What Are the Main Characteristics of High Performance Teams for Software Development?

Alessandra C. S. Dutra[1], Rafael Prikladnicki[1] and Tayana Conte[2]
[1]*Department of Computing, Pontifícia Universidade Católica do RS (PUCRS), Porto Alegre, Brazil*
[2]*Department of Computing, Universidade Federal do Amazonas (UFAM), Amazonas, Brazil*

Keywords: Software Engineering, High Performance Teams, Training, Education, Systematic Literature Review.

Abstract: This paper presents a discussion in relation to current training approaches to software development and their relation to high performance team formation. We performed an ad hoc literature review about training approaches in Software Engineering and a systematic literature review about the characteristics of high performance software development teams. Based on what was found we reflect on the challenges of training high performance teams for software development projects and to what extent the current training approaches overcome such challenges.

1 INTRODUCTION

The software development market operates in a global environment, with rapid changes, and needs to respond to these new opportunities and new markets with agility (Sommerville, 2010).

A study done in 2010 by Standish Group (Standish Group, 2013) with a sample of 10,000 projects around the world produced a report called "Chaos Manifesto 2011", which revealed that the Information Technology (IT) industry faces several challenges; although 37% of the IT projects have been successful, being delivered before the deadline and within the estimated cost; 42% of the IT projects were delivered after the deadline and more expensive than the original plan; and 21% of the IT projects were total failures, being cancelled before the delivered time, or were delivered but never used. Faraj and Sambamurthy (2006) say that improving the productivity and quality of projects are important. Initial approaches were focused on discovering better methodologies and tools, but there is an increasing perception that the projects also face several challenges related to communication, coordination, learning, negotiation, diversity and on how to form high performance teams for software development projects.

This context indicates that the qualified education and training of professionals is more necessary in the society in which we live. Beckman et al (1997) say that, among other factors, the quality of the professional is directly related to the quality of the education he/she received.

The quality of Software Engineering (SE) training can contribute meaningfully to improvements in the state of the art of software development and aid in solving some traditional problems and crises related to software industry practices (Gibbs, 1994). Nowadays, training and capacity-building to prepare a software professional must include not only basic knowledge of the Computer Science field, but also the teaching of concepts, processes and techniques for the definition, development and maintenance of software (ACM/IEEE, 2008).

As a result, the education process in Software Development has begun to question the methods used in training activities (Beckman et al., 1997). Recent studies observe that these methods involve traditional teaching strategies such as theory presentation, expositive classes and complementary reading, with which the students find in the industry a different scenario than what is taught in academia (Prikladnicki et al., 2009). At the same time, software development projects have required high performance team training, and professionals with strong technical, behavioural, and business skills which current educational programs are not able to supply (Monsalve et al., 2011). One of the reasons could be the fact that such programs concentrate on basic education focused on the traditional

C. S. Dutra A., Prikladnicki R. and Conte T..
What Are the Main Characteristics of High Performance Teams for Software Development?.
DOI: 10.5220/0005375601450152
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 145-152
ISBN: 978-989-758-097-0

approaches for software development, instead of preparing the professional to act as a part of a software development team, which requires multifunctional competencies and a multidisciplinary environment.

Thus, the goal of this paper is to develop a reflection about how the current existing SE training approaches cover the various high performance teams characteristics. We first conducted an ad-hoc literature study about the existing training approaches in SE and then a systematic literature review (SLR) about high performance teams characteristics. At the end, we reflected on how the existing training approaches help in forming high performance software development teams.

This paper is divided into six sections. In Section 2 we present the theoretical foundations. In Section 3, we report on existing training approaches. Section 4 provides a systematic literature review of high performance teams. In Section 5 there is a discussion on training versus high performance teams characteristics. Finally in Section 6 the conclusions and future work are addressed.

2 BACKGROUND

2.1 Software Engineering Training

Software Engineering is concerned with theory application, knowledge and practice for the effective and efficient software development of systems that satisfies users requirements (ACM/IEEE, 2008). SE began to be discussed as a discipline in 1968 (ACM/IEEE, 2004) and currently is part of the curriculum of several courses such as Computer Science, Computer Engineering, Information Systems, Automation Control Engineering and Software Engineering.

Software Engineering is related with all software production aspects, from the initial stage to its maintenance, involving not only technical development processes, but also project management activities and tools, methods and theories that support its production (Sommerville, 2010). Therefore, SE goes beyond programming code creation; it tries to discipline development and brings to software development principles, techniques and knowledge to discuss quality questions, deadlines and economic factors (ACM/IEEE, 2004).

The professionals who conclude their undergraduate course, according to curricular recommendations, are able to, among other aspects, master knowledge and abilities that are part of the SE area; work individually or as part of a team to develop software artifacts with quality; design solutions using appropriate SE approaches that integrate ethical, social, legal and economic questions; know how to apply current theories, models and techniques that provide a baseline for identifying and analyzing problems, software design, development, implementation, verification and documentation; demonstrate understanding and appreciation of the importance of negotiation, efficient work habits, leadership, and good communication with stakeholders; and learn new models, techniques and technologies as soon as them emerge (ACM/IEEE, 2004).

By analyzing the curricular recommendation listed, we have identified that there are several required competencies for a SE professional. The SE curriculum (ACM/IEEE, 2004), (ACM/IEEE, 2008) points to the necessity of education apart from expositive class formats, and one of the way to increase education quality involves innovative strategies and didactics. According to Beckman (1997), educational quality is one of the important factors that influence the quality of the professionals. Thus, some of the challenges for improving SE education are: to make SE courses more attractive to students; to focus appropriately on SE education, understanding its dimensions; to present industry practices to the students; provide education to industry professionals; to make education in SE evidence-based; to ensure that SE educators have the necessary experience and knowledge to this assignment; and to increase the research prestige and quality of the educational SE (Sommerville, 2010).

According to Conn (2002), the SE professionals are dissatisfied with the lack of training of the university students that enter the work market, which means that the industry must complement their education with training that gives them necessary knowledge in order to make up this deficiency. This training can involve professionals or teams, including high performance teams.

2.2 High Performance Teams

A high performance team is a group that brings together members committed to the mutual growth and personal success (Moscovici, 2003). According to Chiavenato (2008), the main high performance teams attributes are: participation, accountability, clarity, interaction, flexibility, focalization, creativity and quickness. The participation in a team

increases the commitment and the fidelity of the people, resulting in delivery of high quality work (Cleland and Ireland, 2000).

A high performance team, besides all the requirements of a team, must have its members to be committed to the personal growth and success of each team member. Such a team will exceed the performance of all the other teams and achieve results above expectations (Moscovici, 2003).

Katzenbach and Smith (1993), present some characteristics of high performance teams: "Deeply personal commitments of each one to the growth and the success of the others is what distinguish high performance teams from the majority of the existing teams. Energized by this extra sense of commitment, the high performance team typically reflects a vigorous amplification of the fundamental teams characteristics: deeper sense of purpose, more ambitious performance targets, a more complete approach, more fullness in mutual accountability, knowledge interchangeably and complementarity."

Boyett and Boyett (1998) mention some companies that have achieved great results with high performance teams. The AT&T Credit Corporation has used high performance interfunctional teams in order to improve its efficiency and service to the client.

According to Raj (Raj et al., 2006), there is a major difficulty for an organization in disseminating high performance team practices, such as work reorganization, professional involvement in decision making processes and improvement in workers's skills, despite the evidence that organizations invest in these as practices to achieve greater productivity and efficiency.

3 TRAINING APPROACHES IN SE

Training in SE should prepare the students in both theory and effective participation in a collaborative and interdisciplinary environment. In this regard, it is important to consider the variation in training techniques.

Traditional approaches in SE training are considered to be (Anastasiou and Alves, 2004):

1. Dialogued Expositive Classes: This is a content exposition, with active participation by the students, whose previous knowledge must be considered and can be taken as a foundation.
2. Text Study: This is an exploration of an author's idea from the critical study of a text and/or information research and the author's ideas exploration.
3. Directed Study: This is study under guidance and direction by the professor, aiming to solve specific difficulties.
4. Use of a Discussion List: This is an opportunity for group of people be able to debate, at a distance, a theme in which they are experts or have done a previous study.
5. Verbalization and Observation Groups (VG/OG): This is an analysis of theme/problem under a professor's coordination that divides the students in two groups: one for verbalization (VG) and the other for observation (OG).
6. Seminar: This is a space where a group discusses or debate themes or problems.
7. Case Study: This is the detailed and objective analysis of a real situation that needs to be investigated and that is challenge for the people that are involved.
8. Workshop: This is the gathering of a small number of people with common interests, which aims to study and work for the knowledge and deepening of a theme, under expert orientation.

These alternative approaches can help students learn more effectively. Alternative approaches are considered to be (Prikladnicki et al., 2009) (Gresse and Shull, 2009), (Monsalve et al., 2011), (Halma, 2009):

1. Group Activities, distance education and practice activities: By using this approach, interaction with the students is emphasized through icebreakers that explore specific subjects. The characteristics are: diversification in the techniques for group activities; practical classes in laboratories; the planning of the student work; and part-time classes: 20% of the discipline is done through distance education.
2. Capstone projects and practices activities: a Capstone project is an approach where a student group plans and executes a software project from the beginning to the end during one whole semester.
3. Playgroup and games: For this strategy, related content is first presented to the class. In the end, in order to consolidate comprehension, a playgroup is performed using LEGO®. The game makes it possible to design, from the defined requirements, a product to be built that is similar to the software development.
4. Games and educational simulators: Because of the need for training students in the SE process,

one of the alternatives is the use of games to fill the gap between theoretical and practical aspects. From the reports found in the literature (Monsalve et al., 2011), it was noticed that the majority of the proposals developed are associated with simulator games.

The approaches that are more focused on the students and that promote their further active participation on the classes, for example with games and simulators (Monsalve et al., 2011), (Halma, 2009), have the potential to increase the students interest, motivate them and improve learning at level of concept application.

4 SLR: HIGH PERFORMANCE TEAMS

The purpose of the SLR was to select the main studies in the literature that report from high performance software development teams and identify its characteristics.

4.1 Research Question and Context

The research reported in this article was guided by a two research questions: (1) *What is the concept of high performance teams in software development*? (2) *How are high performance teams characterized?*

This article is actually part of a broader research project that aims to generate a deep understanding of high performing teams in software engineering, by revisiting the definition of high performance teams, and identifying contextual conditions in which teams are likely to flourish. Thus, answering this research question is a cornerstone towards the development of comprehensive models for training and developing effective software engineering teams.

4.2 SLR Protocol

4.2.1 Inclusion and Exclusion Criteria

The papers included in the search were related to high performance teams training, characteristics and environments. We searched for papers available on the web, with the complete text in electronic format for reading as well published in a conference or a journal in the Computer Science field.

We have excluded papers that did not involve software development process and software engineering, did not deal with training of software development teams, and were not written in Portuguese or English.

4.2.2 The String Search

Similar to the Salleh study (Salleh et al., 2011), the database used as the research reference selected was SCOPUS because to its reputation and the greater numbers of abstracts and citations. The search string used was formed with the following composition:

1. "*high performance team*" OR "*high performance teams*"
2. "*performance teams*" OR "*team performance*"
3. "*teams performance*" OR "*high productivity team*"
4. "*high productivity teams*" OR "*good team*" OR "*best team*" OR "*team productivity*"
5. "*limitation"* OR *"practice"* OR *"characteristics"* OR *"environment"* OR *"organization"* OR *"concept"* OR *"productivity"*
6. "*software development*" OR "*software engineering*"

The final string received the following combination:

1 OR 2 OR 3 OR 4 AND 5 AND 6

The data extraction form was developed with the following fields: Paper, Year, Author; Conference (where published); Type (Journal, Conference); Objective; Context (Education, Practice, Tools); Contributing, evidences; Research methodology; Status (Relevant or not relevant); Justification (status regarding); Answers Question 1(Yes or No?); Answers Question 2 (Yes or No?).

4.3 SLR Execution

After we defined the research protocol, the review was executed. The initial search was made in phase 1 and returned only 3 papers. Because of this small quantity of papers, we decided to search for synonyms that match the research question, so for phase 2, we selected 112 papers. The synonyms were: productivity, productivity teams, high productivity, team performance, best team, good team, organization and concept. In phase 3, the 112 papers were classified as relevant or irrelevant, based on reading the title and abstract, and 61 papers were relevant.

In the final phase, all the papers selected in the previous phase were downloaded from the web, fully read and added to a read form implemented with MS Excel according to the protocol. In phase 4, after the reading, 20 papers did not answer the research question of this review and were removed,

so that finished with 41 papers.

4.4 SLR Results

Question 1: What is the concept of high performance teams in software development?

In the papers searched on the SLR, we found studies that present high performance team characteristics that focus on how to increase their productivity. Staples and Cameron (2005), describes how team performance is associated with characteristics such as: appropriate interpersonal skills, low team turnover, appropriate team size so that the resources are enough to complete the tasks, showing strong team spirit, and creation of innovative ways to coordinate the team, helping to achieve their tasks.

In our research, we identified some characteristics that high performance teams must have for software development. We identified organizational, behavioral and technique characteristics. Those most cited are presented in table 1 and are mainly behavioral characteristics.

Thus, we can suggest that the high performance teams (1) have an effective communication, (2) present a diversity that stimulates learning and innovation, (3) have cohesion, motivation, leadership and coordination, in order to achieve their goals.

Table 1: Characteristics most cited in the studies.

Characteristics most cited	Papers
Efficient Communication	[2][6][7][27][13][28]
Coordination	[5][6][23][28]
Team Work	[2][6][7][28]
Team Diversity	[15][17]
Leadership	[2][27]
Team Cohesion	[2][19]
Motivation	[10][13]

Question 2: How are high performance teams characterized?

For this research question, 48 characteristics were cited by the researchers. According to Table 1, Communication was the most cited, with 6 papers, next, with 5 papers, was Coordination and with 4 papers was Team Work.

Hause (2005) in the final results of his paper writes that he found the following characteristics: High performance teams were more focused on specific tasks, were more organized in their work, thus, they communicated less, had to make fewer decisions, and thus worked fewer hours, shared better information and had fewer conflicts; had a leadership style more appropriate for team work; and had a balance between communication, participation and work division.

According to Klimoski and Zukin (1999), the growth in knowledge, abilities, and skills needed to solve the tasks that create competitive advantages in today's organizations makes it impossible for individuals to work independently of teams.

In their systematic review of empirical studies on agile software development, Dybå and Dingsøyr (2008) cite Robinson and Sharp (2004) who characterized the agile development teams as a team that has faith in its own abilities and shows respect and responsibility that establishes the truth and that preserves quality of life at work.

Regarding communication, its importance to team work is evident, according to related studies on this issue (Da Silva et al., 2013), (Jiang et al., 2012) (Fernández-Sanz, and Misra, 2012), (Hause, 2005), (Hause et al., 2003), (Staples and Cameron, 2005). One must highlight Hause's research (2003), explain that the difference between high and low performance groups is measured by the amount of produced communication, with low performance groups producing more communication. Still, the analysis of their work process suggests that is not the quantity, but the quality of communication that is important in the determination of performance.

The characteristics of the teams were classified, based on Capability Maturity Model Integration CMMI (SEI, 2006), where technical competencies are the abilities to use tools, data and required process by a project or process. Organizational abilities, meaning the behavior regarding organizational structure, roles and responsibilities; and Contextual Abilities, which are abilities in self-management, communication and inter-relationship. Table 2 presents all the characteristics we found for high performance teams.

5 DISCUSSION

The reflection on existing training approach and high performance team characteristics for software development led to a necessity and an opportunity. There is a necessity for adopting alternative approaches for forming high performance teams in SE, and an opportunity to use them in undergraduate and graduate classes at Universities.

Considering the high performance team characteristics most cited, we can identify that the majority of the alternative training approaches have focus on the improvement of these characteristics such as teamwork, communication, leadership and

motivation (Prikladnicki et al., 2009).

We also identified, that at an organizational level, little relation is seen between high performance team characteristics and training approaches. From a behavioural viewpoint, characteristics such as leadership, communication, teamwork, motivation, cohesion, and flexibility are characteristics that can be associated to some of the training approaches found. The characteristic related to technical competencies are easier to be worked with current training approaches, given that technical competencies are the aspect most worked on with current training. Therefore, in an initial reflection, we understand that it is important to map training approaches in relation to the high performance team characteristics in software development.

By analyzing some of the approaches in relation to high performance teams characteristics, we can observe that: (1) Verbalization Groups (VG) and Observations Groups (OG), workshops and alternatives approaches, have the goal of developing skills such as teamwork and communication; (2) group activity approaches, distance education and practical activities (Prikladnicki et al., 2009) enable the student to work with characteristics such as teamwork, communication, and responsibility, as well as students' motivation in relation to the work done, (3) expositive classes focus more on the content.

Although the professor asks the students questions, and they interpret and discuss the study object, this approach does not work on team, leadership and communication aspects; Capstone projects and practical activities, icebreakers and educational simulators can benefit the training on communication, teamwork, leadership and organization, along with team activities.

In that regard, and considering this reflection, we have some evidence that: (1) it is important to understand what high performance teams are in terms of software development and their characteristics, (2) it is necessary to define the training approaches based on what one wants to teach, and not only from the approaches that one already know how to teach. In terms of research opportunities, we also identify: (1) the need for mapping between training approaches and high performance team characteristics. Such a study would facilitate the approaches professors choose in relation to the teams characteristics that he/she wishes to work on, in this case a focus on high performance, (2) the opportunity to propose a methodological approach that is aimed at educating high performance teams in SE.

We also identified the following challenges: (1) to be able to identify, in a software development team, the characteristics that one wishes to train; (2) to work on training of professors in order to, through innovative approaches, better prepare them and their

Table 2: Characteristics classified for high performance teams.

High Performance Teams' Characteristics					
Organizational Characteristics	**Contextual Characteristics**			**Technical Characteristics**	
Team Diversity [15][17][2]	Team Work [2][6][7][28][3]	Communication [2][6][7][27][31][28]	Motivation [10][13]	Coordination [5][6][23][28]	Managerial Involvement [38]
Team size [23][28]	Team Leadership [2][27]	Team Cohesion [2][19]	Unexpected Challenges [1]	Professional Orientation [13]	Restriction of External Influence [38]
Team's autonomy [15][28]	Personality [13]	Improvisation [1]	Attitude [13]	Teamwork Orientation [13]	Performance Evaluation [38]
Work less hours [27]	Organization [27]	Respect [11]	Passion to Teach [11]	Focus on Specific Tasks [27]	Competencies of Management [13]
Organizational Commitment [5]	Comprehension [4]	Empathy [4]	Better sharing Information [27]	Experience in Propagation [38]	Usage of Resources [6]
Life quality at work [11]	Accountability [11]	Emotional Intelligence [4]	Believe on own abilities. [11]	Knowledge [13]	
Low Turnover [28]	Flexibility [4]	Cognitive Work / Abilities [13]	Tasks Participation [27]	Less Decision Made [27][31]	
	Intelligence [10]	Less tendency to conflicts [27]	Confidence [11]	Work Tasks Division [27]	
	Analytic [11]	Socialization [10]	Awareness [10]	Goals Fixing [32]	

students to form high performance software development teams.

6 CONCLUSIONS

In this paper we presented a discussion about the current training approaches to software development and their relation to high performance team formation. As any other empirical study, this study has some limitations. The first is related to researcher bias during the paper analysis process. For this reason, two researchers were involved in the systematic literature review execution, both in paper selection and data extraction. The study on existing training approaches also had research bias during the study process as a limitation.

As a next step we intend to identify the practices of high performance software development teams in light of existing training approaches, aiming at proposing ways for developing such practices, involving existing or new training approaches, and thus contributing to the formation of high performance teams for software development.

REFERENCES

ACM/IEEE, 2004. Software Engineering Curriculum. Guidelines for Undergraduate Degree Programs in Software Engineering.

ACM/IEEE, 2008. Computer Science Curriculum, Guidelines for Undergraduate Degree Programs in Software Engineering.

Anastasiou, L. G. C., Alves, L. P., 2004. Teaching Strategies. In: Proceedings of education at the university. Strategies work in the classroom. 3. ed. Joinville: Univille, p. 67-100 (in portuguese).

Beckman, K., Coulter, N., Khajenouri, S., Mead, N., 1997. Collaborations: Closing the industry–academia gap. IEEE Software 14 (6), pp. 49–57.

Boyett, J.H, Boyett, J.T., 1998. The Guru Guide-The Best Ideas of the Top Management Thinkers. New York: Wiley.

Chiavenato, I., 2008. People management: the new role of human resources in organizations. Rio de Janeiro: Elsevier, 3a edition (in portuguese).

Cleland, D. I., Ireland R. L., 2000. Project Manager`s portable handbook. New York: McGraw-Hill, 257p.

Conn, R. 2002. Developing Software Engineers at the C-130J Software Factory. IEEE Software, Los Alamitos, v. 19, n. 5, p. 25-29.

Da Silva, F.Q.B., França, A.C.C., Suassuna, M., De Sousa Mariz, L.M.R., Rossiley, I., De Miranda, R.C.G., Gouveia, T.B., Monteiro, C.V.F., Lucena, E., Cardozo, E.S.F., Espindola, E., 2013. Team building criteria in software projects: A mix-method replicated study. In: Journal Information and Software Technology.

Faraj, S., Sambamurthy, V., 2006. Leadership of information systems development projects. In: IEEE Transactions on Engineering Management.

Fernández-Sanz, L., Misra, S., 2012. Analysis of cultural and gender influences on teamwork performance for software requirements analysis in multinational environments. In: Journal IET Soft.

Gibbs, W., 1994. Software's chronic crisis. Scientific American 271 3, pp. 86–95.

Gresse, V. W. C., Shull, F., 2009. To Game or Not to Game?. Software, IEEE, v. 26, n. 2, p. 92-94.

Halma, A., 2009. Robomind.net – Welcome to Robomind.net, the new way to learn programming. http://www.robomind.net Access in: set. 2014.

Hause, M.L., 2005. Distributed team performance in software development. In: Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education.

Hause, M., Petre, M., Woodroffe, M., 2003. Performance in international computer science collaboration between distributed student teams. In: Proceedings - Frontiers in Education Conference.

Jiang, L., Carley, K.M., Eberlein, A., 2012.Assessing team performance from a socio-technical congruence perspective. In: International Conference on Software and System Process, ICSSP 2012 – Proceedings.

Katzenbach, J. R, Smith D. K., 1993. The Wisom of Teams. Summarized by permission of Harvard Business School Press Copyright by McKinsey & Company, Inc. 275 pages.

Klimoski R., Zukin L.N., 1999. Selection and staffing for team effectiveness. In: E. Sundstrom (Ed.), Supporting Work Team Effectiveness, Jossey-Bass, San Francisco, pp. 63–91.

Monsalve E., Werneck V., Leite J., 2011. Teaching Software Engineering with SimulES-W. Conf. on Software Engineering Education and Training (CSEE&T).

Moscovici, F., 2003. Teams work right: Multiplication of Human Talent. Rio de Janeiro: José Olympio, 8a edition (in portuguese).

Prikladnicki R., Albuquerque A., Wangenheim C., and Cabral R., 2009. Teaching Software Engineering: Challenges, Teaching Strategies and Lessons Learned in FEES - Education Forum in Software Engineering (in portuguese).

Raj, P.P, Baumotte A.C.T, Fonseca D.P.D, Silva, L.H.C.M., 2006. Project Human Resource Management. Rio de Janeiro: Editora FGV – Fundação Getúlio Vargas, 180p. (in portuguese).

Robinson H., Sharp H., 2004. The characteristics of XP teams. In: Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science, vol. 3092, Berlin.

Salleh, N., Mendes, E., Grundy, J., 2011. Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. In: IEEE Transactions on Software Engineering,

37(4):509–525.
SEI, 2006. CMMI® for Development, Version 1.2. CMU/SEI-2006-TR-008 ESC-TR-2006-008. Pittsburgh, PA Software Engineering Institute-SEI, Carnegie Mellon University: 561.
Sommerville, I., 2010. Software Engineering, 9a edition. Pearson Prentice Hall.
Staples, D.S., Cameron, A.F., 2005. The effect of task design, team characteristics, organizational context and team processes on the performance and attitudes of virtual team members. In: Proceedings of the Annual Hawaii International Conference on System Sciences.
The Standish Group, "Chaos", http:// www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf, access in: June 2013.
Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. In: Journal of Science Direct.

APPENDIX

Due to the page limit, the list of all 41 references of the SLR can be accessed here: http://goo.gl/mkDi6O.

[1] Magni, M., Maruping, L.M., Hoegl, M., Proserpio, L., 2013. Managing the unexpected across space: Improvisation, dispersion, and performance in NPD teams. In: Journal of Product Innovation Management.
[2] Da Silva, F.Q.B., França, A.C.C., Suassuna, M., De Sousa Mariz, L.M.R., Rossiley, I., De Miranda, R.C.G., Gouveia, T.B., Monteiro, C.V.F., Lucena, E., Cardozo, E.S.F., Espindola, E., 2013. Team building criteria in software projects: A mix-method replicated study. In: Journal Information and Software Technology.
[3] De Melo, C.O., S. Cruzes, D., Kon, F., Conradi, R., 2013. Interpretative case studies on agile team productivity and management. In: Journal of Information and Software Technology.
[4] Günsel, A., Açikgöz, A., 2013. The Effects of Team Flexibility and Emotional Intelligence on Software Development Performance. In: Journal of Group Decision and Negotiation.
[5] Chen, P.-C., Chern, C.-C., Chen, C.-Y., 2012. Software project team characteristics and team performance: Team motivation as a moderator. In: Proceedings - Asia-Pacific Software Engineering Conference, APSEC.
[6] Jiang, L., Carley, K.M., Eberlein, A., 2012. Assessing team performance from a socio-technical congruence perspective. In: International Conference on Software and System Process, ICSSP 2012 – Proceedings.
[7] Fernández-Sanz, L., Misra, S., 2012. Analysis of cultural and gender influences on teamwork performance for software requirements analysis in multinational environments. In: Journal of IET Software.
[8] Staats, B.R., 2012. Unpacking team familiarity: The effects of geographic location and hierarchical role. In: Journal of Production and Operations Management.
[9] Maheshwari, M., Kumar, U., Kumar, V., 2012. Alignment between social and technical capability in software development teams: An empirical study. In: Journal of Team Performance Management.
[10] Georgieva, K., Neumann, R., Fiegler, A., Dumke, R.R., 2011. Validation of the model for prediction of the human performance. In: Proceedings - Joint Conference of the 21st International Workshop on Software Measurement, IWSM 2011 and the 6th International Conference on Software Process and Product Measurement, Mensura.
[11] Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. In: Journal os Science Direct.
[12] Czekster, R.M., Fernandes, P., Sales, A., Webber, T., 2010. Analytical modeling of software development teams in globally distributed projects. In: Proceedings - 5th International Conference on Global Software Engineering, ICGSE.
[13] Siau, K., Tan, X., Sheng, H., 2010. Important characteristics of software development team members: An empirical investigation using Repertory Grid. In: Journal of Information System Journal.
[14] Ganesh, M.P., Gupta, M., 2010. Impact of virtualness and task interdependence on extra-role performance in software development teams. In: Team Performance Management.
[15] Lee, G., Xia, W., 2010. Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. In: MIS Quarterly: Management Information Systems.