager talking about her prior experience of setting up a software process say "*What that [the setup] experience gave me was the information as to what sort of processes I wanted to put in place and why I wanted them*".

Whilst the background and experience of the software development manager helps to form the process, prior negative experiences can also work against certain process elements. For example, in relation to the adoption of international best practice software development quality standards such as ISO9000 and CMMI (Mora et al, 2009), prior negative experience had an influence on software development manager's decisions in their current companies, as illustrated by a company founder who, when asked what working with the CMMI model was like in his previous company, responded: "*It [CMMI] was dire. It just got in people's way… It wasn't for me… I won't use it here*".

Personality Types

A key determinant in shaping the *Beliefs* of the stakeholders were *Personality types*. The term personality type can be considered as a combination of a set of characteristics that reflect our perceptions, values, and emotions. The Myers-Briggs Type Indicator (MBTI) classifies personality types via four dichotomous dimensions; extroversion-introversion (E-I), which shows the methods for an individual to draw energy (outer word versus inner word of ideas), sensing-intuitive (S-N), which refers the methods of individuals to process data (facts versus possibilities), thinking-feeling (T-F), which is related with people's decision making (objective versus subjective), and judging-perceiving (J-P) identifies whether an individual has a structural or an adaptable style to deal with the world (Yilmaz et al , 2014).

During the study, we observed that majority of interviewees agreed that there should be a visible implication of personality types on stakeholders' beliefs. Moreover, a significant number of managers perceived a relationship between stakeholders' beliefs and process formation, and selection of process standards.

Evidence suggests that there is no one-size-fits-all software process. Therefore, a software process is commonly tailored by a software development manager based on several factors such as team skills, requirements characteristics, and project needs. During the interviews managers have stated that they were very concerned about the impact of personality types of software development personnel especially during the process formation phase. One such example of this is illustrated by the following comment: "*The candidates who passed the technical interviews were asked questions to assess their social and communication skills. However, it seems personally very hard for me to select the suitable candidate by using the personality types*". A deeper insight into these issues is best exemplified in this related comment: "*In general, introverted and judging personality types are more loyal to a software process. However, it might be relatively hard for them to accept and adopt a major requirement change. In fact, they work very hard to keep the deadlines realistic. Extroverted and perceiving personalities, on the other hand, are very good with customer relations. However, they do not like to be involved in the process formation. They could easily cope with the process change but they are relatively less disciplined and eventually prefer to spend time with social activities*".

Furthermore, prior studies have noted the importance of selecting appropriate software process standards. The evidence collected during the interviews suggests that managers with introverted personality characteristics are more connected to plan-driven development approaches used in process standards. In support of this, one interviewee said: "*We live in a world where managers should be confortable under spotlights and obviously most of them are extrovert ideal. Unfortunately, I believe I have limited social skills. I have a team of 40 people who work in a regulated environment to develop medical software. As there are several guidelines and regulations we should have to follow, we mostly work without seeing customers during the development activities and ultimately I can confirm that we do not need to deal with dramatic changes in software requirements*".

The projects that required heavy customer involvement, frequently demands extroverted individuals who are confortable with people skills. An extroverted manager commented to the issue: "*I guess I have to communicate with the client very frequently. Firstly, because the requirements change rapidly and hence the requirement's document should be updated even weekly. Secondly, our competitors in the market are trying to change the client's mind by using their newly developed product, which is equipped with extra features and trendy technologies. How could I ignore the client even for a while*".

Process Evolution

As discussed above we can see that the main issue driving *Process Formation* is *Stakeholder Beliefs*. These beliefs also are the main influencing factors surrounding both the change of process over time, termed *Process* Evolution. Whilst in the overwhelming majority of study companies that *Process Evolution* does not occur in a linear fashion and is directly related to the events that the business experiences over time and impacted by erosion of practice over time, it is the human impact on and reaction to this evolution that is of most interest in the context of this paper.

Many of the study participants reported real difficulties getting the employees to follow the company's software process due to the beliefs and opinion of the employees. Although a specific process or way of working may be appropriate for the company concerned, if the employees ignore some of the process requirements then you have process erosion and a process that is divorced from the 'official' process. In some instances it was a question of winning the employees over to a new way of doing things as explained by one manager "*Getting everyone on board, getting everyone to understand the importance of it and be behind it, was quite difficult… And there's no point in having a system there if people aren't going to be 100% behind it and use it*".

Many of the companies also experience a situation whereby engineers who do not believe in and agree with the current process will actually engage in 'workarounds'. In other words they will bypass the process if they believe ('*Stakeholder Beliefs*') it is a hindrance or unsupportive of their objectives. As one manager put it "*Typically engineers will see more elegant ways of doing something and if they are not allowed do it, you won't get the buy-in*". This is typical of what is experienced by companies who report that if engineers do not believe or see the need for something, are unconvinced by the reasoning behind it, or just believe there is a better way to do it, then they either will do it a different way or not do it at all. Either way, the managers concerned will not get buy-in to the process.

An interesting observation that appears to be common is the difference between new and experienced staff in terms of *Stakeholder Beliefs* and how these affect the decision to follow a process or not.

Companies hiring graduates (with little or no experience of working in other companies and therefore no experience of process other than the individual practices they followed) found them 'easier to manage', as one CTO put it "*They are brand new out of the box and have no real understanding of other ways [of doing process]... so they generally do what they are told, without questioning*". By contrast, several companies commented that the biggest issue rested with the senior staff. Having in most instances worked in other organisations, with their associated processes, and having built up what they saw as an expertise in software development, persuading them to follow a new process, or even adhere rigidly to the existing one, was difficult. In many instances they are often the staff most hostile to process and process improvement. As one company founder noted "*The better the developer, the less likely they are to follow the process... The best ones are literally in mutiny*".

Process Standards

Although commercial SPI models and standards have been highly publicised and marketed, they are not being widely adopted and their influence in the software industry therefore remains more at a theoretical than practical level (O'Connor and Coleman, 2007). A key driving influence behind this negative perception of *Process Standards* is *Stakeholder* Beliefs motivated by both *Prior Experience* and also *Background* of those involved in decision making.

Despite significant published evidence to the contrary, there was an overwhelming belief by participants that the processes as generally described in software standards are not easy to actually tailor and implement in their organizations. In addition, the analysis also indicates that the lack of requirement from the market in general and their customer in particular has contributed to low acceptance of such standards. Examples of interviewee opinion illustrating these would be: "*In a company of our size they [standards] would not necessarily add value... we would only need more sophisticated process if we were a larger company*" and "*Our developers are busy with coding, we don't have resources to do that [standards compliance]".*

The interview analysis indicated standards were held in low regard and seen as a 'sales tool' only. Examples of interviewee opinion exemplifying this include: "*I think a lot of process in quality standard are nonsense. Some standards tell you to do XYZ steps but they are not beneficial to our business*". In stark contract a few CEOs remarked on the importance of standards commenting that "*It would be great to have them [standards accreditation] in order to have a consistent process up and running that can always be relied on*".

DISCUSSION

While the present study has confirmed a number of findings from earlier research, the significance of this study lies in the investigation of actual software process in practice and in particular the underlying beliefs of the key stakeholders. Specifically, how these *Stakeholder Beliefs* influence the formation and evolution of the software process and how these beliefs are primarily a result of the *Background* of management staff and their *Prior Experience*.

The study presented herein takes a broad viewpoint of software process and the people involved in managing and implementing it. Taking this approach to examine SPI in action, we make important new observations surrounding the belief structures of key stakeholders charged with new and innovative work practices and processes, specifically SPI.

Although the study involved accessing a variety of personnel in 30 participating organizations - and such access is very difficult to realise – the sample size is not especially large from a numerical point of view. However, a considerable depth of information has been gathered and the sample size is large when compared to similar studies in this domain. Nonetheless, a similar study incorporating a larger number of participating companies could expect to make stronger claims in relation to the generalizability of findings – and perhaps this is an area worth considering for future research.

Further research could focus on examining the nature of tacit software process knowledge. With software development being a human intensive activity, it is natural that many aspects of improvement in the method of work occur in a social manner. This includes human-centric activities such as dialogue and learning through experience. Given the considerable dependence on people in software development, it would appear to be vitally important that human aspects of software development are adequately catered for in future software development process models and frameworks.

REFERENCES

Agarwal, R., and Prasad, J., "A field study of the adoption of software process innovations by information systems professionals, "IEEE Transactions on Engineering Management, vol.47, no.3, pp.295,308, August 2000.

Basri S, O'Connor RV. A study of software development team dynamics in SPI., Proceedings of the 17th European Conference on Systems, Software and Service Process Improvement (EuroSPI 2010). Springer Verlag: CCIS 172, 2010. Heidelberg, Germany, 2011; 143-154.

Bertelsen, O.W. (1997) "Towards a Unified Field of SE Research and Practice", IEEE Software, November/December, pp 87-88.

Coleman, G. and O'Connor, R., Using grounded theory to understand software process improvement: A study of Irish software product companies, Information and Software Technology, 49 (6), June 2007, Pages 654-667.

Coleman G, O'Connor R. (2008a) Investigating software process in practice: A grounded theory perspective. Journal of Systems and Software; 81 (5): pp. 772-78

Coleman G. and O'Connor R. (2008b), An investigation into software development process formation in software start-ups. Journal of Enterprise Information Management, 21 (6), pp. 633-648.

Crosby, P.B., 1979, Quality is Free: The Art of Making Quality Certain, McGraw Hill, USA.

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

Clarke, P. and O'Connor, R.V., The influence of SPI on business success in software SMEs: An empirical study, Journal of Systems and Software, 85 (10), October 2012, pp. 2356-2367.

Fishbein, M. and Ajzen, I., Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research. Reading, MA: Addison-Wesley, 1975.

Humphrey, W.S., 1995, A Discipline for Software Engineering, Addison Wesley, Boston, MA

Juran, J.M., 1988, Juran on Planning for Quality, The Free Press, New York.

Laporte, C.Y., O'Connor, R.V.: A Systems Process Lifecycle Standard for Very Small Entities: Development and Pilot Trials. In: Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 13–24. Springer, Heidelberg (2014)

Mora, M., Gelman, O., O'Connor, R., Alvarez, F., and Macias-Luevano, J. (2009). An Overview of Models and Standards of Processes in the SE, SwE, and IS Disciplines. In A. Cater-Steel (Ed.), Information Technology Governance and Service Management, pp. 371–387, IGI Global, Hershey, PA.

Myers, M.D. (1997) "Qualitative Research in Information Systems", Management Information Systems Quarterly, Vol 21 No 2, pp. 241-242.

O'Connor and Coleman (2007) An Investigation of Barriers to the Adoption of Software Process Best Practice Models. Australasian Conference on Information Systems, p.780-788, 2007

O'Connor, R.V. (2012) Using grounded theory coding mechanisms to analyze case study and focus group data in the context of software process research. In: Mora, M., Gelman, O., Steenkamp, A., Raisinghani, M. (eds.) Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems, ch. 13, pp. 1627–1645. IGI Global.

O'Connor, Rory and Laporte, Claude (2014) An innovative approach to the development of an international software process lifecycle standard for very small entities. International Journal of Information Technologies and Systems Approach, 7 (1). pp. 1-22.

Pries-Heje, J., and J. Johansen. "SPI Manifesto." European System & Software Process Improvement and Innovation (2010).

Ram S. and Jung, H.-S., Forced adoption of innovation in organizations: Consequences and implications, J. Prod. Innovation Manage., vol. 8, no. 2, pp. 117–126, 1991.

Ryan, Sharon and O'Connor, Rory (2013) Acquiring and sharing tacit knowledge in software development teams: An empirical study. Information and Software Technology, 55 (9). pp. 1614-1624

Rogers, E.M., Diffusion of Innovations, 4th ed. New York: The Free Press, 1995.

Silva, L. & Backhouse, J. (1997) "Becoming Part of the Furniture: The Institutionalisation of Information Systems", in: A.S. Lee, J. Liebenau & J.I. DeGross (Eds.) Information Systems and Qualitative Research, Chapman & Hall, London, pp 389-414.

Sommervile, I., Software Engineering, 8 ed., Addison Wesley, 2007.

Tornatzky, L. G. and Klein, K. J., Innovation characteristics and innovation adoption-implementation: A meta-analysis of findings," IEEE Trans. Eng. Managent, vol. EM-29, pp. 28–45, Feb. 1982.

Yilmaz, Murat and O'Connor, Rory (2012a) Social capital as a determinant factor of software development productivity: An empirical study using Sstructural equation modeling. International Journal of Human Capital and Information Technology Professionals, 3 (2). pp. 40-62.

Yilmaz, Murat and O'Connor, Rory (2012b), Towards the Understanding and Classification of the Personality Traits of Software Development Practitioners: Situational Context Cards Approach, 38th Euromicro Conference on Software Engineering and Advanced Applications, September 2012.

Yilmaz, Murat, O'Connor, Rory and Clarke, Paul (2014) An exploration of individual personality types in software development. In: 21th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2014), CCIS vol. 425, pp. 111–122. Springer-Verlag.

Zahran, S. (1998). Software process improvement: Practical guidelines for business success. Boston, MA: Addison Wesley.