Fall 9-11-2020

A Theory of Agile Software Development

Adarsh Kumar Kakar
Alabama State University, akakar@alasu.edu

A THEORY OF AGILE SOFTWARE DEVELOPMENT

Adarsh Kumar Kakar
Alabama State University
akakar@alasu.edu

ABSTRACT

The Agile Software Development Method (ASDM), in its present form, is guided by the Agile manifesto which consists of an Agile philosophy and a set of 12 principles. Despite the apparent effect of Agile philosophy and principles on the practice of software development around the world, neither its theoretical contribution nor its theoretical base has yet been articulated. In response to calls in literature, in this study we propose and articulate a theory of ASDM to describe and explain its effects. The theory is based on a synthesis of the key concepts underlying Agile principles and is expressed as a model of relationships. The article describes the theory formulation process and elaborates its key propositions. The limitations of the proposed theory and areas of future research are discussed.

KEYWORDS

Agile Software Development, Agile Manifesto, Agile Principles

INTRODUCTION

Agile methods represent a paradigm shift from traditional, plan-based approaches to software development (Dyba and Dingsoyr,2009). For many decades research in software engineering was preoccupied with discovering ways to deliver faster, better, and cheaper software. The contributions of academicians include formal methods for measurement and standardization of the software process and a variety of tools and techniques to aid software development. However, in the late 1990s various suggestions for improvement have come from practitioners culminating in the Agile manifesto (Fowler and Highsmith; 2001):

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

The emerging principles (Table 1) from the Agile manifesto and methods such as Extreme programming (Beck and Gamma 2002), Scrum (Schwaber and Sutherland 2007), Crystal methodologies (Cockburn 2004), Dynamic Software Development Method (Stapleto 1997) Lean Software development (Poppendieck and Poppendieck 2003) and Feature Driven Development (Palmer ad Felsing 2001) were together labeled as agile software development. This new approach has had a huge impact on how software is developed worldwide (Dyba and Dingsoyr,2009). The increasing popularity of agile methods has made it imperative that research be undertaken to discover their conceptual underpinnings and understand what makes them work.

Despite the apparent effect that the ASDM has had on the practice of software development there is little empirical research support for its effectiveness beyond anecdotal evidence. This is in part because no theory describing, explaining, and predicting the impact of the ASDM has been presented to guide the progress of the empirical researcher; neither its theoretical contribution nor its theoretical base has yet been articulated. Academic attention on ASDM is focused largely on atheoretical comparisons of Agile vs. Plan-driven methods which often generate more sound than light.

The Agile principles (Table 1) represent a complex, interrelated set of prescriptions. Although they certainly do suggest and advocate a number of concepts, they, themselves, are not concepts, the building blocks of theory

(Chafetz, 1978). The formalization of the theoretical context of the effectiveness of ASDM is essential for improved implementation of its guiding principles and for further advancement in the area of software development methodologies. Deming (2000) stated that building knowledge involves systematic revision and extension of theory based on comparison of prediction with observation. "Without theory we are just groping in chaos" (Deming, 1986). Yet the formalization and advancement of a theory of ASDM remains largely an unexplored issue (Dingsøyr, Nerur, Balijepally and Moe, 2012). This article aims to fill this gap in the atheoretical domain of software development methodologies by proposing and articulating a theory of ASDM. In the article, we first describe the process of identifying the key concepts underlying Agile principles. We then model the proposed relationships among these key concepts. The reasons for the proposed relationships are then elaborated. Propositions are postulated for the proposed relationships to enable future studies to empirically test and validate them. The article then concludes by discussing the limitations and contributions of the study.

Serial Number	Agile Principles
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4	Businesspeople and developers must work together daily throughout the project.
5	Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7	Working software is the primary measure of progress.
8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9	Continuous attention to technical excellence and good design enhances agility.
10	Simplicity--the art of maximizing the amount of work not done--is essential.
11	The best architectures, requirements, and designs emerge from self-organizing teams.
12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Table 1. The 12 Agile Principles

THEORY DEVELOPMENT PROCESS

The theory-formulation process began with an in-depth examination of what the ASDM is and how and why the effective adoption of the ASDM would lead to the achievement of salutary project outcomes. Research efforts were focused on acquiring an in-depth understanding of what the 12 principles are, how they originated, and why they have prevailed. From the agile manifesto and the agile principles 22 key concepts were extracted. The 22 concepts are highlighted in bold in the Agile manifesto and in Table 1 above. Although insightful the 22 concepts did not suit the purpose of theory building. A theory developed from 22 concepts would be too complex.

Therefore the 22 concepts were combined through further abstraction. This led to 6 higher order concepts or categories: organization culture, customer focus, internal and external cooperation, self-organizing teams, rapid iterative development, simplicity and waste avoidance, and continuous improvement. For example the higher order concept "customer focus" is abstracted from concepts in the Agile manifesto and principles such as "responding to

change", "customer collaboration" "harness change for the customer's competitive advantage", "highest priority is to satisfy the customer".

The Delphi method developed at the RAD Corporation was adopted for identifying and narrowing down the concepts from the Agile principles and the Agile Manifesto. Delphi method is intended for systematically soliciting, organizing, and structuring judgment and opinions on a complex subject matter from a panel of experts until a consensus on the topic is reached (Heimer and Rescher, 1959). 8 members of the panel were selected, 4 were from the faculty of a large university, and 4 were the industry partners of the university who participated in capstone projects for graduate students. All the 8 panel experts are professionally involved with Agile development. The members of the expert panel were verbally and in writing introduced to the task on hand.

Having generated the "What?" of a theory, the next step was to specify the relational linkages among the set of 6 concepts through a conceptual model. The conceptual model provides a visual means of mapping out the causal and/ or associated relationships in the development of a coherent theory. The unidirectional arrows in the conceptual model indicate cause-and-effect and bidirectional arrows represent correlations. These arrows resulted from a logical thought process and were supported by theories and concepts from multiple disciplines. Each possible connection between pairs of concepts was examined by the expert panel and was either included or excluded from the relations diagram. The resulting relational statements, illustrated in Figure 1, led to the following consensual theoretical statement of ASDM:

The effectiveness of ASDM arises from a collaborative organizational culture and customer focus that foster internal and external cooperation, rapid iterative development practices, simplicity and waste avoidance to provide competitive advantage to the customer, higher team morale, productivity and customer (user) satisfaction.

KEY CONCEPTS AND PROPOSITIONS

In this sect ion we juxtapose the proposed theory onto existing body of knowledge to establish the why of the proposed theory, . Insights from this juxtaposition help to explicate the reasoning process that governed the construction of this theory and should lend credibility to the proposed theoretical relations. This juxtaposition is illustrative, not exhaustive; the purpose is to demonstrate supporting or conflicting viewpoints rather than to prove or disconfirm the relationships in the proposed theory. This section begins by elaborating the concepts in the proposed theory, using appropriate literature. The four major propositions are then articulated and rationalized.

Collaborative Organization Culture

A collaborative or clan (Cameron and Quinn, 1999)) culture is an open and friendly place to work where people share a lot of themselves. It is like an extended family. Leaders are considered to be parental figures and mentors. Group loyalty and sense of tradition are strong. great importance is given to group cohesion and long-term benefits of human resources development are well appreciated. There is a strong concern for people. Teamwork, participation, and consensus are rewarded. Decision making is decentralized, and the knowledge of all the staff, customers and suppliers is shared and pooled to optimize the organization's operations and opportunities.

Agile development relies on teamwork, as opposed to individual role assignment that characterizes plan-driven development (Nerur, Mahapatra, and Mangalaraj, 2005; Kakar, 2017a; Kakar, 2017b). The development team is a self-managing, self-organizing community with a culture that emphasizes shared responsibility. Team members organize their work tasks in a way that gave them common ownership of the work product and control over how it was achieved (Dyba and Dingsøyr , 2008; Kakar, 2017c, Kakar and Kakar, 2017e). Agile development is not for everyone. It requires a collaborative working culture to be in place where the allocation of responsibility is removed from the individual status to a team culture—joint and not individual responsibility is emphasized (Cockburn, 2002; Kakar, 2017d; Kakar, and Kakar 2017f).

"Imposing agile principles on process-centric, non-collaborative, optimizing organizations is likely to fail" They operate best in a people-centered, collaborative organizational culture (Cockburn and Highsmith, 2001; Kakar, 2018a); a culture that emphasizes collaboration whether dealing with colleagues, customers or business partners. Research has shown that customer focus is challenging in organizations where the silo mentality prevailed (Hammer, 1990). It requires cooperation internally among functions and externally with business partners. This leads us to proposition1.

Proposition 1: A Collaborative organization culture promotes customer focus and an environment conducive to agile processes and outcomes

Customer focus

Agile development approach brings the marketing concept into software engineering by emphasizing the primacy of addressing evolving customer requirements. Keith's (1990) article, on the marketing concept is one of the earliest and most popular. It illustrates the adoption of the marketing concept in an applied setting. The intuitive appeal of the concept and its successful application in practice played an important role in its acceptance. The article traces the three managerial phases of Pillsbury Company's evolution culmination in what Keith calls a marketing control phase. This evolutionary process Keith from production focus to sales focus and finally marketing focus resulted in a stronger organization. The implication for the business is that this evolutionary process is the correct one for all organizations. Customer focus is a core element of the marketing concept (Rosen, Schroeder and Purinton,1998). Theodore Levittt's (1960) seminal statement of the marketing concept argued that customer needs must be the central focus of the firm's definition of its business purpose.

The needs of customers evolve continuously in response to changes in environment in which they operate. Software developers with customer focus aim *to provide competitive advantage to their customers* by acquiring the ability to address these customer demands rapidly by developing working products in quick iterations, and with minimal waste. This requires the supply function (in this case the software developer) to not recognize the traditional positions of customer and supplier. Customers too expect that its suppliers are active in their integrated search for the rooting out of all forms of waste (Womack et al., 1990; Womack and Jones, 1996). This forges close cooperation between the customer and the supplier leading us to Proposition 2:

Proposition 2: Customer focus promotes rapid iterative development, reduction in waste and internal and external cooperation

Internal/ External cooperation

Shaw (1958) asserted that internal cooperation among employees enables higher individual performance by creating mutually beneficial situations among organizational members and between organizational members and the organization as a whole. The extensive literature review by Johnson and Johnson (1989) suggests that cooperative behavior results in superior achievement under most circumstances, including different tasks and contexts.

Organizations are increasingly using cross-functional teams to enhance their competitiveness (Dumaine 1990; Kakar and Kakar, 2018b). Projects, especially non-routine projects, require cooperation of individuals drawn from various functional areas (Wind 1981). Thus, to facilitate the project implementation process, it is often necessary to first foster cross-functional cooperation (Heany, 1989; Kakar, 2018d). All players—the sponsor, customer, user, and developer—should be on the same team. Merging their different experiences and expertise with goodwill allows the combined group to change directions quickly to produce more appropriate results and less expensive designs (Highsmith and Cockburn, 2001).

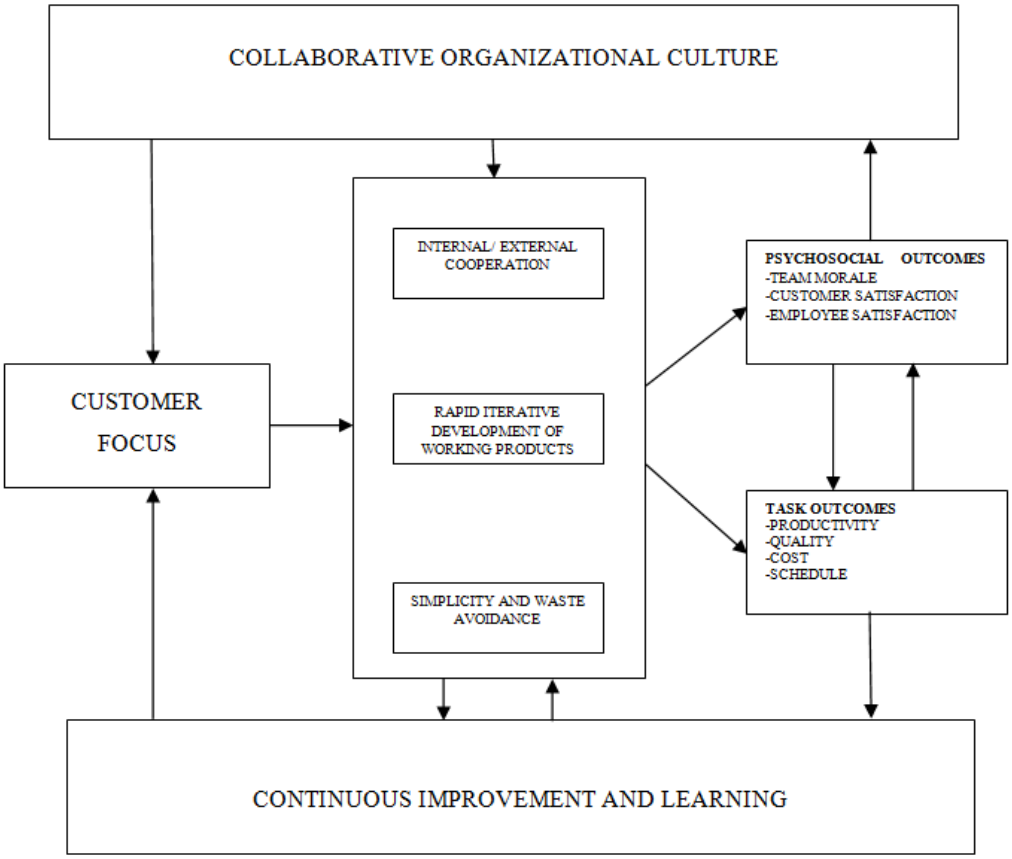

Figure 1: A Proposed Theory of ASDM

Rapid Iterative Development

Agile projects work on minimum critical specification (Nerur and Balijepally, 2007). Agile projects start with the smallest set of requirements to initiate a project and work on the principle of developing working products in multiple rapid iterations (Kakar, 2015). These working products become the basis for further discussions and the team works towards delivering the business solution using the latest input from customers, users, and other stakeholders. Users review actual working product at demonstrations instead of paper reviews or review of prototypes as in plan-driven methods. As a result the project progress is visible and the ability to decide what is to be done next is more complete, thus reducing uncertainty and giving stakeholders more confidence in the state of completion of the project.

The completion effect can be used as the basis for explaining high team member morale and user satisfaction with the iterative development approach of Agile methods. Psychological research suggests that closure, or task completion, is in and of itself a potent influence on behavior (e.g., Katz & Kahn, 1966). The closer one gets to completion the stronger is the motivation to complete a task. This has been empirically supported in various studies (Lewin, 1935; Krech, 1935; Miller, 1944; Brown, 1948; Krech, Crutchfield and Liuson, 1969). If individuals are motivated to complete what they start and if this motive gets stronger as one gets closer to completion, then project completion may be a driving force behind individuals' continuing to invest efforts in projects that are already well under way. It overcomes the costs of persistence, resulting in motivated individuals and teams working towards task closure. This in turn results in greater probability of successful project outcomes and user satisfaction.

Simplicity and avoidance of waste

A lean approach to production prescribes "the maximisation of simplicity, quality and economy" (Conboy and Fitzgerald, 2004). It represents a set of practices focused on the continuous improvement of the production process, by identifying and removing anything that doesn't add value to the customer. Lean Manufacturing started as the Toyota Production System (TPS), developed by the Toyoda (now Toyota) Motor Car Company. Today Lean

Thinking is being used world-wide in a growing number of different types of organizations. It is applied for back room work as well as at points of direct contact with customers.

The basic principles of lean manufacturing also align well with the software development (Kakar, 2014). Agile methods apply the lean approach to the overall software development life cycle. These methods focus on providing value for the customer and support requirements variability. Any activity that does not provide value to the customer is simply not undertaken. Moreover, these methods promote the cohesion of team members and developer and customer interaction (Ceschi, Sillitti, Succi and Panfilis, 2005; Kakar, 2018a; Kakar and Kakar, 2018c) in line with the predominantly people-oriented rather than process-oriented approach of agile methods. The most commonly observed benefits of lean practices include improvement in quality and productivity, reduction in manufacturing costs and reductions in customer lead time, cycle time. (Schonberger, 1982; White et al., 1999).

The above discussion on the concepts of internal/ external cooperation, rapid iterative development and simplicity and waste avoidance lead us to the third proposition:

Proposition 3: Internal and external collaboration, rapid iterative development and minimization of waste lead to improved psychosocial and task outcomes

Continuous improvement

An iterative development approach provides rapid feedback and continuous learning between the customers and the development team (Chau, Maurer and Melnik, 2003). Agile practices such as pair programming serve as a continual design and code review that helps in rapid learning and removal of defects. New code is integrated into the system as often as possible. All functional tests must still pass after integration or the new code is discarded leading to working systems increasing aligned to customer needs. Periodic refactoring of code leads to continual improvement in design. Deming believed that people inherently want to do a good job, and managers need to allow workers on the floor to make decisions and solve problems, build trust with suppliers, and support a culture of continuous improvement of both process and products (Deming, 2000). Instead of espousing rigid processes, Agile methods follow the lean manufacturing approach of creating a culture for continuous improvement, enabling processes to improve by learning from mistakes and successes (Poppendieck, 2001; Kakar, 2017a) This leads us to the fourth proposition:

Proposition 4: Feedback on task and psychosocial outcomes lead to continuous improvement and learning

CONCLUSION

In this article, a theory of ASDM is proposed and articulated in response to calls in litearture. The concepts or building blocks in the proposed theory are largely derived from a multi-disciplinary review and conceptual synthesis of the Agile Manifesto. Although not explicitly identified or stated, the Agile manifesto and principles were found to be derived from concepts in lean and agile manufacturing and are an amalgamation of theories from multiple disciplines. The proposed theory serves and fulfills several purposes from the standpoint of both academic research and practice:

Three areas for future research are readily identifiable: the first being subjecting the proposed theory to empirical examinations to see whether or not real-world data support the proposed relationships in the theory. Researchers may consider testing and extending the theory's boundaries of generalizability to various project contexts and to various countries, and to various time periods. Second, the concepts can be used for alternative conceptualizations of causal linkages among these concepts. Third, opportunities exist for both theoretical and empirical researchers to examine, in greater depth, the various linkages in the articulated theory.

From a practical standpoint, the proposed theory increases understanding of the characteristics of the ASDM. This understanding should lead to more efficient and more effective efforts at achieving ASDM's purpose of transforming and improving the software development practices. However, it is important to emphasize that the theory articulated in this is simply a first attempt at defining and articulating a theory of ASDM and has been presented for the purpose of evaluation and further development. Other researchers are encouraged to critically examine the ASDM. Through such activity, more enriched and unified theories of ASDM can be developed to further the understanding of this important and emerging phenomenon in software development.

REFERENCES

1. Gamma, E., and Beck, K. (2002). *JUnit*, Testing Resources for Extreme Programming.

2. Brown, J. S. (1948). 'Gradients of approach and avoidance responses and their relation to motivation,' *Journal of Comparative and Physiological Psychology* (41), pp. 450-465.
3. Cameron, K. S., and Quinn, R. E. (1999). *Diagnosing and changing organizational culture*. Reading: Addison-Wesley.
4. Ceschi, M., Sillitti, A., Succi, G., and Panfilis, S.D. (2005). "Project management in plan-based and agile companies," *IEEE Software* (22:3), pp. 21–27.
5. Chafetz, J. S. (1978). *A primer on the construction and testing of theories in sociology*. Itasca, IL: Peacock.
6. Chau, T., Maurer, F., and Melnik, G. (2003). "Knowledge sharing: agile methods vs. Tayloristic methods, *Procs," 12thIEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03),* IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 302–307.
7. Cockburn, A., and Highsmith, J. (2001) "Agile Software Development: The People Factor," *Computer*, pp. 131-133.
8. Cockburn, A. (2002). *Agile software development*. Boston: Addison-Wesley.
9. Cockburn, A. (2004). *Crystal clear: a human-powered methodology for small teams*. Pearson Education.
10. Conboy, K., and Fitzgerald, B. (2004). "Toward a conceptual framework of agile methods: a study of agility in different disciplines," *in: Proceedings of XP/Agile Universe*, Springer Verlag.
11. Deming, W.E. (1986). *Out of the Crisis*, Cambridge, MA, MIT Press.
12. Deming, W.E. (2000). *The New Economics: For Industry, Government, Education*, 2nd ed., MIT
13. Press, Cambridge, MA.
14. Dingsoyr, T., and Dyba, T. (2009). "What do we know about agile software development?," *Software, IEEE* (26:5), pp. 6–9.
15. Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. (2012). "A decade of agile methodologies: Towards explaining agile software development" *Journal of Systems and Software* (*85:*6), pp. 1213-1221.
16. Dumaine, B. (1990). "Who Needs a Boss?" *Fortune*, pp. 52-60.
17. Dyba, T., and Dingsøyr, T. (2008). "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50:9/10), pp. 833-859.
18. Hammer, M. (1990). "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review,* pp. 104–112.
19. Heany, D. W. (1989). *Cut Throat Teammates*, Dow Jones-Irwin, Homewood, Il.
20. Fowler, M., and Highsmith, J., 2001. "The Agile Manifesto," *Software Development* (9), pp. 28-32.
21. Highsmith, J., and Cockburn, A. (2001). "Agile software development: the business of innovation," *IEEE Computer* (34:9), pp. 120–122.
22. Johnson, D. W., and Johnson, R. T. (1989). *Cooperation and competition: Theory and research*. Edina, MN: Interaction.
23. Kakar, A. K. (2014). When form and function combine: Hedonizing business information systems for enhanced ease of use. In *2014 47th Hawaii International Conference on System Sciences* (pp. 432-441). IEEE.
24. Kakar, A. K. (2015). Investigating the penalty reward calculus of software users and its impact on requirements prioritization. *Information and Software Technology*, *65*, 56-68.
25. Kakar, A. K. (2017a). Do reflexive software development teams perform better?. *Business & information systems engineering*, *59*(5), 347-359.
26. Kakar, A. K. (2017b). Investigating the prevalence and performance correlates of vertical versus shared leadership in emergent software development teams. *Information Systems Management*, *34*(2), 172-184.
27. Kumar Kakar, A. (2017c). How do perceived enjoyment and perceived usefulness of a software product interact over time to impact technology acceptance?. *Interacting with Computers*, *29*(4), 467-480.
28. Kakar, A. K. (2017d). Assessing self-organization in agile software development teams. *Journal of computer information systems*, *57*(3), 208-217.

29. Kakar, A. K., & Kakar, A. (2017e). A General Theory of Technology Adoption: Decoding TAM from a User Value Perspective. *Southern Association of Information Systems*.
30. Kakar, A. K., & Kakar, A. (2017f). COSTS LOOM LARGER THAN GAINS: AN INVESTIGATION OF CONSUMERS'ONLINE VERSUS INSTORE SHOPPING PREFERENCES. *AMA Winter*.
31. Kakar, A. K. S. (2018a). Engendering cohesive software development teams: Should we focus on interdependence or autonomy?. *International Journal of Human-Computer Studies*, *111*, 1-11.
32. Kakar, A., & Kakar, A. K. (2018b). Assessing Shopper's Penalty Reward Calculus in Online versus Instore Shopping. *e-Service Journal*, *10*(3), 24-45.
33. Kakar, A., & kumar Kakar, A. (2018c). Is team cohesion a double edged sword for promoting innovation in software development projects?. *Pacific Asia Journal of the Association for Information Systems*, *10*(4).
34. Kakar, A. K., & Kakar, A. (2018d). IS THE TIME RIPE FOR BRANDING OF SOFTWARE PRODUCTS. *Southern Association of Information Systems*.
35. Katz, D., and Kahn, R. L. (1966). *The social psychology of organizations*. New York: Wiley.
36. Keith, R. J. (1960). "The Marketing Revolution," *Journal o f Marketing* (24), pp. 35-38.
37. Krech, D., Crutchfield, R. S., and Livson, N. (1969). *Elements of psychology (2nd ed.)*. New York, NY: Alfred A. Knopf.
38. Lewin, K. (1935). *A dynamic theory of personality*. New York, NY: McGraw-Hill.
39. Miller, N. E. (1944). "Experimental studies of conflict. In J. Hunt (Ed.)," *Personality and the behavior disorders* (1:421-465). New York, NY: Ronald Press.
40. Nerur, S., Mahapatra, R., and Mangalaraj, G. (2005). "Challenges of migrating to agile methodologies," *Communications of the ACM*, pp. 72–78.
41. Nerur, S., and Balijepally, V. (2007). "Theoretical reflections on agile development methodologies," *Communications of the ACM* (50:3), pp. 79–83.
42. Poppendieck, M. (2001) *Lean programming*, http://www.agilealliance.org/articles/articles/LeanProgramming.htm.
43. Poppendieck, M., and Poppendieck, T. (2003). *Lean software development: An agile toolkit*. Addison-Wesley Professional.
44. Rosen, D.E., Schroeder, J.E., and Purinton, E.F. (1998). Marketing High Tech Products: Lessons in Customer Focus from the Marketplace," *Academy of Market Science Review* (98:6).
45. Schwaber, K., and" Sutherland, J. (2007). What is Scrum" *URL: http://www. scrumalliance.*
46. *org/system/resource/file/275/whatIsScrum. pdf,[Sta nd: 03.03. 2008]*.
47. Schonberger, R.J. (1999). *Japanese Manufacturing Techniques: Nine Hidden Lessons in Simplicity*. Free Press, New York. White, R.E., Pearson, J.N., Wilson, J.R.
48. Shaw, M. E. (1958). "Some motivational factors in cooperation and competition," *Journal of Personality* (26), pp. 155-169.
49. Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press.
50. Levitt, T. (1960). "Marketing Myopia," *Harvard Business Review*, pp. 45-46.
51. White, R.E., Pearson, J.N., and Wilson, J.R. (1999). "JIT Manufacturing: a survey of implementation in small and large US manufacturers," *Management Science* (45:1), pp. 1–15.
52. Wind, Y. (1981). *Marketing and the Other Business Functions, in Research in Marketing*, JAI Press, Greenwich, CT (5), pp. 237-264.
53. Womack, J., Jones, D. and Roos, D. (1990). *The Machine that Changed the World*, Rawson Associates, New York, NY.
54. Womack, J., and Jones, D. (1996). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, Simon & Schuster, New York, NY.