

Knowledge Management in Software Development: The Case of Agile Software

Indika Dissanayake

University of Texas at Arlington
indika.dissanayake@mavs.uta.edu

Ramakrishna Dantu

University of Texas at Arlington
ramakrishna.dantu@mavs.uta.edu

Sridhar Nerur

University of Texas at Arlington
snerur@uta.edu

ABSTRACT

Today's business environment is extremely dynamic and competitive. In order to sustain the pressure and gain a competitive edge, it is imperative for organizations to be creative in their software development efforts. Agile software development has huge potential for nurturing creativity. However, little research has examined creativity in the context of software development projects, particularly those using agile practices. The objective of this paper is to articulate a model that elucidates the relationship between agile practices and creativity. Further, the model tries to provide an understanding of how Knowledge Integration mediates the relationship between agile practices and team creativity.

Keywords

Software development, agile methodologies, knowledge integration, team creativity.

INTRODUCTION

It is perhaps tautological to assert that software is the lifeblood of contemporary organizations, perfectly capable of helping them leapfrog their competitors. Therefore, it is not surprising that organizations expend a lot of time and money to improve their software development processes. The proliferation of methods, tools, and techniques presents both challenges and opportunities in this regard. While these advances can mitigate some development problems, their impact on processes such as knowledge integration and learning have hardly been subjected to empirical scrutiny.

There is little doubt that software development is a complex process that frequently involves diverse stakeholders whose interests are not often aligned. In addition, the varied and, at times, conflicting perspectives of these constituents have to be reconciled to develop a software product that endeavors to provide uniform satisfaction. The crux of this effort lies in the ability of the software team to effectively create and assimilate, integrate, and deploy the relevant knowledge to satisfy not just the functional specifications of a customer, but also the myriad business rules, architectural and design trade-offs, and non-functional requirements that are often not stated explicitly. Clearly, the method employed to develop software as well as the tools and techniques available play a pivotal role in creating an environment that is conducive to the effective integration and management of knowledge. Despite the enormous strides we have made in the field of software development, there is a paucity of studies on how knowledge integration mediates the relationship between development practices and project outcomes such as team adaptation, knowledge management effectiveness, and team creativity, to name but a few.

The growing importance of agile software development (ASD) provides yet another compelling reason for investigating the issue of knowledge integration in software projects. ASD emphasizes a collaborative, team-based approach that aims to deliver high-quality software in short, iterative cycles (Highsmith and Cockburn, 2001; Highsmith, 2002). The proponents of ASD advocate flexible processes that allow self-organizing teams to thrive. In addition, ASD encourages the active involvement of customers throughout the development process. Reviews and retrospectives at the end of each iteration provide opportunities for the team to interact with stakeholders and reflect on issues related to the product as well as the process. Further, daily stand-up meetings, as suggested by most agile methods, promote the interchange of knowledge. Thus, ASD has very strong implications for how the stock of knowledge is created, exchanged, integrated, and managed within development teams. To the best of our knowledge, there is little or no research on the effect of ASD practices on knowledge integration. Given these gaps in the extant literature, our study endeavors to evolve a conceptual research model that elucidates how contemporary software development practices such as iterative development, process flexibility, and self-

organization impact knowledge integration and the consequences that follow from it. Specifically, we explore how agile principles/techniques can foster a climate that is conducive to the creation of knowledge, which in turn can facilitate software development team creativity.

The remainder of the paper is organized as follows. The next section reviews the literature that is relevant to our study. This is followed by an articulation of our research model and the propositions that ensue from it. Subsequently, we discuss the limitations of the paper and provide some directions for future research. Finally, we conclude our paper with a summary of the implications for research and practice.

LITERATURE REVIEW

Knowledge integration in agile teams

Agile methods such as Scrum, Extreme Programming (XP), Feature Driven Development, and lean software development have become increasingly popular among software developers since the articulation of the agile manifesto in 2001 (Lee and Xia, 2010). Agile practices, such as sprint/iteration reviews, encourage interactions among diverse stakeholders, including customers, system designers, developers, and management. One of the key characteristics of ASD is collaborative decision-making (Nerur, Mahapatra and Mangalaraj, 2005). Collaboration facilitates the interchange of different kinds of knowledge that members bring to the table, thereby enabling the team to make informed decisions (Robert, Dennis and Ahuja, 2008). Some of the agile principles such as collaboration, self-organizing teams, retrospective meetings, and daily stand-up meetings facilitate the creation and exchange of knowledge (Chau and Maurer, 2004). Further, the active involvement of customers and/or product owners enables team members to acquire external knowledge (Lee and Xia, 2010).

As mentioned earlier, ASD uses iterative development that involves multiple short iterations in the process of software development. These iterations help in incorporating evolving and changing customer requirements into the design and development process (Bonner, Teng and Nerur, 2010). Agile software development practices strongly emphasize frequent feedback loops. At the end of each iteration, agile team members receive feedback directly from their customers or from the other members of the team. Agile teams use multiple mechanisms to incorporate these feedbacks into subsequent iterations of the project to ensure that the end product meets the business needs (Meso and Jain, 2006). This continuous interaction and feedback process leads to knowledge integration in agile development teams.

According to Cavaleri and Obloj (1993), systems can reinvent themselves by venturing beyond their equilibrium thresholds to the “edge of chaos”. Likewise, ASD has often been described as “chaordic” (e.g., Ambler, 2002), exhibiting characteristics of chaos and order in its efforts to engender change while retaining its ability to respond to shifts in requirements. Agile practices facilitate variety in teams through increasing team diversity (customer, developers, and other stakeholders) and variety in skills through interchangeable roles. For example, in pair programming, the pairs continually switch while programming (Beck, 2000). The emphasis on collaboration, iterative development, adaptive planning, frequent reviews, and retrospectives are “variety” enhancing mechanisms that provide the requisite variety for agile teams to cope with high degrees of uncertainty (Ashby, 1968; Nerur and Balijepally, 2007). Further, in the context of agile development, variety and autonomy enhance the ability of teams to self-organize in response to contingencies that might arise during problem solving. Leadership and collaborative management style in agile teams provide team members with more freedom, flexibility, and decision making power on their job-related activities and enjoy greater level of autonomy (Mangalaraj, Mahapatra and Nerur, 2009).

Concept of “ba”

Knowledge creation and integration happen through continuous interaction between tacit and explicit knowledge (Nonaka and Von Krogh, 2009). Nonaka’s knowledge creation model identifies four stages in the knowledge creation process: Socialization, Externalization, Combination and Internalization (SECI). According to Nonaka and Konno (1998), socialization is sharing of tacit knowledge among individuals engaged in collaborative work, particularly when they are collocated. Externalization is the process by which tacit knowledge is converted to explicit knowledge. Combination refers to transfer of explicit knowledge into more comprehensive form of explicit knowledge through acquisition, integration, synthesis, and dissemination of knowledge. During the Internalization phase explicit knowledge is converted into tacit knowledge through “learning by doing”, and training (Nonaka and Konno, 1998; Nonaka, Toyama and Byosi`ere, 2003)

Nonaka’s notion of “ba” is particularly relevant to this study. According to Nonaka and Konno (1998; p. 40), “ba” is “a shared space that serves as a foundation for knowledge creation”. Further, Nonaka et al. (2003; p. 499) note that, “...the power to create knowledge is embedded not just within an individual but also within the interactions with other individuals or

with the environment. “ba” is a space where such interactions take place”. “ba” facilitates the nurturing of knowledge within teams. In a sense, “ba” refers to an environment that engenders new knowledge - associated with the individual or the group – and is largely determined by the nature of interactions and sharedness among individuals. Nonaka’s notion of “ba” provides a valuable theoretical foundation for this study. Nonaka et al. (1998) identified five factors which facilitate “ba”: autonomy, fluctuations, redundancy, requisite variety, and trust and commitment.

Providing more autonomy to the team members encourages them to share knowledge. Self-organizing teams can be used as a way to create autonomy (Nonaka et al., 2003). Fluctuations refer to changes in the routines such as new technology, and competition, market demand. Fluctuations in the work environment facilitate knowledge creation in several ways. When fluctuations occur, employees alter their routine behaviors and start questioning, and rethinking the process. This could lead to generation of new knowledge. Prigogine and Stengers (1984) refer to this phenomenon as ‘Order out of noise’ or ‘Order out of chaos’. In addition, fluctuations stimulate interactions of the team members with the environment, which in turn enhances knowledge integration and creation opportunities (Nonaka et al., 2003).

Redundancy refers to overlapping of information and responsibilities. Overlapping duties broaden the understanding of the domain among the co-workers, which in turn helps effective exchange of knowledge among them. Furthermore, redundancy facilitates job rotation, which, in turn, helps acquire additional knowledge and skills. Redundancy, particularly in technical fields, creates highly specialized generalists and breeds innovation through knowledge sharing (Nonaka et al., 2003). Redundancy helps enhance creativity and common understanding among team members through “Parallel processing” and information sharing (Morgan, 2006).

Striking a balance between order and chaos provides a strong foundation for knowledge creation. Requisite variety is seen as the means for achieving this balance. When the internal diversity is matched with diversity of the environment, the system will be more adept at dealing with environmental changes (Nonaka et al., 2003). In addition, variety facilitates emergence of multiple alternative solutions for the same problem. Hence, variety enhances formation and accessibility of new knowledge.

Finally, in order to share knowledge and facilitate knowledge integration and creation, team members must trust each other and be committed to the task at hand.

Knowledge integration

Drawing from Robert et al. (2008), we define knowledge integration in the context of software development teams as utilization of knowledge and skills from individual team members. Software development is characterized by complexity and extensive interchange of knowledge. Comprehensive knowledge about developing the entire product/process does not reside in a single individual (Robert et al., 2008); rather, team members bring a variety of knowledge and skills to bear on the problem. Hence, effective integration of this distributed knowledge is important for product development and quality decision making. Previous literature identified three mechanisms for knowledge integration: directives, organizational routines, and teams. In a highly complex and dynamic software development environment, teams play a key role in successful knowledge integration (Robert et al., 2008).

Previous researchers addressed different dimensions of knowledge integration and its impact on team performance. Tiwana and Mclean (2005) studied the effect of team composition on knowledge integration. According to them, characteristics such as team member heterogeneity, relational capital, and absorptive capacity influence knowledge integration. Robert et al. (2008) analyzed the effect of the dimensions of social capital on knowledge integration. In a study on the impact of knowledge integration on team performance, Basaglia et al. (2010) found that team climate consisting of autonomy and experimental climate leads to knowledge integration. In an information systems development context, authors found that formal (job rotation, participative decision making, autonomous teams) and informal organizational integrative practices (e.g., informal communications) are positively related to knowledge integration (Patnayakuni, Rai and Tiwana, 2007).

Team creativity

Team creativity has been studied various contexts (for a review, see Conboy et al. 2009; Couger, 1990). Amabile et al. (1996; p. 1155) define team creativity as “the production of novel and useful ideas in any domain”. Information systems projects are usually very complex and often require innovative solutions. Creative approaches are needed for various aspects of IS development, including but not limited to, planning, managing people, and implementing applications (Couger, 1990). Highsmith and Cockburn (2001) notes that agile teams follow generative rules and depend on the ingenuity of team members to solve problems. Previous scholars studied how different work environments impact team creativity. Amabile et al. (1996) have shown that work group characteristics such as diversity, mutual openness, shared commitments, and autonomy facilitate team creativity.

RESEARCH MODEL AND PROPOSITIONS

Agile software development practices stress the significance of interchangeable roles, iterative development, self-organizing teams, customer participation, and team collaboration. These aspects map on to the five “ba” characteristics of redundancy, fluctuation, autonomy, requisite variety, and trust and commitment. As mentioned earlier, Nonaka’s notion of “ba” provides a conducive environment for knowledge integration; therefore, we argue that agile practices provide the platform for knowledge integration.

Nerur et al. (2007) have developed a conceptual framework that maps agile practices to Morgan’s (2006) holographic principles. Specifically, among other things, they mapped self-organizing teams, iterative development, interchangeable roles, and minimalistic design, to the corresponding holographic principles articulated by Morgan. In this paper, we adapt this framework to map agile practices to characteristics of “ba”. Figure 1 depicts the mapping.

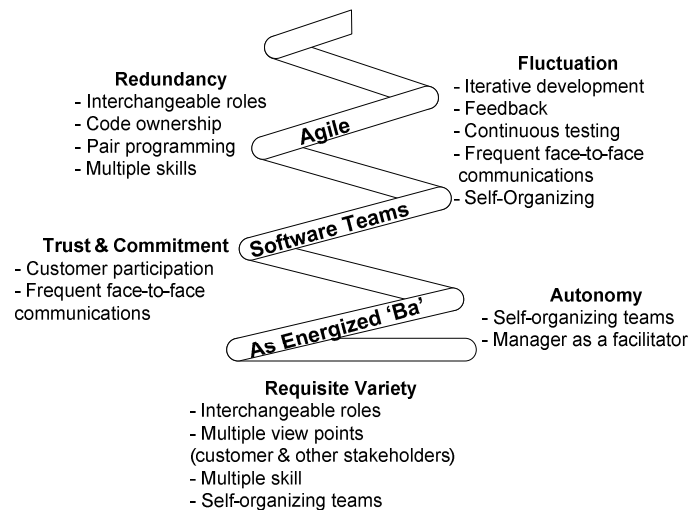


Figure 1: Alignment of agile practices with Nonaka’s notion of “ba”.

Agile practices encourage interchangeable roles. Team members are not confined to specific roles. These practices have the potential to enable team members to develop multiple skills, thus enabling them to perform other members’ roles should the circumstances demand it. In pair programming, pairs continually interchange their roles. Hence, we can see characteristics of redundancy in agile teams (Nerur and Balijepally, 2007). Over time, it is expected that agile team members will become “generalizing specialists” (for example, see Ambler 2005) with a variety of skill sets.

Agile practices encourage participative decision making. Customers are also expected to be collocated and part of the team, and actively engage in the software development process. Open spaces advocated by some of agile methods encourage face-to-face communications. These practices help in building trust and commitment among team members.

Agile teams are often described as self-organizing (Highsmith and Cockburn, 2001). Team members are empowered to make decisions. Project managers play a facilitating role, mentoring and guiding the team members without constraining their creativity while addressing the tasks. Thus, such teams have greater autonomy than do traditional teams that operate in a command-and-control mode (Nerur and Balijepally, 2007).

Agile is predicated on the assumption that requirements are volatile and subject to frequent changes. Agile practices such as iterative development and self-organizing teams enhance the team’s ability to sense and respond to such fluctuations. Further, agile teams behave as open systems and continuously interact with its environment. Continuous testing and feedback loops help incorporating evolving customer requirements in the design and development process (Bonner et al., 2010).

Previous studies have shown that knowledge integration leads to creativity (Tiwana and Mclean, 2005), better decision quality (Robert et al., 2008), and better team performance (Basaglia, Caporarello, Magni and Pennarola, 2010; Patnayakuni et al., 2007) in software development teams. Relying on Nonaka’s articulation of “ba”, we argue that agile practices such as

iterative development, self-organizing teams, and process flexibility foster a climate suitable for the generation of new knowledge.

Based on these discussions, we developed the research model shown in figure 2.

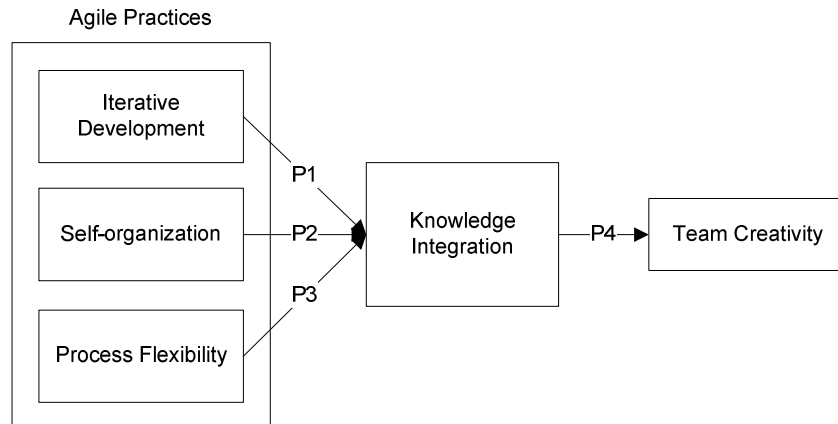


Figure 1: Research Model

Propositions

Iterations in agile development occur in 2 to 6 week cycles (Highsmith and Cockburn, 2001). These iterations facilitate continual rethinking and reevaluation of designs. Feedback is received at the end of every iteration. This provides an opportunity to reframe issues in light of new knowledge and/or insight offered by various stakeholders. Iterative development is best suited for accommodating fluctuations in the requirements or changing business needs. Thus, it is aligned with the “ba” characteristic of “fluctuation”. We have seen that fluctuation aids in knowledge integration. Hence, we propose:

P1: Iterative development is positively related to Knowledge integration

Members of self-organizing teams possess overlapping skills and are not confined to any specific roles. This enhances the diversity and skill variety of the teams. Further, team members enjoy autonomy, so they have high decision making power (Nerur et al., 2005).

Previous research has shown that practices such as job rotation, participative decision making, autonomous teams, and informal communications have a positive impact on knowledge integration (Patnayakuni et al., 2007). All these characteristics are apparent in self-organizing teams that use agile software development. Since self-organizing teams reflect “ba” characteristics of variety and autonomy, we argue that self-organizing teams lead to Knowledge integration. Thus, we propose:

P2: Self-organization is positively related to Knowledge integration

Process flexibility has been viewed in several ways in the software development literature. According to Bonner et al. (2010), “process flexibility encompasses the reaction capabilities inherent in agile processes”. Nidumolu and Knotts (1998) define software process flexibility as “the speed with which the organization's software development approach can respond effectively to changes in the organization's environment”.

Frequent interactions and heavy information flow are common in flexible environments. Process flexibility allows team members to quickly interchange their roles and adapt to changes in the environment. As argued earlier, redundancy and autonomy are likely to be higher in agile teams. Hence, we propose:

P3: Process flexibility is positively related to Knowledge integration

Previous studies have shown that knowledge integration leads to enhanced team efficiency and effectiveness (Basaglia et al., 2010); increase in the quality of decision making (Robert et al., 2008); increase in team creativity (Tiwana and Mclean, 2005); and reduction in software defects (Robert et al., 2008) .

Tiwana and Mclean (2005) have studied team creativity in the context of information systems development. In a software development context, team members from diverse backgrounds have different cognitive structures. Collaboration of these

members with different cognitive structures results in team creativity (Mumford and Gustafson, 1988). Further, they bring different ideas and expertise to the table. Exposure to alternative ideas helps looking at the problems in a completely different perspective. This in turn helps generating creative solutions. Further, knowledge integration facilitates teams to come together for a shared understanding of the project and the product. This common understanding helps them to see the big picture and come up with more creative solutions (Tiwana and Mclean, 2005). Thus, we propose:

P4: Knowledge Integration is positively related to Team creativity.

LIMITATIONS AND FUTURE RESEARCH

Our study aims to address an area of research that has received scant attention in the literature. Specifically, it endeavors to evolve a tentative model to explain the roles of knowledge integration and creativity in the context of software development. An obvious limitation of the study is that the model has not been empirically validated. This includes testing for direct relationship between agile practices and team creativity. Having said that, we would like to point out our articulation of a research model is the first step towards empirical assessment of knowledge integration and its outcomes in a contemporary software approach. Future research may build on our efforts in the following ways:

1. There are bound to be extraneous factors that affect the results of any empirical affirmation of the model. Therefore, appropriate control variables have to be identified prior to testing the model.
2. The model may be extended to include outcome variables related to team performance, such as team reflexivity, the ability of the team to adapt, team responsiveness, and knowledge management effectiveness, to name but a few.
3. This research may be extended to explore the differences between best practices suggested by agile methodologies and knowledge management. While much of our focus is on the synergies between agile methodologies and knowledge management, it would be interesting to explore the barriers that agile practices may present to the creation and dissemination of knowledge.

CONCLUSION

Complex software projects entail creative solutions to meet the demands of an increasingly turbulent business environment. Our paper argues that knowledge integration is a likely determinant of creativity. Further, we assert that agile software development practices provide the right environment and opportunities for agile teams to effectively integrate the variety of knowledge that is typically used in software projects.

In this study, we developed a framework aligning Nonaka's (2003) "ba" characteristics with agile practices. We further developed a conceptual model of agile practices leading to creativity through knowledge integration.

Our study contributes to theory and practice in the following ways. (i) It articulates a theoretically-grounded research model to investigate knowledge integration in agile teams, and (ii) Role of agile practices in promoting team creativity.

Creativity is an important aspect of software development practices. Organizations endeavor to build creative teams that can bring value. This study looks at the role of agility in fostering creativity. Specifically, we look at agile practices such as self-organizing teams, iterative development, and process flexibility that might be useful in understanding team creativity in a software development context. This could be helpful to software organizations in their efforts to build effective strategies to promote learning and to improve creativity of their software development teams.

REFERENCES

1. Amabile, T., R. Conti, H. C. and Herron, M. (1996). Assessing the Work Environment for Creativity. *The Academy of Management Journal*, 39(5), 1154-1184
2. Ambler, S. (2002). *Agile modeling: Effective practices*. John Wiley and Sons.
3. Ambler, S. (2005). Quality in an agile world. *Software Quality Professional*, 7(4), 34.
4. Ashby, W. R. (1968). Variety, constraint, and the law of requisite variety. *Modern systems research for the behavioral scientist*, 129-136.
5. Basaglia, S., Caporarello, L., Magni, M. and Pennarola, F. (2010). IT knowledge integration capability and team performance: The role of team climate. *International Journal of Information Management*, 30(6), 542-551.
6. Beck, K. (2000). *Extreme programming explained*. New York: Addison-Wesley.

7. Bonner, N. A., Teng, J. T. and Nerur, S. (2010). The Perceived Advantage of Agile Development Methodologies By Software Professionals: Testing an Innovation-Theoretic Model. *AMCIS 2010 Proceedings*. 93. <http://aisel.aisnet.org/amcis2010/93>.
8. Cavaleri, S. and Obloj, K. (1993). *Management systems: A global perspective*. Wadsworth Belmont, CA.
9. Chau, T. and Maurer, F. (2004). Knowledge sharing in agile software teams. *Logic versus approximation*, 173-183.
10. Conboy, K., Wang, X. and Fitzgerald, B. (2009). Creativity in Agile Systems Development: A Literature Review. *Information Systems—Creativity and Innovation in Small and Medium-Sized Enterprises*, 122-134.
11. Couger, J. D. (1990). Ensuring creative approaches in information system design. *Managerial and Decision Economics*, 11(5), 281-295.
12. Highsmith, J. A. (2002). *Agile software development ecosystems*. Addison-Wesley Professional.
13. Highsmith, J. and Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
14. Lee, G. and Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87.
15. Mangalaraj, G., Mahapatra, R. and Nerur, S. (2009). Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems*, 1-11.
16. Meso, P. and Jain, R. (2006). Agile software development: adaptive systems principles and best practices. *Information Systems Management*, 23(3), 19-30.
17. Morgan, G. (2006). *Images of Organization*. California: Sage Publication, Inc.
18. Mumford, M. D. and Gustafson, S. B. (1988). Creativity syndrome: Integration, application, and innovation. *Psychological Bulletin*, 103(1), 27-43.
19. Nerur, S. and Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3), 79-83.
20. Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
21. Nidumolu, S. R. and Knotts, G. W. (1998). The Effects of Customizability and Reusability on Perceived Process and Competitive Performance of Software Firms. *MIS Quarterly*, 22(2), 105-137.
22. Nonaka, I. and Konno, N. (1998). The concept of "ba": Building a foundation for knowledge creation. *California Management Review*, 40(3), 40-55.
23. Nonaka, I. and Von Krogh, G. (2009). Perspective - Tacit knowledge and knowledge conversion: Controversy and advancement in organizational knowledge creation theory. *Organization Science*, 20(3), 635-652.
24. Nonaka, I., Toyama, R. and Byosi`ere, P. (2003). *A Theory of Organizational Knowledge Creation: Understanding the Dynamic Process of Creating Knowledge*. Oxford University Press, 491-517
25. Patnayakuni, R., Rai, A. and Tiwana, A. (2007). Systems development process improvement: A knowledge integration perspective. *Engineering Management, IEEE Transactions on*, 54(2), 286-300.
26. Prigogine, I., Stengers, I. and Toffler, A. (1984). *Order out of chaos: Man's new dialogue with nature*. Bantam Books, New York.
27. Robert, L., Dennis, A. and Ahuja, M. (2008). Social Capital and Knowledge Integration in Digitally Enabled Teams. *Information systems Research*, 19(3), 314-334.
28. Tiwana, A. and Mclean, E. R. (2005). Expertise integration and creativity in information systems development. *Journal of Management Information Systems*, 22(1), 13-43.