

# Role of Situational Agile Distributed Model to Support Modern Software Development Teams

AMBER SARWAR HASHMI\*, YASER HAFEEZ\*, MUHAMMAD JAMAL\*\*, SADIA ALI\*, AND  
NAILA IQBAL\*

RECEIVED ON 01.06.2018 ACCEPTED ON 17.08.2018

## ABSTRACT

ADSD (Agile Distributed Software Development) is a growing trend for software development organizations to develop quality software with limited cost and time. However, it gives rise to additional situational challenges. Situational variations result in unstable agile architecture which gets highly affected. Although the present literature focuses on agile architecture but effect of situational variation on the agile architecture still needs consideration. One possible solution is development of conceptual model and incorporation of multiple situational factors. This research study aims to identify the most relevant situations and propose situational ADSD approach for the development of situation-based software architecture framework for agile distributed projects. The study focuses on agile methods, rendering identified situational variations. The approach is developed through analyzing data from literature and their associated work documents. Comprehensive survey helps to bridge the gaps from conceptual to architecture model. The experimental results are acquired through both practical and statistical analysis. The results support Conway's law, e.g. correlation that maps architecture with the communication and coordination needs. Correlation results show that architecture has major contribution in ADSD and validate the relationship amongst conceptual model and architecture model. The results also suggest that stable architecture in ADSD can positively affect the product.

**Key Words:** Agile Distributed Model, Situational Agile Distributed Model, Agile Architecture, Agile Methods, Situational Variations.

## 1. INTRODUCTION

Many modern development strategies such as agile paradigm has received much attention in last few years. ADSD [1] is gaining popularity due to the combined benefits that it offers, like reduced development costs, better and quicker quality production [2], shorter time to market [3-4] and

more cost-efficient workforces located in low-cost countries [5]. Furthermore, benefit of having access to distributed pool of expert workforce, thus creates virtual co-located corporations. Along with the above discussed success factors, many critical issues also exist like distance effects on communication within a team [6].

Authors E-Mail: (ambersarwar@hotmail.com, yasir@uaar.edu.pk, jamal@uaar.edu.pk, sadiaalief@gmail.com, nailaiqbal760@gmail.com)

\* University Institute of Information Technology, Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, Pakistan.

\*\* Department of Mathematics & Statistics, Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, Pakistan.

The architecture plays an important role within an organization and is responsible for defining and maintaining framework structure to certify requirements satisfaction. It helps to analyze stakeholder's requirements, formulate framework design and ensures that everyone viewpoints understand in architecture, support developers, and do implementation verification in order to comply with system requirements. Therefore, architecture describes an organizational framework components and their interactions [7].

Similarly, researchers have assessed agile architecture and its role in conjunction with management of late changes [7-11]. An agile architecture helps in team coordination for system organization and identifies new ideas from each member view points for proper requirement management. Agile architect is responsible for change consideration freely, and control complexity. However, agile methodology still lacks answers to architectural questions [8]. For instance, agile only suggests so-called "architectural runway" infrastructure for set of renewed interest [9].

ADSD projects are not straight forward as they make the process complex due to situational variations [12-13]. The software development practitioners seek to utilize development practices according to different situations that fit within the development environment [14-15]. Still no single methodology is applicable enough thus the significance of SME (Situational Method Engineering) has been increased within agile practitioners [16-17]. The agile practitioners seek the creation and adjustment of ADSD methodologies to specific situations which are highly dependent on the relationships of concepts [1]. Due to the situational variations in ADSD, architectural design is highly affected [15]. However, architecture is a potential barrier in agile as it proposes architectural runway. Thus, it is desired to have structured key architectural design decisions [18]. Finding the optimal

balance between conceptual information to the architecture design is significant for architectural decision [19-20].

In this research work, we propose a situational ADSD approach. Furthermore, situations identified are incorporated within the agile architecture. This is achieved through bridging the gap from conceptual model to architecture model. However, far too little attention has been paid to the formulation of conceptual model related to situational variations. The proposed conceptual model plays an important role in architecture model based on situational factors for ADSD.

The proposed work primarily targets practitioners (managers and team members) involved in agile distributed development. It will also serve as a useful source of practical advice for those who are practicing agile methods as an agile transformation and agile adoption. The work will hopefully motivate companies to improve their sourcing strategies by adopting best practices. The proposed work is based on empirical research and identifies gaps in the existing state of the art practice. Therefore, the proposed work can also be relevant and interesting for academic audience and researchers working in the field of ADSD.

Some researchers do list some situational method engineering approaches while developing agile distributed projects. Furthermore, there exist certain approaches where authors propose conceptual to architecture model for ADSD. However, they do not cover situation-based agile architecture for distributed agile development. While there are several studies providing recommendations and lessons learned to facilitate situational ADSD [21] and defined some quality criteria i.e. adaptability, complexity, agility etc. [22]. The evidences and research contributions in the field of ADSD are briefly discussed in the following paragraphs.

O'Connor [15] proposed a reference model based on situational factors in ADSD. The researchers combined related research of the situational factors into a primary reference framework. They applied Grounded Theory for creating reference framework of situational factors which classified 44 factors manipulating the software development process. Although, the initial framework focused distributed projects but was not applicable to agile environment. Further, data sources included in initial framework are limited. Additional domains like architectural patterns could be incorporated into the framework.

An extension to the above mentioned approach provides a set of situational factors which are important while adopting agile methodologies; experience of team members, skills, performance in terms of adoption, rigidity and quality of the new process [23-26]. In addition, supported relevancy with SME in agile methods of Akbar et. al. [27] and Dwivedi [28] scrutinized agile method tailoring approaches and proposed configuration efforts that were required during agile method tailoring. Previous studies have reported quality considerations while adopting agile methodologies but they did not highlight quality consideration for adoption of agile methodologies in distributed environment.

Brasil et. al. [25] and Bellomo et. al. [26] proposed a situational method-based enterprise architecture in incorporating architecture patterns. Whereas, Noordeloo et. al. [29] and Khan et. al. [30] highlighted the effect of situational factors in agile method based global software development process. These studies [25-26,29-30] highlighted situational factors like; distance, team size, skills, knowledge, maturity, complexity and architecture. Furthermore, expanding their work also conducted a structured literature review to elicit requirements for enterprise architecture planning from a research perspective. Their research combined the

results of both the practitioner interviews and the literature review thus, emphasizing the gap between the two worlds. They identified that current research does not adequately address the situational problems of architecture planning practice and solution shall to be addressed.

Similarly, O'Connor [15] revealed the link of software development process to its context. However, there exists lack of explicit guidance for the harmonization of a development process with situational context. Bakhat et. al. [31] supported agile methodologies in dynamic context with the changing situations. According to Tripp and Armstrong [2] even though an organization's objects may be attentive that various teams' and team members' goals may not be entirely affiliated with organization's objectives. Organizations should uninterruptedly reconsider agile practices used to ensure that agile methodologies are vital for the cost saving and increase the production in tight schedule. In addition, ADSD organization have versatile background teams faced situational problem along with architectural stability issue [17-18].

Although situational factors have been identified for agile methodology as well as distributed software development, still there is lack of situational factors in ADSD. Therefore, we proposed situational repository in our propose model. The situational repository will map those situational factors in agile architecture.

The rest of the paper is organized as follows. Section 2 presents the materials and method. Section 3 discusses the experimental results. Section 4 concludes the paper.

## **2. MATERIALS AND METHOD**

The study is focused to empirical software engineering domain. In doing so, we conducted industrial evaluation

in order to get detail insight. Both survey and interviews were conducted in order to understand the effect of situational variations with the ADSD context in ADSD domain. A detailed questionnaire was formulated and an online survey was used to identify the situational factors that impact ADSD. The objective was to determine the impact of situational factors on ADSD development with reference to practitioners. This research highlights the relevancy of situation to the concepts with specific context. This research aims to develop conceptual model on the basis of identify situational factors which further identified the impact of situational factors on agile architecture.

Survey was conducted with practitioners to recognize in depth familiarities regarding the effect of situational variations on ADSD. Practitioners from LinkedIn and Google+ were selected on voluntary basis. Eleven (11) interviews were conducted based on different dimensions of model on different situational factors from ADSD practitioners having experience between 9-24 years and affiliated with ADSD organizations. The practitioners were selected from Pakistan, US, UK, and Dubai. Practitioners were contacted via Skype call. The practitioners experienced diverse team size working in different domains mainly; developing software solutions for IT, Telecom, Finance etc. The Project Managers of some well-known large organizations also participated in the survey.

**Survey Instrument:** The questionnaire contained 14 partially structured questions measured on ordinal and nominal scales. The Likert-scale had 5 points ranking of significant to least significant. The “do not know” option was also included to diminish clamor in the received data. The questionnaire included 4 segments, including; introductory questions, ranking of situational factors, effect of situational factors on ADSD, relationship between architecture model based on situational factors

and conceptual model. The survey clearly stated the objectives and data sharing statement to guarantee respondents confidentiality. Pilot testing is a 2 stage process, firstly validation by experts, and secondly, assessment from mature ISO-9001 certified organization. Applied probing method for cognitive testing of the survey was used. Mainly the questions were related to retrieval, comprehension, and judgement.

**Survey Execution:** The online platform of Survey Crest was used for conducting online survey with the active timeline of 2 months. Total 106 respondents participated, however, only 54 responses were included. A contingency question was used to filter responses in the classes of industry or practitioners. Responders from social media took long time to respond. Also forwarded e mails to the project managers and team members of ADSD organizations like; Microsoft, IBM, Oracle and Siemens.

**Proposed Approach:** In this study, we present an effective model for applying situational method engineering in ADSD. We also review the existing models and determine their strengths and limitations. Based on these findings, we present situational method engineering in ADSD i.e. team size, commitment, process reuse etc. Whereas, contextual factors are application, projects and program. Table 1 shows the taxonomy of situational factors identified from literature and highlights the area of research.

From the literature review, it is clear that apparently, with synchronizing trends of ADSD approach, various situational issues emerge due to the inherited contextual factors which comprise of instable architecture, versatile situations and unhandled conceptual factors. Therefore, situational ADSD approach is needed to handle the above mentioned challenges and resulting in bridging the gap from conceptual to architecture model.

The proposed approach is three tier architecture. First is application tier, which has an Always-On availability group, and is accessible to all distributed teams as shown in Fig. 1. Requirements are gathered in form of user stories and situational factors are defined. The product owner is

responsible for creating product backlog and prioritizing user requirements. As we manage requirements early, it assists in further development phases. Method engineer classifies situational factors from the requirements in product backlog.

TABLE 1. CLASSIFIED SITUATIONAL FACTORS

Contextual Parameter	Situational Classification	Situational Factors	Data Sources												
			Azzouz et. al. [22]	Brasil et. al. [25]	Dwivedi [28]	Noordeloo et. al. [29]	Khan et. al. [30]	Bakhat et. al. [31]	Agh and Ramsir [32]	Akbar et. al. [27]	Imtiaz and Ikram [33]	Gul and et. al. [12]	Tipp and Armstrong [2]	Bass [18]	Hekkala [9]
Agile Distributed Software Development	Personnel	Team size				*		*		*		*			*
		Distance, Language							*	*		*		*	
		Experience, Skill	*	*	*				*			*			*
		Commitment	*	*	*			*		*					*
		Interaction Styles	*	*	*				*						*
	Requirements	Feasibility, Changeability			*		*		*	*	*				*
		Standard, Rigidity	*			*	*		*	*	*				*
	Application	Degree of Risk				*	*	*							
		Performance	*	*	*	*	*	*						*	
		Complexity, Type, Size									*			*	
		Process Reuse													
		Development phase							*						
		Agile Methodology					*				*		*	*	
	Technology	Quality	*	*	*										
		Knowledge								*		*			
	Organization	Emergent								*		*			
		Maturity					*			*		*			*
		Size					*			*		*			*
		Commitment					*			*		*			*
	Management	Structure				*	*	*	*	*	*	*			*
		Expertise								*				*	*
		Continuity													*
		Accomplishment								*				*	*
		Time to market								*				*	*
		Customer Satisfaction	*	*		*	*			*				*	*
		Business Drivers						*					*	*	

Second is the middle tier where requirements are further stored in situational method repository in form of artifacts, PDF (Portable Document Format), XML (Extensible Markup Language) or word documents. In the data tier, these requirements are passed to the cross functional teams. Spikes are created by extracting situational factors which linked spikes through data retrieval technique for the architectural design.

The proposed model is based on the principles and practices of agile methodologies and situational method engineering thus keeping ADSD values in focus as shown in Fig. 1. From the literature review, importance of situational method engineering is evident in ADSD. There exists significant gap between conceptual model and architecture model.

### 3. RESULTS AND DISCUSSION

The results focus on the relative impact of situational factors and are discussed through bridging the relationship between situation based conceptual factors

and the architecture model. The survey responses regarding importance of agile architecture and conceptual model are also discussed.

**Impact of Situational Factors:** The relative impact of situational factors is shown in Fig. 2. Likert scale options have been depicted through multiple colors on the stacked. For better readability lower frequency options like; ‘unimportant’ and ‘do not know’ have been removed. To limit the relative ranking value of situational factors, different weights have been assigned; 4 to most significant and 0 to least significant. Fig. 2 shows the relative ranking of each situational factor assigned by the practitioners. Correlation value has also been included which depicts the importance of each situational factor. In Fig. 2, response data depicts the importance of expertise and site characteristics as ranked by the GSD practitioners. Note that the remaining situational factors are ranked low. Whereas, time differences and cultural factors have been ranked as less significant by the practitioners. The comprehensive frequency has been shown in Table 2.

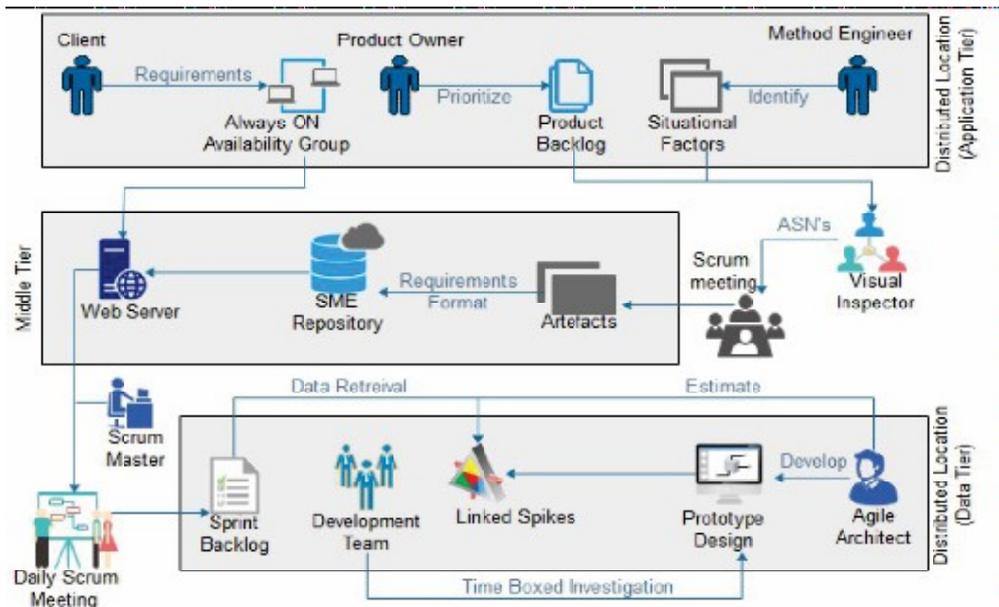


FIG. 1. SITUATIONAL AGILE DISTRIBUTED SOFTWARE DEVELOPMENT MODEL

**Relationship between Situations Based Conceptual Factors and Architectural Dependency:** The relationship between situation and concept within specific context is due to architectural or conceptual dependency. The success of ADSD depends on the optimal balance between situational factors and architectural dependency. Similarly, architectural dependency is defined as the dependency of architecture on different situational factors. Therefore, we identify the impact of

architectural dependency from the viewpoint of Situational variations. Architectural dependency comprises of four types of dependencies; (1) conceptual dependency between tasks, (2) conceptual dependency amongst people, (3) conceptual dependency amid resources and (4) conceptual dependency amongst sites. Fig. 3 shows the architectural dependencies. The ranking depicted by practitioners with reference to architectural dependency is shown in Fig. 4.

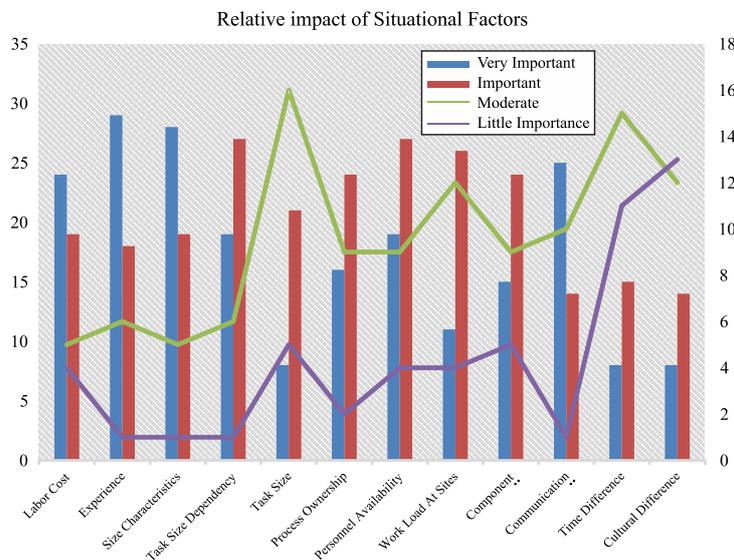


FIG. 2. IMPACT OF SITUATIONAL FACTORS IN AGILE DISTRIBUTED SOFTWARE DEVELOPMENT

TABLE 2. RELATIVE RANKING OF SITUATIONAL FACTORS

Situational Factor	Value Score	Correlation
Expertise	183	5
Site Characteristics	180	5
Task Site Dependency	170	2
Labor Cost	167	4
Personnel Availability	164	2
Communication Overhead	163	6
Process Ownership	156	6
Component Dependency	155	7
Workload at Distributed Sites	150	2
Task Size	130	3
Time Difference	118	3
Cultural difference	111	3

**Relation Between Conceptual and Architecture Model:**

Agile architecture in form of conceptual dependency is highly dependent on situational variations. It is considered important by 76% of the practitioners. For exact impact of situational variations on the architecture, the practitioners' selected conceptual dependencies look into situational factors before maintaining the architecture relationship. The architecture model formed on the basis of situational variations had three viewpoints. (1) Conceptual view type. (2) Allocation view type. (3) Component and connector view type. Fig. 5 highlights the architecture viewpoints according to ADSD practitioners based on situational variations. The figure highlights the significance of conceptual view type thus showing relationships between concepts as being significant for architecture development. Fig. 6 depicts the relation between architecture and conceptual dependencies on basis of identified situational variations.

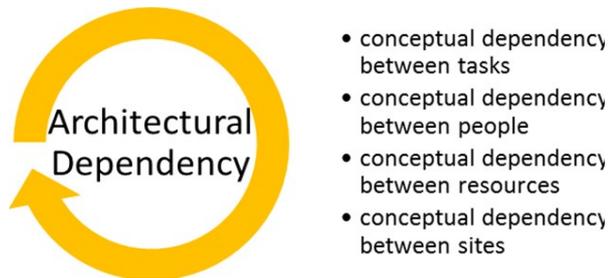


FIG. 3. CONCEPTUAL DEPENDENCY

**Statistical Analysis:** The survey was conducted through Survey Crest, which facilitated exporting results to Excel files. Data analysis was performed by importing results to the Statistical Package [8,32-33]. Mann Whitney U test, Spearman correlation and Vargha and Delaney test were performed. Significance and impact of situational factors were calculated through Mann Whitney U test, where situational factors variables are termed as dependent. The alpha value (significant value) was <0.05, thus depicting 95% chance of difference between groups. Significance of Mann Whitney U test was that dependent variable represented ordinal while independent variables represented nominal scale. The Mann Whitney U test

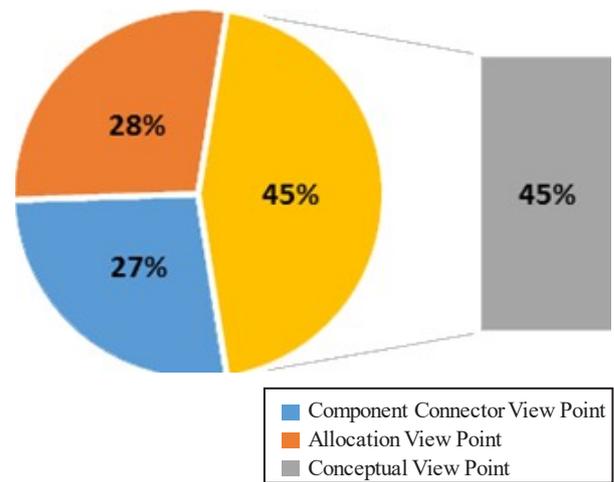


FIG. 5. ARCHITECTURE VIEWPOINTS

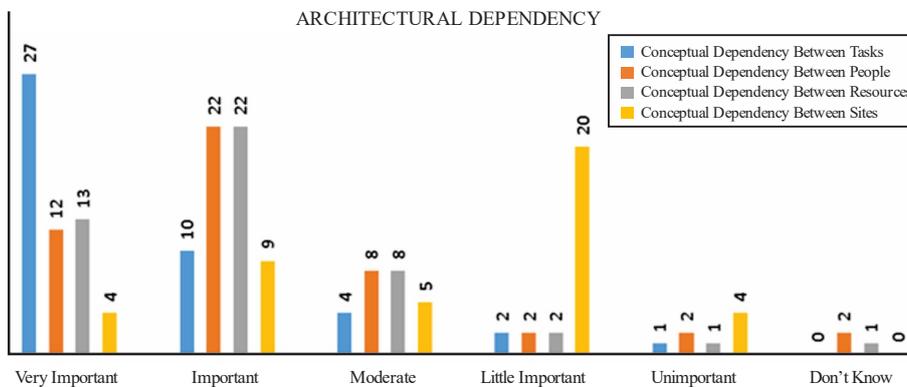


FIG. 4. ARCHITECTURAL DEPENDENCY ON THE BASIS OF SITUATIONAL VARIATIONS

was best suited as data was not normal. If the resulted value was below the alpha, it was rejected. The resulted value should lie between 0 and 1. Where 0.5 magnitude showed no difference, small difference lied in 0.56, medium and large difference were 0.64 and 0.71 respectively. Similarly, the organization's ADSD expertise affected the projects. With the expertise skills comes the maturation. Thus, effect of ADSD expertise was important for noticing situational factors in ADSD. Similarly, impact of the situational variations is depicted in Fig. 7 and defined based on correlation. Correlation is shown in form of bubble diagram.

By looking at the results, situational factors like expertise has high significance, whereas labor cost has less

significance according to ADSD practitioners. Similarly, in Fig. 2 expertise factors such as site characteristics have high significance. Although labor cost serves as the motivating factor for distributed development, it is not on priority as compared to expertise when developing architecture. Labor cost is highlighted as significant factor by ADSD practitioners that distribute tasks on experimental or low value added basis.

The correlations between situational factors as per the Spearman coefficient were also calculated and shown in Table 2. The results show that correlations have been significant due to the handling of architectural dependency. The results have become significant through incorporation of situational repository and linked spikes

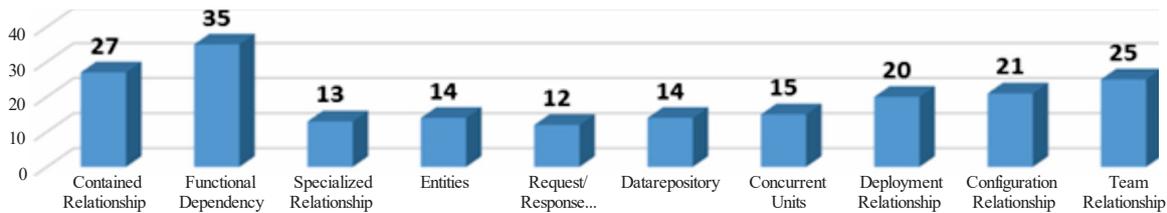


FIG. 6. RELATION BETWEEN ARCHITECTURE AND CONCEPTUAL DEPENDENCIES ON BASIS OF SITUATIONAL VARIATIONS

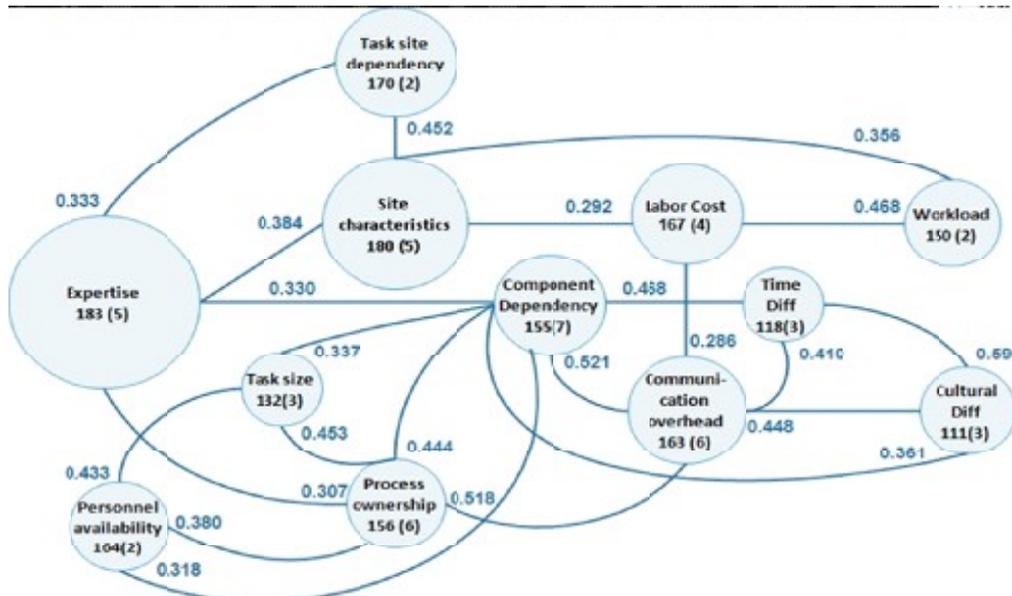


FIG. 7. IMPACT OF SITUATIONAL VARIATION AND CORRELATION AMONG TEAM

in our proposed model. Situational repository supports situational factors while linked spikes map those situational factors with the architecture. However, previous model did not cater the issue of situational factor and architectural dependency. The results also support Conway's law, e.g. correlation that maps architecture with the communication and coordination needs. 0.521 correlation results in moderate significant relationship. Results also reveal that 70% architecture contributes in ADSD. Therefore, relationship amongst conceptual model and architecture model is verified. Types of architectural dependencies considered for situational development highlight more importance than traditional development models. Also, it was revealed that a stable architecture in ADSD can positively affect the product.

#### **4. CONCLUSION**

Situation based architecture development is of high significance to ADSD as its success depends on effective conceptual and architecture dependency. ADSD is a complex task prejudiced by numerous identified situational features. The significance of the identified situational factors has been discussed in literature. However, the literature lacks depth when it comes to architecture development based on situational variations. Distributed teams work on limited criteria as there is lack of empirical research on instable architecture and impact of situational variations on the architecture. This breach encouraged us to gain insight into ADSD industry. The results have highlighted the significance of situational factors for bridging the gap between conceptual and architecture model by handling dependencies. The correlations among the situational factors emphasize that situational factors cannot be

evaluated individually. Each architecture trade off has associated benefits and shortfalls, that are estimated only with relevance to situation factors. Discrepancies in situations arise due to asymmetrical dependencies liketask, organization, ADSD configuration, and ADSD objective. We have presented situational variations and situational factors that need to be considered for well-developed architecture.

#### **5. FUTURE WORK**

Our proposed approach targets only small and medium sized organizations. In the future, we will extend our work to large scale organizations and multiple case studies will be conducted in order to evaluate the identified situational factors.

#### **ACKNOWLEDGEMENT**

The authors are grateful for the support of F3 Technology, Rawalpindi, and GR Solution, Islamabad, Pakistan, for vetting of data and survey.

#### **REFERENCES**

- [1] Alzoubi, Y.I., Gill, A.Q., and Al-Ani, A., "Distributed Agile Development Communication: An Agile Architecture Driven Framework", JSW, Volume 10, No. 6, pp. 681-694, 2015.
- [2] Tripp, J.F., and Armstrong, D.J., "Agile Methodologies: Organizational Adoption Motives, Tailoring, and Performance", Journal of Computer Information Systems, pp. 1-10, Available at: <https://www.tandfonline.com> [DOI: 10.1080/08874417.2016.1220240], 2016.
- [3] Allahyari, S., Salari, M., and Vigo, D., "A Hybrid Metaheuristic Algorithm for the Multi-Depot Covering Tour Vehicle Routing Problem", European Journal of Operational Research, Volume 242, No. 3, pp. 756-768, 2015.

- [4] Alahyari, H., Svensson, R.B., and Gorschek, T., "A Study of Value in Agile Software Development Organizations", *Journal of Systems and Software*, Volume 125, pp. 271-288, 2017.
- [5] Ilyas, M., and Khan, S.U., "Software Integration in Global Software Development: Success Factors for GSD Vendors", 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 1-6, June, 2015.
- [6] Bavani, R., "Critical Success Factors in Distributed Agile for Outsourced Product Development", *Proceedings of International Conference on Software Engineering*, pp. 75-79, December, 2009.
- [7] Alzoubi, Y.I.A., "An Agile Enterprise Architecture Driven Approach to Enhance Communication in Geographically Distributed agile Development", Ph.D. Thesis, Faculty of Engineering & Information Technology, University of Technology, Sydney, 2016.
- [8] AL-Tahat, M.D., and Bataineh, K.M., "Statistical Analyses and Modeling of the Implementation of Agile Manufacturing Tactics in Industrial Firms", *Mathematical Problems in Engineering*, pp.1-23, 2012 Available at: <http://www.hindawi.com/journals/mpe/2012/731092/> [Accessed May 23, 2015].
- [9] Hekkala, R., Stein, M., and Rossi, M., "Challenges in Transitioning to an Agile Way of Working", pp. 5869-5878, 2017.
- [10] Ali, S., Iqbal, N., and Hafeez, Y., "Towards Requirement Change Management for Global Software Development using Case Base Reasoning", *Mehran University Research Journal of Engineering & Technology*, Volume 37, No. 3, Jamshoro, Pakistan, July, 2018.
- [11] Saeed, K., Hafeez, Y., Ali, S., Iqbal, N., and Shahid, U., "Enabling the Usability Heuristics of Agile Base Systems to Improve Quality of Local Software Industry", *Mehran University Journal of Engineering & Technology*, Volume 38, No. 2, pp. 341-350, Jamshoro, Pakistan, April, 2019.
- [12] Gul, Z., Hafeez, Y., Hashmi, A.S., Ali, S., and Iqbal, N., "Introducing Team Coordination Framework Support Globally Distributed Teams for Pakistani Software Industry", *Mehran University Journal of Engineering & Technology*, Volume 37, No. 4, pp. 669-684, Jamshoro, Pakistan, October, 2018.
- [13] Khan, H., Mahrin, M., and Chuprat, S., "Risk Generating Situations of Requirement Engineering in Global Software Development", *Engineering and Information Science*, pp. 221-233. 2013 Available at: <http://sdiwc.net/digital-library/risk-generating-situations-of-requirement-engineering-in-global-software-development>.
- [14] Rolland, C., and Prakash, N., "A Proposal for Context-Specific Method Engineering", *Method Engineering*, pp. 191-208, Springer, Boston, MA, 1996.
- [15] O'Connor, D.J., "An Introduction to the Philosophy of Education", Routledge, 2016.
- [16] Al-Zewairi, M., Suleiman, D., and Almajali, S., "An Experimental Software Defined Security Controller for Software Defined Network", *IEE 4<sup>th</sup> International Conference on Software Defined Systems*, pp. 32-36, May, 2017.
- [17] Campanelli, A.S., "A Model for Agile Method Tailoring", *Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento*, Volume 3, No. 2, 2014.
- [18] Bass, J.M., "Artefacts and Agile Method Tailoring in Large-Scale Offshore Software Development Programmes", *Information and Software Technology*, Volume 75, pp. 1-16, [DOI: 10.1016/j.infsof.2016.03.001], 2016.
- [19] Luna, A.J.H., de O., Kruchten, P., and Moura, H.P. de., "Agile Governance Theory: Conceptual Development", 12th International Conference on Management of Technology and Information Systems, pp. 23, 2015. Available at: <http://arxiv.org/abs/1505.06701>.
- [20] Guo, K., Pan, W., Lu, M., Zhou, X., and Ma, J., "An Effective and Economical Architecture for Semantic-Based Heterogeneous Multimedia Big Data Retrieval", *Journal of Systems and Software*, Volume 102, pp. 207-216, 2015.

- [21] Khan, S.U., Niazi, M., and Ahmad, R., "Barriers in the Selection of Offshore Software Development Outsourcing Vendors: An Exploratory Study Using a Systematic Literature Review", *Information and Software Technology*, Volume 53, No. 7, pp. 693-706, 2011. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0950584910001527> [Accessed April 2, 2015].
- [22] Azzouz, S., Kraiem, N., and Ghezala, H., "Defining Quality Criteria in Situational Method Engineering", *International Journal of Computers & Technology*, Volume 5, No. 2, pp. 109-119, 2013. Available at: <http://74.220.215.92/~ijctonli/index.php/ijct/article/view/684>.
- [23] Agh, H., and Ramsin, R., "A Pattern-Based Model-Driven Approach for Situational Method Engineering", *Information and Software Technology*, Volume 78, pp. 95-120, 2016.
- [24] Farwick, M., Schweda, C.M., Breu, R., and Hanschke, I., "A Situational Method for Semi-Automated Enterprise Architecture Documentation (SoSyM Abstract)", *ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems*, pp. 448-448, September, 2015.
- [25] Brasil, M.A.B., Pereira, G.V., and Fontoura, L.M., "Software Process Tailoring Using Situational Method Engineering Based on Criteria of Quality Improvement", *Proceedings of 38th Latin America Conference on Informatics*, pp. 7, 2012.
- [26] Bellomo, S., Nord, R.L., and Ozkaya, I., "Elaboration on an Integrated Architecture and Requirement Practice: Prototyping with Quality Attribute Focus", *Proceedings of 2<sup>nd</sup> International Workshop on the Twin Peaks of Requirements and Architecture*, pp. 8-13, Twin Peaks, 2013.
- [27] Akbar, R., Hassan, M.F., and Abdullah, A., "A Framework of Software Process Tailoring for Small and Medium Size IT Companies", *International Conference on Computer & Information Science*, Volume 2, pp. 914-918, 2012. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6297156>.
- [28] Dwivedi, R., "Configuration Issues and Efforts for Configuring Agile Approaches-Situational based Method Engineering", *Volume 61, No. 17*, pp. 23-27, 2013.
- [29] Noordeloos, R., Manteli, C., and Vliet, H.V., "From RUP to Scrum in Global Software Development: A Case Study", *IEEE 7<sup>th</sup> International Conference on Global Software Engineering*, Volume 3, pp. 31-40, 2012. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337395>.
- [30] Khan, H.H., Mahrin, M.N.B., and Chuprat, S.B., "Situational Factors Affecting Requirement Engineering Process in Global Software Development", *IEEE Conference on Open Systems*, pp. 118-122, 2013. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6735059>.
- [31] Bakhat, K.A., Sarwar, A.A., Motla, Y.H., and Akhtar, M.C., "A Situational Requirement Engineering Model for an Agile Process", *Bahria University Journal of Information & Communication Technology*, Volume 8, No. 1, pp. 21, 2015.
- [32] Arcuri, A., and Briand, L., "A Practical Guide for Using Statistical Tests to Assess Randomized Algorithms in Software Engineering", *33<sup>rd</sup> International Conference on Software Engineering*, pp. 1-10, 2011.
- [33] Imtiaz, S., and Ikram, N., "Dynamics of Task Allocation in Global Software Development", *Journal of Software: Evolution and Process*, Volume 29, No. 1, pp. 1-17, 2017.