A Systematic Literature Review on the Technical Criteria for the Evaluation of Software Development Methodologies

Seyed Mohammadbagher Jafari
College of Farabi,
University of Tehran, Tehran, Iran
e-mail: sm.jafari@ut.ac.ir

***Abstract*—Nowadays, due to increasing demands for developing commercial software, software development has become an important issue and several methods have emerged to produce software. However, Software Development Methodologies' (SDMs) selection has always been a challenging task for developers. Moreover, the software industry faces the need for more realistic and practical software development methods to develop high-quality software products, paying attention to the allocated budget and resources in projects. The technical criteria that takes SDM technical suitability and adaptability for the project and organization into account are essential things that need to be borne in mind. Unfortunately, only a few studies are available to pay attention to this issue. This study by using systematic literature review approach, seeks to identify the technical criteria for SDMs evaluation. As a result, this study identified 15 technical criteria comprised of 56 indicators, including different technical aspects of SDMs. The results provide valuable insights into the evaluation of SDMs' technical aspects for developers.**

***Keywords*— Software Development Methodology, Evaluation Criteria, Technical Criteria, Systematic Literature Review.**

I. INTRODUCTION

Nowadays, due to increasing demands for developing commercial software, software development has become an important issue and several methods have emerged to produce software according to user requirements, overall features of the system, managerial and economic situation, timing and quality level. Software development is channeling users or customers' demands into a software product [1]. Software Development Methodology (SDM) selection has always been a controversial issue. One of the problems is the management weakness in defining criteria in order to select a suitable SDM and evaluate its different elements [2]. Although it is largely believed that SDM acceptance improves the software development process, still SDM selection encounters resistance from one part of developers [3]. Defining criteria so as to consider the SDM acceptance and technical aspects create a new vision of SDM acceptance and technical efficiency in organizations. Some organizations defined their special formally documented procedures and rules as an SDM. Others rely on unofficial agreements among developers for the development process. These studies support the notion that formal SDM acceptance usually increases productivity and quality. Despite the growing enhancement in the emergence of commercial SDMs over the past decades, SDM usage rate is low in practice in software organizations. Moreover, even organizations using an SDM do not follow them precisely. Prior studies have noted the importance of two aspects affecting SDM acceptance and rejection. One of them is the technical aspect considering SDM suitability for the organization and its needs. The second one is SDM social competency defined in terms of organizational culture, attributes, etc. As a result of using SDMs which are technically ill-suited to the project, or socially to the team, regardless of spending huge resources and expenses, developers consider SDM as useless and do not accept it [4]. Some researchers suggest their own evaluation criteria. But their criteria are usually limited to their own vision and experience concerning methodologies. Moreover, these criteria are most of the time incapable of being practically measured or their reliability and validity are not calculated [5]. Software development mainly depends on developers' own experience and expertise. These kinds of developing approaches have resulted in wrong definitions of user demands, weak controlling and management of the project, and ultimately customer dissatisfaction [6]. Most of the organizations develop their own customized methodologies to improve their project management and planning procedures as well to standardize their production and development process. Bubenko [6] claims there are 1000 methodologies whereas Jayranta [7] states their number is more than thousands. Aveson and Fitzgerald [8] declare that SDM numbers have been overestimated because most SDMs are similar and their differences are due to commercial goals. Nevertheless, Aveson and Fitzgral [8] confirm methodologies proliferation and name it as "methodology jungle".

II. SOFTWARE DEVELOPMENT METHODOLOGY

A software development methodology is a means to succeed in developing software-based systems usually including phase determination, procedures, techniques, documentation and so on [9]. Management uses SDM

intermediate work-products such as test-plans and requirement engineering techniques to lead the development process [3]. SDMs are constantly evolving to adapt to technology changes and to meet customers' changeful needs in today's volatile environment. Recently a new generation of SDM has been evolved, which claims to be more compatible with today's ever-changing business environment [10]. Over the past 10 years, a huge number of organizations have moved towards implementing agility, which has been widely accepted by software organizations. However, most of the organizations either failed to accept agile methodologies or did not manage to be as effective as they were supposed to [11].

III. The Evaluation of Software Development Methodology

The diversity of different software processes and methodologies makes SDM selection complicated for a special project. Furthermore, this causes difficulties in a combination of SDM elements for the construction of appropriate SDM [12]. Such problems make SDM evaluation an important issue. Organizational environment, user preferences and their acquaintance with methodology are dependent variables that highly affect users' decision in SDM selection. If somebody asks a person "what is the best language to speak with?", the answer may not be one single specific language. Actually, people speak a language that is common and they know well. The same condition occurs in reply to what is the best methodology to be used. An alternative could not be appropriate for all situations. Proof of this claim is the increasing rate of SDM customization and adaptation by users according to their environment, project, and organization [13].

IV. Methods

In order to identify the technical criteria for SDM evaluation, we used systematic literature review approach. This approach tries to find all related sources on a topic for a comprehensive review and understanding of concept. For this purpose, we followed the four steps suggested by Webster and Watson [67]. These four steps include:

1) review articles,
2) related articles,
3) go backward and
4) go forward steps.

The investigation conducted among major databases to find related sources. Based on this investigation, about 65 related articles found and reviewed. Grounded on a thorough review of these sources, technical criteria for SDM evaluation extracted that will be presented and discussed in the following part of this study.

V. Technical Criteria for SDM Evaluation

Technical criteria are concerned with SDM suitability for technical characteristics of project and organization, these criteria help to determine SDM technical efficiency and to support perceptual measuring [4]. Technical criteria identified in this study include 15 items and 56 indicators. These items and their corresponding indicators are shown in Table 1.

Table 1: Technical evaluation criteria

Item	Indicator	Source
Implication of SDM on project	Helping to reduce the time needed for completion of the project	[14, 15]
	Helping to improve the project's documentation and traceability	[16]
	Helping to a better estimate of project risks	[4, 17, 18]
	Helping to reduce the number and impact of different problems at the project	[4, 19]
	Helping to improve control over the project	[4]
	Helping to reduce the costs of project and control over the project	[18, 20]
Implication of SDM on system	Helping to develop a complete system	[4]
	Helping to develop a more coherent system	[4, 21]
	Helping to develop more reusable system	[4, 15, 18, 22]
	Helping to develop a more reliable system	[15, 17, 20, 23, 24]
	Helping to develop a more maintainable system	[4, 13, 15, 23-25]
	Helping to develop a more portable system	[4]
	Helping to develop a more efficient system	[7, 14, 20, 26]
Implication of SDM on SDM users	Facilitating the cooperation and unambiguous communication between SDM users	[4, 26]
	Diminishing the number of conflicts concerning SDM users responsibilities and duties	[15]
	Improve the training and facilitate the SDM user's understanding of their duties and responsibilities	[4, 15]
Implication of SDM on organization	Facilitating standardization in the organization	[20]
	Helping the organization in achieving its goals	[15, 27]
	Helping organization to improve its reputation of excellent work	[4, 16]
Implication of SDM on the development process	Helping to decrease the different costs during the software development	[21, 28-30]
	Helping to reduce the cost of support process during software development	[21, 30]
	Covering the main phases of the software development process	[31-33]
	Helping to improve the project management process	[16, 21, 30, 33, 34]

Item	Indicator	Source
	Helping to improve the configuration management process	[11, 21, 30, 31, 33]
	Helping to improve the risk management process	[14, 17, 27, 34-37]
Implication of SDM	Improving customers' trust in the organization	[28, 29, 38]
	Helping to improve customers overview of the project progress	[14, 20]
	Being in line with customers actual needs	[21, 39-41]
	Helping to improve a customer's general satisfaction with the organization	[3, 14, 34, 39, 42-44]
Suitability for project	Project size and complexity (number of developers, number of organizations involved, etc.)	[14, 45-51]
	System type and complexity (size and complexity of subsystems, number, and complexity on interactions between subsystems, integration of legacy applications, etc.)	[19, 29, 39, 45, 51-54]
	Project type (new development, upgrade and repair)	[21, 31, 46, 50, 55]
	Criticality of the system	[4, 11, 14, 20, 21, 29, 38, 56-59]
	Project priorities (productivity, traceability, etc.)	[4, 20]
Suitability for team	Suitability for the development team's knowledge and experience level	[4]
	Having no problem in SDM usage and understanding	[4]
Compliance with modern development approaches	Align with modern approaches and development methods	[4]
	Compatibility with the general standard	[4]
Compatibility with information technology and program environment	Compatibility with the development tools used by the development team and facilitating their use	[52]
	Compatibility with programming languages and facilitating their use	[4]
	Compatibility with the information technology used by development team and facilitating their use	[49]
Compatibility with general standards	Compatibility with general standards in its field(standardized notion, languages, techniques and etc.)	[7]
	Not prescribing a very different way of doing work from others generally performed	[60]
Adaptability to technical needs of the project	The possibility of tailoring SDM according to the size and complexity of the project	[4, 7, 26, 35, 49, 60]
	Possibility of customizing SDM according to the project type (new development, upgrade, repair)	[7, 14, 21, 26, 30, 60]
	Possibility of customizing according to project priorities	[4, 20, 26, 27, 35, 49]
Adaptability to technical needs of the SDM user	Possibility of tailoring SDM according to SDM users knowledge and experience	[14, 61]
Agility	Helping to do more work at minimum time and effort	[20, 48, 51, 52]
	Increasing flexibility of team while encountering changes through the development process	[11, 14, 21, 27, 30, 37, 50, 61-65]
	Helping to improve the modularity of development process	[21, 26, 30, 60,66]
	Helping to improve knowledge and experience of SDM user and learning about what the cause of earlier failure has been.	[20, 54, 68]
	Helping the acceleration of frequent delivery of software	[14, 17, 20, 21, 26, 27, 31, 39, 51, 60, 62-64]
	Improving the team's capability of confronting changing requirements during the development	[21, 26, 35, 39, 61, 69, 70]
	Increasing the ability to receive constant feedback during software development	[20, 27, 30, 33, 46, 61, 71]
	Increasing the transparency of development procedure (being working practice easy to learn and modify and documented accordingly)	[4, 24, 69, 72-76]

VI. CONCLUSION

The aim of this research was to identify technical criteria for software development methodologies. To this end, this study identified 15 technical criteria for SDM evaluation including 56 indicators to describe each of the criteria. These results led to effective software development methodology evaluation and indicated the ultimate success of selection project. Previous studies reported that SDMs are not used all

at once, but rather gradually. In other words, organizations usually use and accept some of SDM fragments such as practices, tools, technique and so on. Thus, these criteria can be used for evaluation of SDM elements, fragments or agile practices. We suggest that the provided criteria and indicators be refined and validated in future studies through different research approaches such as quantitative approach via survey and qualitative approach via expert panel discussion to provide more sound technical SDM evaluation criteria.

REFERENCES

[1] N. D. Birrell, *A practical handbook for software development*, Cambridge university press, 1985.

[2] V. R. Basili, "Quantitative Evaluation of Software Methodology", Tech. rep. TR-1519. University of Maryland, 1985.

[3] F. K. Chan, and J. Y. Thong, "Acceptance of agile methodologies: A critical review and conceptual framework". D*ecision Support Systems*, vol. 46, pp. 803-814, 2009.

[4] D. Vavpotic, and M. Bajec, "An approach for concurrent evaluation of technical and social aspects of software development methodologies". *Information and software technology*, vol. 51. pp. 528-545, 2009.

[5] D. Avison, and G. Fitzgerald, *Information systems development: methodologies, techniques and tools*, McGraw Hill, 2003.

[6] J. A. Bubenko Jr, "Information system methodologies—a research view". *Proceedings of the IFIP WG 8.1 working conference on Information systems design methodologies: improving the practice*, North-Holland Publishing Co., 1986.

[7] K. Siau, and X. Tan, "Special theme of research in information systems analysis and design-IV evaluation criteria for information systems development methodologies". *Communications of the Association for Information Systems*, vol. 16, pp. 44, 2005.

[8] D. E. Avison, and G. Fitzgerald, "Where now for development methodologies?" *Communications of the ACM*, vol. 46, pp. 78-82, 2003.

[9] G. M. Karam, and R. S. Casselman, "A cataloging framework for software development methods", *Computer*, vol.26, pp. 34-44, 1993.

[10] A. Cockburn, and J. Highsmith, "Agile software development: The people factor", *Computer*, vol. 11, pp. 131-133, 2011.

[11] S. Hesari, H. Mashayekhi, and R. Ramsin, "Towards a general framework for evaluating software development methodologies". *Proceedings of the Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual*, IEEE, 2010.

[12] A. S. Griffin, and A. A. Brandyberry, "System development methodology usage in industry: a review and analysis", *Journal of Information Systems Applied Research*, vol. 3, pp. 1-18, 2010.

[13] M. Bajec, and D. Vavpotič, "A framework and tool-support for reengineering software development methods". *Informatica*, vol. 19, pp. 321-344, 2008.

[14] D. Batra, W. Xia, D.VanderMeer, and K. Dutta, "Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry", *Communications of the Association for Information Systems*, vol. 27, pp. 21, 2010.

[15] J. Iivari and M. Huisman, "The relationship between organizational culture and the deployment of systems development methodologies", *MIS Quarterly*, pp. 35-58, 2007.

[16] Z.S.H.S. Abad, M. Hasani and R. Ramsin, "Towards tool support for situational engineering of agile methodologies". *Proceedings of the Software Engineering Conference (APSEC), 2010 17th Asia Pacific*, IEEE, 2010.

[17] L. Vijayasarathy, and D. Turk, "Drivers of agile software development use: Dialectic interplay between benefits and hindrances", *Information and Software Technology*, vol. 54, pp. 137-148, 2012.

[18] C. Gonzalez-Perez, P. Giorgini, and B. Henderson-Sellers, "Method construction by goal analysis", *Proceedings of the 16th International Conference Information Systems Development*, Springer. p. 79-91, 2009.

[19] D. Vavpotič, M. Bajec, and M. Krisper, "Software development methodology evaluation model", *Proceedings of the 12th International Conference on Information Systems Development,* Springer, 2004.

[20] D. Turk, R. France, and B. Rumpe, "Assumptions underlying agile software development processes", *Journal of Database Management (JDM)*, no., pp. 62-87, 2005.

[21] M. Taromirad, and R. Ramsin, "Cefam: Comprehensive evaluation framework for agile methodologies. *Proceedings of the Software Engineering Workshop", 2008. SEW'08. 32nd Annual IEEE*, IEEE, 2008.

[22] M. Khalifa, and J. M. Verner, "Drivers for software development method usage", *Engineering Management, IEEE Transactions on*, vol. 47, pp. 360-369, 2000.

[23] M., Huisman, and J. Iivari, "The individual deployment of systems development methodologies", *Proceedings of the Advanced Information Systems Engineering*, Springer, 2002.

[24] C. K., Riemenschneider, B. C. Hardgrave, and F. D. Davis, "Explaining software developer acceptance of methodologies: a comparison of five theoretical models", S*oftware Engineering, IEEE Transactions on*,. vol. 28, pp. 1135-1145, 2002.

[25] M. Huisman, and J. Iivari, "Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers", *Information & Management*, vol. 43. pp. 29-49, 2006.

[26] T. Madi, Z. Dahalin, and F. Baharom, "Content analysis on agile values: A perception from software practitioners", *Proceedings of the 5th Malaysian Conference Software Engineering (MySEC), 2011*, IEEE, 2011.

[27] A. Sutharshan, "Human factors and cultural influences in implementing agile philosophy and agility in global software development", PhD. thesis, Edith Cowan University, 2013.

[28] D. Vavpotič, and T. Hovelja, "Improving the evaluation of software development methodology adoption and its impact on enterprise performance", C*omputer Science and Information Systems*, vol. 9, pp. 165-187, 2012.

[29] D. Vavpotič, and O. Vasilecas, "Selecting a methodology for business information systems development: Decision model and tool support", *Computer Science and Information Systems*, vol. 9, pp. 135-164, 2012.

[30] M. Taromirad, and R. Ramsin, "An appraisal of existing evaluation frameworks for Agile methodologies", *Proceedings of the 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, 2008.* IEEE, 2008.

[31] M. Tanner, and U. von Willingh, "Factors leading to the succ ess and failure of agile projects implemented in traditionally waterfall environments", Human Capital without Borders: Knowledge and Learning for the Quality of Life. *Portoroz, Slovenia: Make Learn*, 2014.

[32] A. Qumer, and B. Henderson-Sellers, "Measuring agility and adoptability of agile methods: a 4-dimensional analytical tool", *IADIS Press*, 2006.

[33] A. Qumer, and B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice", *Journal of Systems and Software*, vol. 81, pp. 1899-1919, 2008.

[34] D. Vavpotic, and O. Vasilecas, "An Approach for Assessment of Software Development Methodologies Suitability", E*lektronika ir Elektrotechnika*, vol. 114, pp. 107-110, 2011.

[35] M. Umbreen, J.A. Shah, and S.M. Shaheed, "A Comparative Approach for SCRUM and FDD in Agile". *International Journal of Computer Science and Innovation*, 2015.

[36] S.C. Misra, "Adopting agile software development practices: success factors, changes required, and challenges", *Carleton University Ottawa*, 2007.

[37] A. Qumer, B. Henderson-sellers, and T. Mcbride, "Agile adoption and improvement model", In European, Mediterranean and Middle Eastern Conference on Information Systems, *Polytechnic University of Valencia*, 2007.

[38] C. V. Geambasu, I., Jianu, I., Jianu, and A. Gavrilă, "Influence factors for the choice of a software development methodology", *Accounting and Management Information Systems*, vol. 10, pp. 479, 2011.

[39] A. L. Asnawi, A. M. Gravell, and G. B. Wills, "Empirical investigation on agile methods usage: issues identified from early adopters in Malaysia", *Proceedings of the International Conference on Agile Software Development*, *Springer*, 2011.

[40] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices", *Journal of Systems and Software*, vol. 82, pp. 1869-1890, 2009.

[41] F., Karlsson, and P. Ågerfalk, "Exploring agile values in method configuration", *European Journal of Information Systems*, vol. 18, pp. 300-316, 2009.

[42] J. D., Arthur, R. E. Nance, and S. M. Henry, "A procedural approach to evaluating software development methodologies", *The foundation*, 1986.

[43] M. Awad, "A comparison between agile and traditional software development methodologies", School of Computer Science and software Engineering *University of Western Australia*, 2005.

[44] C. Tolfo, “Agile methods and organizational culture: Reflections about cultural levels”, *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 23, pp. 423-441, 2011.
[45] G., Mikulenas, and K. Kapocius, “A Framework for Decomposition and Analysis of Agile Methodologies during their Adaptation”, *Information Systems Development,* Springer, 2011.
[46] G. Mikulenas, and K. Kapocius, “An approach for prioritizing agile practices for adaptation”, *Information Systems Development*, Springer, 2011.
[47] M. Cohn, *Agile estimating and planning*, Pearson Education, 2005.
[48] B., Boehm, and R. Turner, “Using risk to balance agile and plan-driven methods”, *Computer*, pp. 57-66, 2003.
[49] T. Päivärinta, “Method deployment in small-and medium-sized firms: Addressing the organizational context”, *Information Systems Development,* Springer, 2009.
[50] C. de O. Melo, “The evolution of agile software development in Brazil”, *Journal of the Brazilian Computer Society*, vol. 19, pp. 523-552, 2013.
[51] A. S., Campanelli, and F.S. Parreiras, “A Conceptual Model for Agile Practices Adoption”, *Zenodo. Org*, vol.10, 2012.
[52] A. S. Campanelli, “A Tailoring Criteria Model for Agile Practices Adoption”, *Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento*, vol. 4, 2016.
[53] B., Boehm, “Get ready for agile methods with care”, *Computer*, vol. 35, pp. 64-69, 2002.
[54] B. Boehm, and R. Turner, “Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods”, *Proceedings of the 26th international Conference on Software Engineering*, IEEE Computer Society*,* 2004.
[55] A., Cockburn, *“Crystal clear: a human-powered methodology for small teams”*, Pearson Education, 2004.
[56] G. Mikulėnas, and R. Butleris, “An approach for constructing evaluation model of suitability assessment of agile methods using analytic hierarchy process”, *Elektronika ir Elektrotechnika*, vol.106, pp. 99-104, 2005.
[57] L., Vijayasarathy, and C. Butler, “Choice of Software Development Methodologies - Do Project, Team and Organizational Characteristics Matter?” *IEEE Software*, 2015.
[58] D. Vavpotič, M. Bajec, and M. Krisper, “Measuring and improving technical efficiency and social adoption of software development methodologies”, *Proceedings of the Thirteenth International Conference on Information Systems Development,* 2004.
[59] E., Mnkandla, and B. Dwolatzky, “Agile methodologies selection toolbox”, *Proceedings of the International Conference on*,*Software Engineering Advances, 2007. ICSEA 2007,* IEEE, 2007.
[60] S. Soundararajan, J. D. Arthur, and O. Balci, “A methodology for assessing agile software development methods”, *Proceedings of the Agile Conference (AGILE),* IEEE, 2012.
[61] J. F. T., Abrantes, G. H., Travassos, “Towards pertinent characteristics of agility and agile practices for software processes”, *CLEI Electronic Journal*, vol. 16, pp. 6-6, 2013.
[62] M. Stoica, M. Mircea, and B. Ghilic-Micu, “Software Development: Agile vs. Traditional”, *Informatica Economica*, vol. 17, pp. 64, 2013.
[63] J. B. Barlow, “Overview and guidance on agile development in large organizations”, *Communications of the Association for Information Systems*, vol. 29, pp. 25-44, 2011.
[64] A. Sidky, “A structured approach to adopting agile practices: The agile adoption framework”, *Virginia Polytechnic Institute and State University*, 2007.
[65] G. G. Miller, “The characteristics of agile software processes”. *Proceedings of the International Conference on Technology of Object-Oriented Languages,* IEEE, 2011.
[66] M. Aoyama, “Agile software process and its experience”, *Proceedings of the 20th international conference on Software engineering*, IEEE Computer Society, 1998.
[67] J. Webster, and R. T. Watson, “Analyzing the past to prepare for the future: Writing a literature review”, *MIS quarterly*, pp. xiii-xxiii, 2002.
[68] M. Lindvall, “Empirical findings in agile methods”, *Proceedings of the Conference on Extreme Programming and Agile Methods,* Springer, 2002.
[69] H. Holmström, “Agile practices reduce distance in global software development”, *Information Systems Management*, vol. 23, pp. 7-18, 2006.
[70] H. Y. Shahir, S. Daneshpajouh, and R. Ramsin, “Improvement strategies for agile processes: a SWOT analysis approach”, *Proceedings of the International Conference on Software Engineering Research, Management and Applications, 2008. SERA'08. Sixth*, IEEE, 2008.
[71] K. Mohan, and F. Ahlemann, “What methodology attributes are critical for potential users? understanding the effect of human needs”, *Proceedings of the Advanced Information Systems Engineering*, Springer, 2011.
[72] C. K. Riemenschneider, and B. C. Hardgrave, “Explaining software development tool use with the technology acceptance model”, *Journal of Computer Information Systems*, vol. 41, pp. 1-8, 2001.
[73] F. D. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology”, *MIS quarterly*, pp. 319-340, 1989.
[74] A. S. Griffin, “Examining the decision process and outcomes of system development methodology adoption”, *Kent State University*, 2008.
[75] B. C. Hardgrave, and R. A. Johnson, “Toward an information systems development acceptance model: the case of object-oriented systems development”, *Engineering Management, IEEE Transactions on*, vol. 50, pp. 322-336, 2003.
[76] B. C. Hardgrave, F. D. Davis, and C. K. Riemenschneider, “Investigating determinants of software developers' intentions to follow methodologies”, *Journal of Management Information Systems*, vol. 20, pp. 123-151, 2003.