Research Journal of Information Technology

ISSN 1815-7432

Academic
Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 3 (2): 69-80, 2011
ISSN 1815-7432 / DOI: 10.3923/rjit.2011.69.80

Directions and Advancements in Global Software Development: A Summarized Review of GSD and Agile Methods

[1]R. Akbar, [1]M.F. Hassan, [2]A. Abdullah, [1]S. Safdar and [1]M.A. Qureshi
[1]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia
[2]Department of Management and Humanities, Universiti Teknologi PETRONAS, Malaysia

Corresponding Author: Rehan Akbar, Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia

ABSTRACT

The consequences of Global Software Development (GSD) on software development processes are evolutionary. It has reshaped and reformed the traditional software development approaches. The preferences of the developers and priorities of the client have tremendously been changed. A set of new generation of processes, agile methodologies, has been appeared in response to the GSD. Many researchers have presented a good quality study on agile based methodologies. In this study we have discussed the work of the various researchers on GSD. A review of the prominent study of the researchers and practitioners on agile based methodologies is presented. The combined effects of both GSD and agile methodologies on overall software development paradigms are discussed. Based on the analysis of existing models and frameworks produced by the researchers and practitioners, the study suggests the future directions of the software development processes as required by the software industry. This study provides a guideline to the researchers and actual practitioners in formalizing and shaping up their future research works. Finally, the research findings emphasize on the need of applied and practical approaches of software development and contribution of the researchers in this regard.

Key words: Agile, client, global software development (GSD), light weight processes, software processes

INTRODUCTION

The globalization that started in the last decade of twentieth century has become more intense during recent years. Like other sectors of the society (Cho, 2007) it has affected the IT industry in great. Unlike before, many options such as the most sophisticated software development processes, tools and technologies are available to the companies. The software development teams have now different alternatives in hand. Due to their characteristics and simple rules of development and light weight processes the agile based methodologies are being preferred by the companies. In the form of agile models, the software development teams have various options such as XP, scrum, crystal and pair programming.

Ramasubbu and Balan (2007) have attributed these changes to the impacts of IT globalization. As compared to the others, the effects of IT globalization are more visible on the field of information technology. Project outsourcing to offshore teams as a consequence, has emerged as the most promising practice. Ktata and Levesque (2009) have discussed the various factors that lead to the project outsourcing.

The agile based methodologies, project outsourcing and distributed offshore software development teams are considered as the outcomes of the IT globalization. The software engineering researchers have produced a large number of good quality works in the areas such as agile methodologies, software process improvement, requirement management, software quality assurance and software project management etc. A number of revolutionary works have been produced in these areas during past several years. Other areas of theoretical computer science, artificial intelligence and software engineering have also been explored by Safdar *et al.* (2009, 2011), Qureshi *et al.* (2009) and Hassan and Robertson (2008a, b). Bioinformatics has also emerged as a new applied field of interdisciplinary research in recent years. By applying techniques of IT in the field of biological sciences, had produced much diversified type of study in this field by Tambunan and Parikesit (2010), Amir *et al.* (2010) and Nasiri *et al.* (2010). In this study, the analysis is made on the changes appeared as the consequences of GSD and how practical solutions do agile methodologies provide to face its challenges.

In this study, we have presented the review of the contributory works produced by the researchers on agile based methodologies and GSD. The GSD is discussed in a way of its impacts on software development practices. The review is presented on the very selective good quality studies published in well recognized and indexed proceedings and journals.

Based on the analysis, we have made present recommendations on the future research directions and the best practices for the software industry. The outcome of the research is to provide guidelines to the researchers and practitioners to produce more applied and industry oriented works. This study aims to provide the support to face the challenges of IT globalization. Finally, future directions according to the requirements of the software industry are suggested.

GLOBAL SOFTWARE DEVELOPMENT (GSD)-A PARADIGM SHIFT

The consequences of globalization started to appear during 1990s. Globalization not only affected the political, social, cultural and economic aspects of present society but also the engineering and technological fields, especially Information Technology (IT) (Cho, 2007). The consequences of globalization on the IT industry appeared mainly in the form of project outsourcing practices which are also termed as offshore development (Gopal *et al.*, 2002). This was the era of internet and the whole world was being realized as a global village. The developing countries such as Pakistan, China and India were providing good quality services of software development at cheap rates. It resulted into offshore development. Companies and investors (also called clients) outsourced their project to these countries to compete with the market. Project outsourcing was the advent of new generation of processes in software engineering (Akbar *et al.*, 2010). It is believed that traditional software development approaches have gradually become obsolete. Many factors such as cheap labor, skilled developers, market competition and latest tools and technologies have been identified as the main reasons behind this paradigm shift (Akbar and Hassan, 2010b).

Till the late 1990s project outsourcing had become the most common practice in the software industry. Mostly projects were being outsourced to the offshore companies. Offshore development started a new debate among the researchers regarding the processes, project management and performance issues had been discussed by Akbar *et al.* (2010), Akbar and Hassan (2010a), Rao (2009), Sterba *et al.* (2008), Taylor (2005) and Kolawa (2004). Gopal *et al.* (2002) presented a conceptual model for offshore development as shown in Fig. 1.

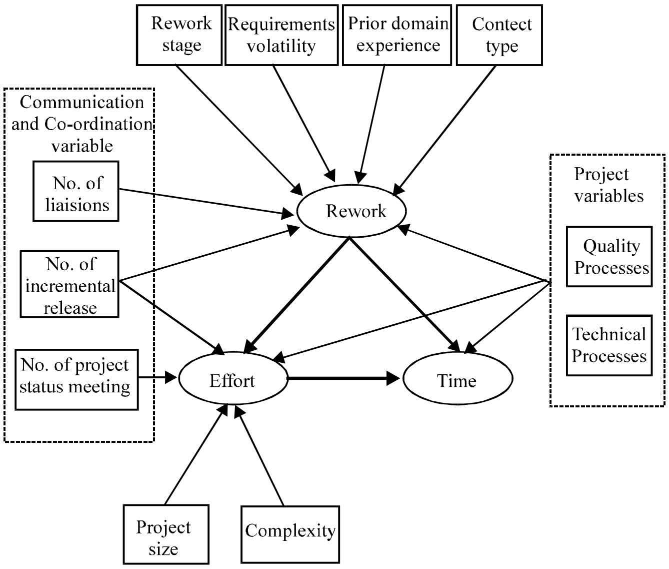

Fig. 1: Conceptual model (Gopal *et al*., 2002)

According to Fig. 1, time, effort and rework are introduced as three factors to measure project performance. The quality processes, technical processes and communication and coordination have been identified as the variables that affect the project performance. Other variables that affect the project performance are shown in Fig. 1. The effect of each variable is determined through empirical data. The results show that the communication and coordination processes greatly affect the project performance. Realizing the importance of communication and coordination processes, Akbar and Hassan (2010a), Akbar *et al.* (2011) and Gopal *et al.* (2002) for offshore development and (Succi *et al*., 1997) in general have presented and discussed interaction strategies. They have described the interactions among major key roles of a software development project. Akbar and Hassan (2010a) and Akbar *et al.* (2011) for the first time presented a formal model in this regard that highlighted the importance of communication and coordination processes in outsourced projects and distributed environments.

Besides the success of projects outsourcing, a number of certain and uncertain risks are also involved in it (Akbar and Hassan, 2010b). A number of studies are available on risks for in-house and offshore development. Various models and frameworks to avoid and prevent risks have been presented by Lu and Ge (2003), Lu and Ma (2004), Debnath *et al.* (2006) and Verner *et al.* (2008). Major works in offshore development are observed on client based risks. Risks involved in offshore development from providers perspective are categorized into vendor specific and project specific (Taylor, 2005). The vendor specific risks are related to the commercial environment while risks to projects involve client and vendor factors, software package, location and processes. Akbar and Hassan (2010b) had presented a set of measures against the risks faced by outsourced projects where teams are distributed. Such risks are mostly related to the clients requirements, frequent modifications in requirements, interaction and communication problems, expertise of team resources, time constraints for testing, language barriers, geographical time differences, infrastructure facilities, job rivalry and poor project management. Software testing is the most critical phase of software development that takes around 40-70% of the

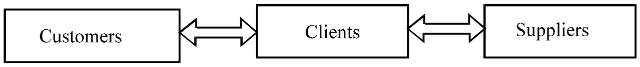

Fig. 2: Three main players in GSD (Cho, 2007)

effort, time and cost (Kosindrdecha and Daengdej, 2010) of the software team. The set of measures presented by Akbar and Hassan (2010b) to handle such risks are very practical and applied in nature.

In another study Narayanaswamy and Henry (2005) has emphasized on the importance of the role of cross cultural issues in the performance of outsourced projects. In this regard, a road map to make a governance framework for distributed software development is presented in which characteristics of organizational level standardization, project execution, planning and infrastructure are discussed (Ramasubbu and Balan, 2008). For small and medium scale organizations, Rao (2009) described common issues and their solutions in outsourced projects such as contract management, demand supply management, documentation, tool support, cultural and team level. These approaches provide significant control over such issues.

In addition to the project outsourcing and offshore development, IT globalization has also changed the concept of traditional software development by introducing the third role of client in between two existing roles of customers and suppliers as shown in Fig. 2 (Cho, 2007).

The role of client in software engineering research works, model and framework is not new for the researchers and industry practitioners. Inspite of this fact no major contribution from the researchers on this important role has seen. Management studies group though had realized the importance of this factor very early. Understanding the clients perspective (Hills, 2007) in business management group is the key element of their policies and strategies. Unfortunately, researchers in the field of IT could not produce works on this important factor. Client in the software development projects is the key role player whose satisfaction determines the project success. The spider web model is believed to be the first model in software engineering that describes how to understand and maintain the clients perspective during software development (Akbar *et al.*, 2010).

The authors have defined client's perspective in a way that how client thinks, how does he/she behave, responds, what are his/her requirements and what does he/she has in his/her mind. This definition of clients perspective is more similar to the definition of (Hills, 2007). According to the spider web model as shown in Fig. 3 and its cross section view of Fig. 4, during each iteration, the clients perspective should be maintained in each and every phase of the software development. As each iteration reaches to its acceptance point as shown in Fig. 3 and 4, it is handed over to the client. Acceptance point is the point when client approves the release of that specific build/iteration after confirming the completion of all the requirements and testing.

The spider web model is the new generation process model of client based software development and emphasizes on client or customers satisfaction. The model supports the agile rules. Client centric approaches are best suitable for the environments where clients exactly know what they want and even if they have no idea of what they want (Olberding, 2007). In this study, Chung *et al.* (2010) have referred customer's satisfaction as a part of overall organizational management activities.

In short, as the consequences of global software development, a paradigm shift from the traditional heavy weight approaches to the light weight agile based methodologies has been reported by many researchers. Since late 1990s till to-date the response to agile methodologies is

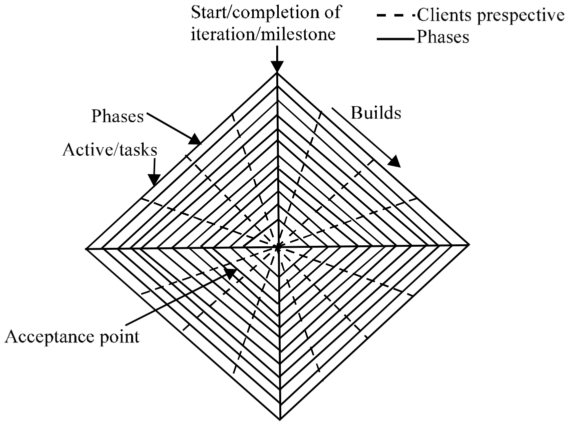

Fig. 3: The spider web model (Akbar *et al.*, 2010)

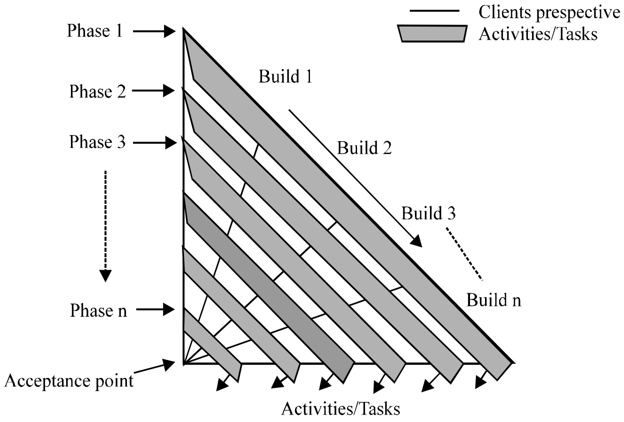

Fig. 4: The cross-section view (Akbar *et al.*, 2010)

quite encouraging and still increasing. The sole reason behind this was the style of development emerged as a result of outsourcing projects to offshore teams.

AGILE METHODOLOGIES-LIGHT WEIGHT PROCESSES

As a result of Global Software Development (GSD), the light weight agile based methodologies appeared as a new generation of software development processes. The agile based methodologies are characterized by their light weight processes, quick and fast development pattern, short iterations and focusing on the completion of working code.

Numerous process models and frameworks of agile processes have been produced by a large number of researchers and practitioners. Therefore, to standardize such works in February, 2001, an agile manifesto (http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-of-agile-software/) comprising of the following twelve rules were formed (Turk *et al.*, 2002):

- Present highest priority is to satisfy the customer through early and continuous delivery of valuable software

- Welcome changing requirements, even late in development. Agile processes harness change for the customers competitive advantage
- Deliver working software frequently, from a couple of weeks to a couple of months with a preference to the shorter timescale
- Business people and developers must work together daily throughout the project
- Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation
- Working software is the primary measure of progress
- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely
- Continuous attention to technical excellence and good design enhances agility
- Simplicity-the art of maximizing the amount of work not done-is essential
- The best architectures, requirements and designs emerge from self-organizing teams
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

All the models and framework that claim to be agile must need to qualify these twelve principles of agile manifesto. This was a remarkable achievement that led to the more standardized and goal oriented agile based approaches.

To stay competitive in the market, the clients/customers prefer to launch their products early in the market. This requirement of the clients keeps the developers under continuous pressure (Aoyama, 1998; Cusumano and Yoffie, 1999). As reported by Akbar *et al.* (2010), the ultimate requirement of the client is the completion of working code only. The support to release the working code early is provided in agile models as mentioned earlier by Turk *et al.* (2002). Based on agile principles, the agile developers believe in minimum or no documentation of the software development project, unlike traditional approaches. The principles of agile manifesto have started debates between supporters of agile methodologies and their critics. Two schools of thought are prominent in which one supports the agile based development and the others do not (Turk *et al.*, 2002; Theunissen *et al.*, 2003; Nerur *et al.*, 2005). Two agile based methodologies Extreme Programming (XP) (Beck and Fowler, 2000; Fraser *et al.,* 2000) and Scrum (Schwaber and Beedle, 2001; Rising and Janoff, 2000) among others are widely used by the developers due to the available support of fast paced development. Theunissen *et al.* (2003) discussed agile methodologies in connection with ISO standards. Software engineering researchers have produced a number of models and frameworks on agile based methodologies. Like other approaches, agile methodologies also have some limitations with respect to the different environments and project requirements. Turk *et al.* (2002) identified that agile processes provide limited support to the:

- Geographically distributed development environments
- Subcontracting
- Reusable components development
- Large team structures
- Quality control in safety-critical software
- Large and complex software systems

The disadvantages and benefits of agile processes that are claimed by Turk *et al.* (2002) are based on a set of assumptions. Therefore, their existence and non-existence in different environments may vary. As mentioned earlier that there is always a debate between agile supporters and traditional approaches supporters. In this regard, Nerur *et al.* (2005) has compared traditional approaches with agile based mainly on the factors such as control, project management style, roles of team members, formal and informal communication, role of client, process model and technology etc. The control of project, project management and role of client are considered as more important and critical in agile as compared to traditional approaches. Glass (2001), Turner and Jain (2002), Fritzsche and Keil (2007), Paetsch *et al.* (2003) and Eberlein (2003) presented the similarities and differences between both approaches. It is strongly believed that all software engineering methodologies have limitations (Basili, 1996). In his framework (CHAPL) to understand the relationship between both traditional approaches and agile based methodologies, Jiang and Eberlein (2008) concluded that both have common philosophical origins and are technically compatible and complementary to each others. Therefore, a reasoning framework to determine the suitability and selection of software development processes for different kinds of development environments is required (Basili, 1996).

In addition to the other factors, the selection of software engineering methodologies also depends on size of the company, project the size and resources. Limitations of like financial, human etc., bounds the small companies to adopt light weight agile processes. Large companies with a large number of resources prefer more standardized heavy weight approaches. Most of the software development companies irrespective of their sizes have web applications development projects works. As compared to large companies, small companies face the uncertain situations many times during software development life cycles. Tarawneh *et al.* (2007) presented a five step model of the guidelines for the web application development in small software development companies. Light weight agile methodologies have proven to be the most result oriented methodologies for small as well as large companies. In developing countries about 80% companies are small and medium sized. Agile methodologies are equally beneficial for the company of any size because of being considered more success oriented. Agile methodologies for example Extreme Programming (XP) and Scrum have proved their worth in all kinds of environments. Aiken (2004) on overview of XP and has made recommendations on how pair programming, a form of XP, can be implemented. The development time of web based applications has remarkably been reduced to few months due to such advancements in technology and methodologies. The requirement gathering and management is the most critical phase of software development. Clear, understandable, precise and well managed requirements are critical for project success, client's satisfaction and tasks allocation. Austin and Devin (2003) believed that meeting requirements is necessary for projects but is not a mandatory condition. Many researchers believe that poor requirement engineering is the main reason of project failure. A framework for requirement analysis as a part of requirement engineering was proposed by Bastani (2007). The author has modified the existing methods and reproduced them into new model. The agile approaches due to their quick and light weight processes and maintaining minimum documentation have also been proved beneficial for web based development and requirement engineering. In this regard, the agile hypertext design method has been presented by Wills *et al.* (2007). In another study, Souza and Falbo (2005) have proposed an agile based standard architecture for web application development. In a South African empirical study of 59 projects, results show that agile practices are significantly beneficial in process improvement and project success which ultimately leads to the satisfaction of stakeholders (Ferreira and Cohen, 2008).

As believed by Basili (1996) that all software engineering methodologies have limitations, it is also believed that there always exists a margin for improvement process. Agile based methodologies have proven their applicability in the software industry. Still a lot of study needs to be done by the researchers and actual industry practitioners to make further improvements and extensions in agile models according to the ultimate requirements of the software industry.

The contribution of both academic researchers and industry practitioners is required in the form of joint research projects. It would help in bringing the industry best practices and problems in the research focus and providing solutions for them to face the challenges of GSD.

GSD-IN THE LONG RUN

From last twelve years GSD has profoundly given the new directions to the software development. The evolution in software development processes is gradually taking place since years (Zhang *et al*., 2010). It has introduced numerous new practices, trends and standards of software development. From centralized co-located team structures, it has distributed them to the geographically remote locations. This is just the beginning. As distributed infrastructures are getting more popular, in the coming days portable software development would be the next era of software development. The travel that started with the project outsourcing to offshore teams, due to the limitations of resources and infrastructure constraints, portable development would be the next step. The product of GSD in the form of agile processes has already decreased the dependency on formal processes and resources. The agile model due to its limited support in particular conditions would lead to boundary-less, portable kind of development behaviors. The most advanced ways of interaction and communication on the other hand have also finished the need of co-located teams and office space. Though big giants in software development have started portable development but in coming years it would be a common practice of majority of the companies. Especially small and medium scale organizations would prefer it due to limited resources to save cost. The software engineering researchers need to extend their research works on agile to provide support for such kind of portable development environments.

CONCLUSION AND FUTURE WORK

GSD or IT globalization has introduced numerous new approaches of software development. The traditional rules of development have been changed. Agile processes have been emerged as incremental, short iterative in nature and quick releases methodologies. GSD has shortened the geographical distances. Advanced ways of communication are available. The beginning of the project outsourcing to geographically distributed offshore teams has completely reshaped the existing practices. Inspite of this, to meet the rapidly changing trends, the more applied and practical approaches and standards for software development processes are required. Applied and sustainable processes are required to solve the industry issues. Though contribution from the researchers in this regard has been quite good but it is not fulfilling the requirement of the software industry. In its current form, it is unable to provide solutions to the industry problems. More concrete efforts both by the software engineering researchers and actual practitioners are required.

Agile software development processes have proven their worth in the software industry but still the procedures of requirement engineering, interaction communication and distributed development need to be further improved and formalized. For large, complex and critical systems, an improvement in the existing agile based methodologies is required. A dynamic framework for filtering the best agile based practices in various conditions is required. In this regard we are working on a framework to provide support in agile for distributed offshore development. As we have observed, agile processes would gain more response of the researchers and practitioners in coming days.

REFERENCES

Aiken, J., 2004. Technical and human perspective on pair programming. ACM SIGSOFT Software Eng. Notes, 25: 1-14.

Akbar, R. and M.F. Hassan, 2010a. A collaborative-interaction model of software project development: An extension to agile based methodologies. Proceedings of International Symposium in Information Technology (ITSim), June 15-17, Kuala Lumpur, pp: 1-6.

Akbar, R. and M.F. Hassan, 2010b. Limitations and measures in outsourcing projects to geographically distributed offshore teams. Proceedings of the International Symposium in Information Technology (ITSim), Jun. 15-17, Kuala Lumpur, pp: 1581-1585.

Akbar, R., M.F. Hassan, M.A. Qureshi and S. Safdar, 2011. Structured role based interaction model for agile based outspresentced IT projects: Client's composite structure. Inf. Technol. J., (In Press)

Akbar, R., M.F. Hassan, S. Safdar and M.A. Qureshi, 2010. Client's perspective: Realization as a new generation process for software project development and management. Proceedings of 2nd International Conference on Communication Software and Networks, Feb. 26-28, Singapore, pp: 191-195.

Amir, A., M.A. Siddiqui, N. Kapoor, A. Arya and H. Kumar, 2010. *In silico* molecular docking of influenza virus (PB2) protein to check the drug efficacy. Trends Bioinform., (In Press).

Aoyama, M., 1998. Web-based Agile software development. IEEE Software, 15: 56-65.

Austin, R. and L. Devin, 2003. Beyond requirements: Software making as art. IEEE Software, 20: 93-95.

Basili, V.R., 1996. The role of experimentation in software engineering: Past, current and future. Proceedings of the 18th International Conference on Software Engineering, Mar. 25-29, Berlin, Germany, pp: 442-449.

Bastani, B., 2007. A requirement analysis frame study for open systems requirements engineering. ACM SIGSOFT Software Eng. Notes, 32: 1-19.

Beck, K. and M. Fowler, 2000. Planning Extreme Programming. Addison-Wesley, New York.

Cho, J., 2007. Globalization and global software development. Issues Inform. Syst., 8: 287-290.

Chung, Y.C., Y.W. Hsu and C.H. Tsai, 2010. Research on the correlation between implementation strategies of TQM, organizational culture, TQM activities and operational performance in high-tech firms. Inform. Technol. J., 9: 1696-1705.

Cusumano, M.A. and D.B. Yoffie, 1999. Software development on internet time. IEEE Comput., 32: 60-69.

Debnath, N.C., R. Uzal, G. Montejano and D. Riesco, 2006. Software projects leadership: Elements to redefine risk management scope and meaning. Proceedings of the IEEE International Conference on Electro/information Technology, May 7-10, East Lansing, pp: 280-284.

Eberlein, A., 2003. Requirements engineering and agile methods: can they benefit from each other. Position Statement in the Proceedings of Canadian Invited Workshop on Scaling XP/AgileMethods, Banff, Canada.

Ferreira, C. and J. Cohen, 2008. Agile systems development and stakeholder satisfaction: A South African empirical study. Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT research in Developing Countries, Oct. 6-8, South Africa, ACM, pp: 48-55.

Fraser, S., K. Beck, W. Cunningham, R. Crocker, M. Fowler, L. Rising and L. Williams, 2000. Hacker or hero? - Extreme programming today. Proceedings of the Conference on Object-Oriented Programming, Systems, Languages and Applications, (OOPSLA'00), MIneapolis, pp: 5-7.

Fritzsche, M. and P. Keil, 2007. Agile methods and CMMI: Compatibility or conflict. e-Inf. Software Eng. J., 1: 9-26.

Glass, R.L., 2001. Agile versus traditional: Make love not war. Cutter IT J., 14: 12-18.

Gopal, A., T. Mukhopadhyay and M. Krishnan, 2002. The role of software processes and communication in offshore software development. Commun. ACM, 4: 193-200.

Hassan, M.F. and D. Robertson, 2008a. Addressing the brittleness of agent interaction. Lect. Notes Artifi. Intell., 5357: 241-221.

Hassan, M.F. and D. Robertson, 2008b. Constraint relaxation approach for over-constrained agent interaction. Lect. Notes Artifi. Intell., 5351: 668-677.

Hills, J., 2007. Putting yourself in the client's shoes: A little-appreciated skill that can boost the success of HR people. Hum. Respresent. Manage. Int. Dig., 15: 35-38.

Jiang, L. and A. Eberlein, 2008. Towards a framestudyfor understanding the relationships between classical software engineering and agile methodologies. Proceedings of the 30th International Conference on Software Engineering, May 10-18, Leipzig, Germany, pp: 9-14.

Kolawa, A., 2004. Outsourcing: Devising a game plan. ACM Queue, http://queue.acm.org/detail.cfm?id=1036501

Kosindrdecha, N. and J. Daengdej, 2010. A test case generation process and technique. J. Software Eng., 4: 265-287.

Ktata, O. and G. Levesque, 2009. Agile development: Issues and avenues requiring a substantial enhancement of the business perspective in large projects. Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering, May 19-21, Canada, pp: 59-66-10.1145/1557626.1557636.

Lu, X. and Y. Ge, 2003. Risk analysis in project of software development. Proceedings of the Engineering Management Conference on Managing Technologically Driven Organizations: The Human Side of Innovation and Change, Nov. 2-4, Albany, New York, pp: 72-75.

Lu, X.N. and Q.G. Ma, 2004. Risk analysis in software development project with owners and contractors. Proc. IEEE Int. Eng. Manage. Conf., 2: 789-793.

Narayanaswamy, R. and R.M. Henry, 2005. Effects of culture on control mechanisms in offshore outsourced IT projects. Proceedings of the 2005 ACM SIGMIS CPR Conference on Computer Personnel Research, Apr. 14-16, Atlanta, Georiga, pp: 139-145.

Nasiri, J., A. Haghnazari and M. Alavi, 2010. Evaluation of prediction accuracy of genefinders using mouse genomics DNA. Trends Bioinform., (In Press).

Nerur, S., R. Mahapatra and G. Mangalaraj, 2005. Challenges of migrating to agile methodologies. ACM. Commun., 48: 72-78.

Olberding, C., 2007. Modes of client interaction: A proposal. http://www. stationfpresent .com /blog/post/2007/11/modes-of-client-interaction-a-proposal.aspx

Paetsch, F., A. Eberlein and F. Maurer, 2003. Requirements engineering and agile software development. Proceedings of the 12th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Jun. 09-11, Linz, Austria, pp: 308-308.

Qureshi, M.A., F.B. Hassan, S. Safdar and R. Akbar, 2009. A O (| E |) time shortest path algorithms for non-negative weighted undirected graphs. Int. J. Comput. Sci. Inform. Secur., 6: 40-46.

Ramasubbu, N. and R.K. Balan, 2007. Globally distributed software development project performance: An empirical analysis. Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium of the Foundations of Software Engineering, Sept. 3-7, Croatia, pp: 125-134.

Ramasubbu, N. and R.K. Balan, 2008. Towards governance schemes for distributed software development projects. Proceedings of the 1st International Workshop on Software Development Governance, May 12, Leipzig, Germany, pp: 11-14.

Rao, N.M., 2009. Challenges in execution of outsourcing contracts. Proceedings of the 2nd India Software Engineering Conference, Feb. 23-26, ACM, Pune, India, pp: 75-79.

Rising, L. and N.S. Janoff, 2000. The Scrum software development process for small teams. IEEE Software, 17: 26-32.

Safdar, S., M.F. Hassan, M.A. Qureshi and R. Akbar, 2009. Biologically inspired execution frameworkfor vulnerable workflow systems. Int. J. Comput. Sci. Inform. Secur., 6: 47-51.

Safdar, S., M.F. Hassan, M.A. Qureshi and R. Akbar, 2011. Effective methods for secure authentication in workflows under intrusion threat. Inform. Technol. J., (In Press).

Schwaber, K. and M. Beedle, 2001. Agile Software Development with Scrum. 1st Edn., Prentice Hall, New Jersey.

Souza, V.E.S. and R.D.A. Falbo, 2005. An agile approach for web systems engineering. Proceedings of the 11th Brazilian Symposium on Multimedia and the Web, Dec. 05-07, Pocos de Caldas-Minas Gerais, Brazil, AC., pp: 1-3.

Sterba, C., T. Grechenig and M. Pazderka, 2008. Outsourcing as a strategy for IT harmonization: A public sector case study proposing an approach in independent stakeholder scenarios. Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance, Dec. 1-4, Cairo, Egypt, pp: 245-250.

Succi, G., L. Benedicenti, P. Predonzani and T. Vernazza, 1997. Standardizing the reuse of software processes. ACM Standard View, 2: 74-82.

Tambunan, U.S.F. and A.A. Parikesit, 2010. *In silico* design of drugs and vaccines for dengue disease. Trends Bioinform., (In Press).

Tarawneh, H., A. Elsheikh and S. Lahawiah, 2007. Web-based applications development in small firms. Proceedings of the 6th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems, Feb. 16-19, Corfu Island, Greece, pp: 75-79.

Taylor, H., 2005. The move to outspresentced IT projects: Key risks from the provider perspective. Proceedings of the 2005 ACM SIGMIS CPR Conference on Computer Personnel Research, Apr. 14-16, Atlanta, Georgia, pp: 149-154.

Theunissen, W.H.M., D.G. Kourie and B.W. Watson, 2003. Standards and agile software development. Proceedings of the 2003 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology, Sept. 17-19, South Africa, pp: 178-188.

Turk, D., R. France and B. Rumpe, 2002. Limitations of agile software processes. Proceedings of 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering, May 2002, ACM, USA., pp: 1-4.

Turner, R. and A. Jain, 2002. Agile meets CMMI: Culture clash or common cause? Proc. Extreme Programm. Agile Methods - XP/Agile Universe 2002. LNCS., 2418: 153-165.

Verner, J., J. Sampson and N. Cerpa, 2008. What factors lead to software project failure? Proceedings of the 2nd International Conference on Research Challenges in Information Sciences, Jun. 3-6, Marrakech, pp: 71-80.

Wills, G.B., N. Abbas, R. Chandrasekharan, R.M. Crowder and L. Gilbert *et al.*, 2007. An agile hypertext design methodology. Proceedings of the 18th Conference on Hypertext and Hypermedia, Sept. 10-12, Manchester, England, UK., pp: 181-184.

Zhang, X., T. Hu, H. Dai and X. Li, 2010. Software development methodologies, trends and implications: A testing centric view. Inform. Technol. J., 9: 1747-1753.