Agility practices for software development: an investigation of agile organization concepts

Full paper

Kevin Kusuma
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
Auckland, New Zealand
Email: kevinkusuma21@yahoo.com

Ramesh Lal
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
Auckland, New Zealand
Email: Ramesh.lal@aut.ac.nz

Shahper Richter
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
Auckland, New Zealand
Email: shahper.richter@aut.ac.nz

Abstract

In the context of agile software development in New Zealand and Australia, this paper examines the organizational agility related practices with agile software development. The paper looks at agile software development practices in established software development teams and proposes further practices based on organizational agility concepts that can support the agile software development manifesto. With a focus on the organizational agility for agile software development, this study adds to the limited body of research into theories for agile software development. The survey method is used in in conjunction with partial least squares (PLS) method to examine the organizational agility practices that best support agile software development. Based on eight organizational agility concept related practices for agile software development are proposed and validated through this process. Our findings suggest that, knowledge management, organizational culture, organizational learning, competencies, responsiveness, speed, team effort, and workforce agility are vital elements for achieving software development agility.

Keywords: Agile software development, agile organisation concepts, software development agility, agility practices.

1 INTRODUCTION

While agile approach has become mainstream there appears to very little research focussing on organisational agility (agile organization) with software development (West and Grant, 2010). Organisational agility ought to enable software development teams to deal with unpredictable situations and rapid changes to develop market-driven software (Highsmith, 2002). However, while agile adoption is driven by its 4 values and 12 principles of agile software development (agile manifesto-https://agilemanifesto.org/principles.html) it does not appear to provide the mind-set for organizational agility (Highsmith, 2002). The agile manifesto appears to be limited to software engineering.

Most studies undertaken to investigate agile approach for software development focus on method adoption. Hardly any adoption experiences report organizational agility as the key motivation for the agile method adoption. There are just a few research that investigated agility. Lee and Xia (2010) investigated team autonomy, team diversity and responsiveness ability. However, there are several other organizational agility concepts (Sherehiy et al., 2007).

The biggest issue facing agile software development is the lack of theoretical foundation to guide development for better outcome. According to Jacobson and Spence (2009) theoretical foundations are needed to extract essential software development concepts independent of the development method(s). Under the agile software development umbrella there are several agile methods with their own values, principles and practices, besides the 4 values and 12 principles of agile software development. Agile adoption can be a challenge since it is a difficult to relate the agile manifesto with agility.

This study investigates organizational agility concepts to identify agility practices for agile software development leading to organizational agility. Knowing these organizational agility practices together with the agile values and principles (manifesto) are critical (Highsmith, 2002), as they shape agile method adoption enabling development teams to have a strategic development process.

One critical reason for organizational agility for software development is due to the requirements being impacted by dynamic business environment and emerging technologies (Schmidt, Lyytinen, and Mark Keil, 2001). A report produced in 2012 by the Project Management Institute shows that 75% of the participants stated "the ability to rapidly respond to the opportunities" and another 64% of respondents indicated "having the capability for shorter development, review or decision life cycle" as vital characteristics of organizational agility. However, there appears to be a lack of understanding on agility practices that drive agile adoption despite almost two decades of agile software development.

1.1 Benefits of Organizational Agility

The organisational agility concept comes from the organizational psychology discipline (Goldman et al., 1995). Researchers such as Christopher (2000), Gunasekaran (1998), Sherehiy et al., (2007) and Yusuf et al., (1999) have identified organizational agility concepts. Agility is defined as the ability of organizations to respond to challenges and deal with rapidly changing global markets to provide high-quality products (Bernardes and Hanna, 2009).

Organizational agility capability enables identification of market opportunities with speed and enhance opportunities (Mason-Jones, Naylor, and Towill, 2000). Moreover, organizational agility enables to adapt to meet customer demand. Finally, organizational agility capability enables to enrich customers through cooperation, (Goldman, Nagel, and Preiss, 1995). Thus, organizational agility is also critical for software development teams to have an on-going capability to learn and adapt their processes and skills.

1.2 Research Objectives

The main objective of this study is to provide practices based on organisational agility concepts supporting agile manifesto to enable organisational agility with agile software development.

The research question for this study is as follows:

"What are the software development agility practices based on organizational agility concepts?"

2 LITERATURE REVIEW

Initially, a significant amount of literature review was undertaken on agile organisation concepts, agile approach for software development (on different agile methods and practices including the scaled-agile methods). Table 1 presents a list of organizational agility concepts not only applicable to agile software development but for any industry or organization with an agile mindset. Hence, the eight individual organizational agility concepts identify critical elements or abilities organizations ought to have for agile

software development. For this paper, due to space limitation a summarised version of the literature including a limited number of references are provided.

Organizationial Agility Concepts	**Definition**
Workforce Agility	The ability to learn and adapt skills and practices in a short time frame to act on change for successful product development outcomes. Breu, Hemingway, Strathern, and Bridger, 2002). Ability for spontaneous collaboration and work in multiple roles (Dyer and Shafer, 2003). The ability to change direction and achieve different objectives (Sharifi and Zhang, 1999). Empowered individuals that can work in cross-functional teams and are able to make effective decisions (Yusuf, Sarhadi, and Gunasekaran, 1999).
Competencies	Organizational ability to develop unique business and development practices including services/products that make it hard for competitors to copy. (Sharifi and Zhang, 1999) Wider abilities to improve productivity (efficiency and effectiveness) (Sharifi and Zhang, 1999).
Speed	The organization's capability to execute business operations swiftly in delivering products/services to market. (Sherehiy et al., 2007). The ability to perform tasks in shortest possible time (Sharifi and Zhang, 1999).
Responsiveness	The organization's capability to detect, anticipate and deal with changes at the marketplace. (Sharifi and Zhang, 1999) Flexibility to respond to changes, quick to adapt if change happens and swiftly upskill (Breu, Hemingway, Strathern, and Bridger, 2002). The ability to identify changes and respond quickly (Sharifi and Zhang, 1999).
Knowledge Management	The process of creating and maintaining knowledge for continuous improvement and effective effort. Utilize the network connection and have strategic partnership to create high quality information (Lin, Chiu, and Chu, 2006).
Organizational Learning	The process of building, supplementing and organizing knowledge around activities to improve efficiency and adaptation. Learning and self-development, problem-solving ability and ability to generate innovative ideas (Plonka, 1997).
Organizational Culture	The organizational assumptions, beliefs and values that are shared organization-wide to be successful at the marketplace. Supportive environment that enables employees to perform continuous improvement and innovation and have the ability to re-configure (Sherehiy et al., 2007).
Cooperative Teams (Team Effort)	Collect cross-functional effort for decision-making and product development. Knowledge in teamwork and work in multi-functional workforce (Gunasekaran, 1999).

Table 1. Agile Organization Concepts

Organization learning is recognised to be an important organizational concept for agility since it influences the other agility concepts (Plonka, 1997). The ability for continuous learning to be able to adapt allows employees to find the best solutions to fulfil the organization's goals and objectives in a rapidly changing business environment (Giesecke and McNeil, 2004). Organizational learning promotes employees to unlock individual creativity and knowledge creation within the an adaptive and flexible organizational structure (Dasgupta and Gupta, 2009).

In software development projects, organizational learning based on incremental approach can help software development teams to capture and maintain necessary knowledge (Kavitha and Ahmed, 2011). The learning focus in organizations develop understanding on how the speed of development is related to other goals such as the cost, risk, quality and innovations in projects (Mathiassen and Pries-Heje, 2006)

Table 2 presents the organizational agility concepts (identified in Table 1) mapped with agile software development practices identified through literature review on various agile software development practices. A discussion was done with an industry individual who has been an engineering manager with agile software development since 2001 to confirm if the mapping (Table 2) of the organisational agility concepts underpin the agile software development practices. Hence, Table 2 identifies the specific organisational agility concepts that ought to drive agile software development practices if organizational agility with software development is the main goal.

Organizationial Agility Concepts	Agile Software Development Practices
Workforce Agility	Self-organizing teams, daily stand-up meeting sprint planning, retrospective, user stories.
Competencies	Continuous integration, self-organizing teams, product backlog, test-driven development, user stories, unit testing, user story acceptance testing.
Speed	Sprint planning, product backlog, test-driven development, continuous integration, refactoring.
Responsiveness	Product backlog, release planning, sprint planning, retrospective, user story acceptance testing.
Knowledge Management	Pair programming, test-driven development, retrospective, user story acceptance testing, regression testing, refactoring, coding standard
Organizational Learning	Vision planning, retrospective, release planning, sprint planning, retrospective, test-driven development
Organizational Culture	Open workspace, coding standard, collective code ownership, retrospective, user story acceptance testing
Cooperative Teams	Vision planning, product backlog, Daily stand-up meeting, Pair programming, Sprint planning, Coding standard, retrospective.

Table 2. Agile Organization Concepts Related to Agile Software Development Practices

Source: (Pichler, 2010), (Schwaber and Sutherland, 2012), (Sivanantham, 2012), (Donaldson and Siegel, 2001), (Shalloway, Beaver, and Trott, 2009), (Agarwal, Karimpour, and Ruhe, 2014), (Danesh, 2011) (Chromatic, 2003), (Cockburn and Highsmith, 2001), (Hunt, 2006), (Farcic and Garcia, 2015), (Leffingwell, 2010), (Schiel, 2009), (Resnick, Bjork, and de la Maza, 2011), (Resnick et al., 2011), (Fowler et al., 2012), (Chromatic, 2003), (Kelly, 2015), (Pollard, 2016), (Cohn, 2004), (Monochristou and Vlachopoulou, 2007), (Babar, Brown, and Mistrik, 2013), (Schifferstein and Hekkert, 2011), (Duvall, Matyas, and Glover, 2007), (Moreira, 2013), (Hoda et al., 2013), (Lewis, 2016).

For this study, the agility concepts investigated to identify agility practices for software development include workforce agility, competencies knowledge management, speed, responsiveness, organisational learning, organisational culture, and cooperative teams. Hence, following hypothesis were tested:

Hypothesis 1: Knowledge management positively affects the organizational learning of agile software development teams.

Hypothesis 2: Organizational culture positively affects the Organizational Learning of software development teams.

Hypothesis 3: Organizational Learning positively affects the Competencies of software development teams or organizations.

Hypothesis 4: Organizational Learning positively affects the Responsiveness Behaviour of agile software development teams or organizations.

Hypothesis 5: Organizational Learning positively affects the Speed of agile software development teams or organizations.

Hypothesis 6: Organizational learning positively affects the Team Effort of agile software development teams or organizations.

Hypothesis 7: Organizational Learning positively affects the Workforce agility of software development teams or organizations.

3 DATA COLLECTION

The survey method was used to involve widely targeted participants from New Zealand and Australia. The sampling framework for this study was based on involving software organizations or development teams using agile development approach or methods. The companies were identified through an index website that contains the software company name and contact information in New Zealand and Australia. They were invited through an email. Some of the potential participants' information was also acquired from the contact information of 2016 NZ agile conference participants. Respondents were also identified through the agile community groups and agile software development conferences held in New Zealand and Australia.

The survey questions were developed based on the literature review and addressed the 7 hypotheses. Initially, it had 120 questions. After revision and improvement, the survey had 76 questions. Minor revisions were done based on a pilot study conducted with three experts in agile software development.

In the invitation email to the participants it was indicated that the study was part of a master's thesis and the objective was to investigate and identify the agility practices with agile software development. It also informed participants that the survey was anonymous and voluntary. In addition, their identities would not be included in study report. The survey was made available online on Google Survey between May 2017 and July 2017. The survey was closed on 10th July 2017.

3.1 Data Analysis Techniques

After data screening for quality and accuracy, the dataset was analysed using Structural Equation Modelling (SEM). This study used Partial Least Squared (PLS) since the research model has formative constructs (Chin and Newsted, 1999).

This survey had a 35% success rate in terms of actual participation. There were 181 invitations emailed out to the prospective participants (in-house agile software development teams, agile software development contracting companies and agile software vendors), mostly in New Zealand and some in Australia. From the 63 responses, 58 were screened to be suitable for data analysis. With 58 responses from different agile software development organizations, we had a sufficient number of responses for data analysis to able to identify agile practices for gaining agility in software development. The 33% of respondents were from in-house software development teams, 43% of the respondents were software vendor organizations and 24% were from contracting software development organizations.

Our survey results showed that for in-house software development 89% of respondents were part of a large in-house agile software development teams. This includes 47% of respondents' part of 10 to 50 member teams and 32% over 50 member teams. For software vendors 80% (68% 10 to 50 team members + 12% had over 50 team members) of the respondents were part of large agile software development teams. For contracting development teams, 64% of the respondents were part of large agile software development teams while 36% of respondents had less than 10 team members. The large number of respondents from large team sizes shows that agile software development is now regarded as a useful development approach for providing an effective work practices not only for small size teams but also for large size teams. When agile software development emerged, it was seen as a useful approach for small sized teams.

4 MEASUREMENT MODEL VALIDATION

To ensure the reliability and validity of the model the theoretical measurement is compared with the structural model and collected data. The reliability of the model was assessed using an internal consistency reliability technique, while the validity of the model was assessed using convergent validity and discriminant validity.

4.1 Internal Consistency Reliability

The result of Internal Composite Reliability (ICR) in Table 3 shows that all latent variables (agile organizations concepts) in the PLS model have internal consistencies greater than 0.7, showing the reliability of all the constructs (agile organizations concepts) used in this study. The reliability of the constructs was calculated separately and assessed independently. The value may vary between 0 and 1 with a minimum value of 0.7. while higher values mean higher reliability of the constructs (Hair et al., 2011).

Construct	Composite Reliability
Competencies	0.842
Knowledge Management	0.851
Organizational Culture	0.815
Organizational Learning	0.836
Responsiveness	0.739
Speed	0.838
Team Effort	0.855
Workforce Agility	0.749

Table 3. Composite reliability results

4.2 Convergent Validity

Table 4 provides the AVE result, calculated using the statistical analysis tool SmartPLS, which suggests that the constructs used in this study all have convergent validity, confirming the relationship between the indicators of each construct. According to Hair et al. (2011), the AVE value should be greater than 0.50 in order to be accepted.

Construct	**AVE**
Competencies	0.774
Knowledge Management	0.857
Organizational Culture	0.724
Organizational Learning	0.618
Responsiveness	0.760
Speed	0.768
Team Effort	0.797
Workforce Agility	0.605

Table 4 – Convergent Reliability Results

4.3 Discriminant Validity

Table 5 provides the results of discriminant validity testing showing the constructs are distinct from each other. The comparison is done by calculating the square root of the AVE values with the latent variable correlations. The recommended AVE value must be higher than the corresponding correlations among the latent variables (Hair et al., 2016). Table 4 shows the outer loadings and cross loadings of model constructs.

Construct	**COM**	**KM**	**OC**	**OL**	**RES**	**SPD**	**TE**	**WA**
Competencies (COM)	**0.807**							
Knowledge Management (KM)	0.529	**0.860**						
Organizational Culture (OC)	0.640	0.549	**0.774**					
Organizational Learning (OL)	0.670	0.661	0.629	**0.798**				
Responsiveness (RES)	0.479	0.648	0.672	0.658	**0.747**			
Speed (SPD)	0.656	0.549	0.542	0.712	0.475	**0.804**		
Team Effort (TE)	0.287	0.463	0.540	0.443	0.668	0.303	**0.823**	
Workforce Agility (WA)	0.493	0.611	0.501	0.616	0.572	0.501	0.419	**0.736**

Table 5. Discriminant Validity Results

4.4 Structural Model Validation

The outputs of PLS-SEM result are shown below in Figure 2 (page 7). The results of the validation tests suggest that the proposed model passed all the criteria for assessment in relevance with the path coefficient and model loadings or weights on their measured latent variables. In addition, the formative measurement models are established properly with data from convergent validity (Table 4).

4.5 Coefficient of Determination

To evaluate the structural model the coefficient of determination (R2 value) is provided in Table 6 showing our model as fairly accurate. The coefficient of determination measures the accuracy of the proposed model and allows comparison between the correlation of a specific construct's actual and predicted value (Hair et al., 2011). The value range for R2 is between 0 and 1, where a value over 0.2 can be considered as high.

Construct	**R2 Value**
Competencies	0.618
Organizational Learning	0.706
Responsiveness	0.602
Speed	0.681
Team Effort	0.643
Workforce Agility	0.537

Table 6. Coefficient of Determination (R2 Values) Results

Our R2 values shows that the strongest relationship is provided by organizational learning (0.706), followed by speed (0.681), then team effort (0.643), next is competencies (0.618), then responsiveness

(0.602), and finally workforce agility which has the lowest value. The R2 values provide a measurement of how the outcomes are replicated in the model which means organizational learning is the most critical organizational concept (Hair et al., 2016).

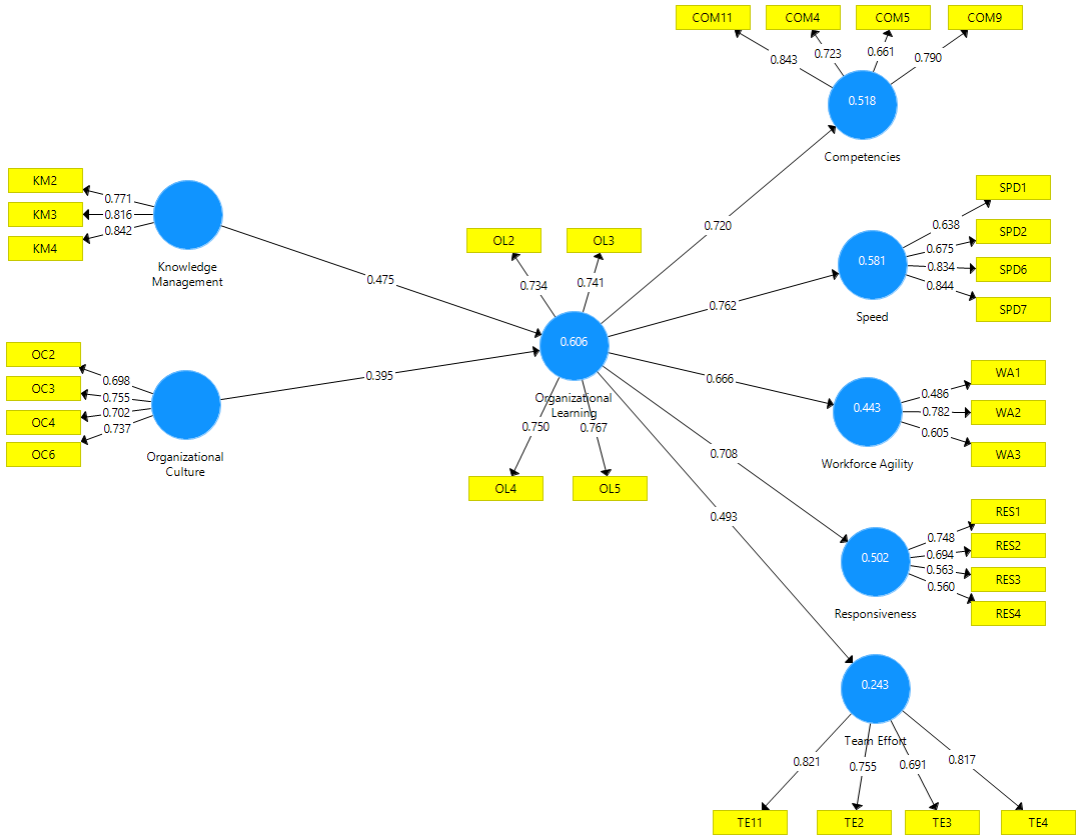

Figure 1. PLS-SEM proposed model results

4.5.1 Hypothesis testing results

Hypothesis	Path	Original Sample (O)	Sample Mean (M)	Standard Deviation (STDEV)	T-Values	P-Values
H1	KM→ OL	0.475	0.448	0.142	3.348	0.001
H2	OC → OL	0.395	0.420	0.103	3.816	0.000
H3	OL → COM	0.720	0.724	0.070	10.242	0.000
H4	OL → RES	0.708	0.727	0.080	8.819	0.000
H5	OL → SPD	0.762	0.763	0.070	10.908	0.000
H6	OL → TE	0.493	0.503	0.170	2.893	0.004

Table 7. Bootstrap Path Coefficient and T-Values in the Structural Model

The column, Original Sample (O), in Table 7 is used to determine how strong the relationship is between the hypothesized agile organization concepts in the structural model. According to Hair et al. (2011), the path coefficient values must range if the relationship is very weak (0 – 0.2), weak (0.2 – 0.4), moderate (0.4 – 0.6), strong (0.6 – 0.8) or very strong (0.8 – 1.0).

- *Hypothesis 1: Knowledge Management and Organizational Learning*

Table 7 shows a moderate relationship between knowledge management and organizational learning (Path = 0.475, t = 3.348, p<0.001). These results show that H1 is supported.

- *Hypothesis 2: Organizational Culture and Organizational Learning*

Table 7 shows a weak relationship between organizational culture and organizational learning (Path = 0.395, t = 3.816, p<0.000). These results indicate that H2 is also supported.

- *Hypothesis 3: Organizational Learning and Competencies*

Table 7 shows a strong relationship between organizational culture and organizational learning (Path = 0.720, t = 10.242, p<0.000). These results indicate that H3 is also supported.

- *Hypothesis 4: Organizational Learning and Responsiveness*

Table 7 shows a strong relationship between organizational learning and responsiveness (Path = 0.708, t = 8.819, p<0.000). These results indicate that H4 is also supported.

- *Hypothesis 5: Organizational Learning and Speed*

Table 7 shows a strong relationship between organizational learning and speed (Path = 0.762, t = 10.908, p<0.000). These results indicate that H5 is also supported.

- *Hypothesis 6: Organizational Learning and Team Effort*

Table 7 shows a moderate relationship between organizational learning and team effort (Path = 0.493, t = 2.893, p<0.004). These results indicate that H6 is also supported.

- *Hypothesis 7: Organizational Learning and Workforce Agility*

Table 7 shows a strong relationship between organizational learning and workforce agility (Path = 0.666, t = 5.904, p<0.000). These results indicate that H7 is also supported.

4.5.2 Organizational agility practices for agile software development

Table 8 presents a list of organizational agility practices identified from the result of the survey undertaken to test the hypothesis. The percentage, obtained through the analysis of the survey data, indicate support for a practice. These practices that ought to be part of agile software development to enable organizational agility with software development, answering the research question that was:

"What are the software development agility practices based on agile organization concepts?"

Agile Organization Concepts	**Agile Software Development Practices for Organizational Agility**
Knowledge Management	Teamwork to enhance critical thinking and problem-solving abilities. (94.7%) Task sharing practice vital to build knowledge and experience. (91.4%) Method practice must build knowledge for undertaking next project. (86.2%)
Organizational Culture	Spontaneous collaboration must be a critical culture. (89.7%). Individuals must take part in a wide variety of task and decision-making. (89.6%) Collective decision-making as an essential practice. (91.3%) Collective responsibility is a key work practice. (89.7%)
Organizational Learning	Continuous reflections on practices, team structure, and roles. (91.4%) Continuous learning to enhance firm-specific core competency. (89.7%) The team-effort, ownership and cross-functional cooperation enforced through adoption of practices mutually accepted by all stakeholders. (93.1%) Mind-set for continuous learning through projects. (91.4%)
Competencies	Quality driven development incorporating continuous integration. (87.9%) Individual ability to perform in formal and informal roles. (93.1%) Technological ability and reliable development infrastructure. (84.5%) Flexible development systems (short development cycles). (82.8%)
Responsiveness	Empowerment as a critical factor to learn and adapt. (94.9%) Ability to accept change any time. (89.7%). Ability to adopt and adapt on the fly new practices and skills. (94.8%) The mind-set for feedback to deliver useful features. (82.7%)
Speed	Team only works the normal hours on a consistent basis. (91.3%) Development infrastructure must not place limitations on work pace. (82.8%) Avoiding technical debt is a critical mind set. (89.7%) Upfront testing is essential practice to deliver software in short cycles. (86.2%)
Team Effort	Product planning must have input from the software engineering. (96.6%) Project planning must be based on a cross-functional effort. (93.1%) Work item reviews with stakeholders to enable strategic benefits. (91.4%) Team participation to achieve reliable plans. (94.8%)
Workforce Agility	Teams must consist of individuals with appropriate skills and knowledge to carry out multiple tasks. (98.3%) Development team consists of highly skilled and competent individuals. (96.6%) Development team must have negotiation and consensus capabilities to accept change, generate new ideas and accept new responsibilities. (91.4%)

Table 8. Agile software development practices for organizational agility with Software Development.

5 DISCUSSION

The statistical analysis of our data shows that all the hypotheses are supported meaning that the theoretical concepts of organizational agility specifically, knowledge management, organizational culture, organizational learning, competencies, responsiveness, speed, team effort, and workforce agility are also vital elements for achieving software development agility.

In addition, our survey results show that 91% of the respondents reported project success of 76% to 100%. Of this result, 43% of the respondents reported having 76% to 90% success rate while another 48% had 91% to 100% success rate. Our success rate findings are similar to Versionone (https://www.stateofagile.com/#ufh-c-473508-state-of-agile-report) research. According to their 12th annual report, 98% of their respondents indicated success with agile development projects. Specifically, 74% indicated more than half of their agile projects were successfully delivered. In their 13th state of agile report it shows that 48% reported most or all agile projects were successful.

Our investigation supports that organizational learning as an important factor for the organisational agility in software development as analysis of the data shows that it has the highest R2 value (0.706) (Table 6) compared to six other agility elements (speed, team effort, competencies, responsiveness, and workforce agility). Other researchers have identified organisational learning is critical to achieve business goals through having the ability to adapt and develop best software solutions, enabling creativity and knowledge creation, and to develop tacit knowledge and speed required for development (Giesecke and McNeil, 2004; Dasgupta and Gupta, 2009; Mathiassen and Pries-Heje, 2006). Continuous learning ability improves organization's performance by having capacity to recognise and adapt to changing situations for better results, encourage cooperation and get highly valued feedback (Iivari and Iivari, 2011; Sherehiy et al., 2007). Hence, organisational learning ability is critical factor to identify and swiftly deal with the necessary changes in software development environment (Lyytinen and Rose, 2006). Our investigation supports this argument since our result shows a strong positive relationship between organisation learning and speed (H5).

This study also shows organisational learning has a strong relationship with majority of the organizational agility elements (knowledge management, competencies, responsiveness, speed, and workforce agility). However, organisational learning has weak relationship with organizational culture and a moderate relationship with the team effort. There are possible explanations for the weak and moderate relationships. According to Lal (2011) and Lal and Clear (2017) instead of achieving a mutual benefit for all stakeholders, agile software development in reality benefits the product management team rather than the development team. This is due to projects driven by short development cycles i.e. sprint after sprint where the focus is on swift development with little or no time spent for in-depth investigation (running spikes) for deep understanding or minimising risks with user stories with little time left for reflection or reviews for learning limiting the agility capability. In addition, the product backlog may not necessarily capture everything the engineers do in short development cycles. While sprints are planned based on the product backlog items, there is no clear visibility on actual work required to be done in projects, limiting development agility.

Organizational agility culture is based on coordination, collaboration and informal communication which must also provide mutual benefits (Lal, 2011). This allows for the needed cross-functional support to have competitive advantage. However, agile approach is limited to software engineering level only rather than encompassing the entire organisation. Hence, this limits the organizational agility culture spreading in the entire organisation and limits the organizational ability for development agility.

This study supports that organizational agility capability is a critical element for software development success. The agile manifesto requires to be supported with organizational agility concepts so that software engineering community can sustain on-going development success through organizational agility. This study identifies the necessary organizational agility practices for agile software development (Table 8) when adopting agile approach and related structures (functional units and roles).

5.1 Limitations, theoretical and practical contributions

With 35% success rate with actual participation (58 from 63 responses) generalisability of this study findings may not be considered applicable to agile software development. However, our respondents represent three ways (in-house, contracting companies and software vendors) the software is most likely to be developed. The research outcomes echo the means by which organizational agility with agile software development is achieved. This study adds to the literature by identifying several organizational agility concepts such as knowledge management organizational culture, organizational learning, competencies responsiveness, speed, team effort, and workforce agility as critical for agile software

development. The specific organizational agility practices for software development is provided. Therefore, we believe these findings will be of value to practitioners and for future research.

6 CONCLUSION

Through an empirical investigation involving survey method we have identified several organizational agility practices for agile software development. These agility practices support the agile manifesto for adopting agile approach and related structures. This paper adds to the limited body of knowledge on theoretical perspectives used to research on agile approach including on its core philosophy of development agility (Dingsoyr et.al., 2012). We found a rich set of organizational agility concepts and related practices with agile software development independent of agile manifesto and development approach to continuously deliver software in a dynamic business environment. Without being driven by agility mind set development success will be limited since there will be limited understanding and required ability to learn and adjust the structures and work competency in the entire organisation.

7 REFERENCES

Agarwal, N., Karimpour, R., and Ruhe, G. 2014. "Theme-based product release planning: An analytical approach". *Symposium conducted at the meeting of the System Sciences (HICSS)*.

Babar, M. A., Brown, A. W., and Mistrik, I. 2013. *Agile Software Architecture: Aligning Agile Processes and Software Architectures*: Elsevier Science.

Bernardes, S., and Hanna,D. 2009. "A theoretical review of flexibility, agility and responsiveness in the operations management literature." *International Journal of Operations and Production Management, 29*:1, pp. 30-53.

Breu, K., Hemingway, C. J., Strathern, M., and Bridger, D. 2002. "Workforce agility: the new employee strategy for the knowledge economy." *Journal of Information Technology,* 17:1, pp. 21-31.

Chin, W. W., and Newsted, P. R. 1999. "Structural equation modeling analysis with small samples using partial least squares." *Statistical strategies for small sample research, 2*, pp. 307-342.

Chromatic. 2003. *Extreme Programming Pocket Guide*: O'Reilly Media.

Cockburn, A., and Highsmith, J. 2001. "Agile software development, the people factor". *Computer* 34:11, pp. 131-133.

Cohn, M. 2004. *User Stories Applied: For Agile Software Development (Adobe Reader)*: Pearson Education.

Danesh, A. S. 2011. "A survey of release planning approaches in incremental software development". In *Computational Intelligence and Information Technology* (pp. 687-692): Springer.

Dasgupta, M., and Gupta, R. 2009. "Innovation in organizations: A review of the role of organizational learning and knowledge management." *Global Business Review,* 10:2, pp. 203-224.

Dingsoyr, T., Nerur, S. and Moe, B.N. 2012. "A decade of agile methodologies: towards explaining agile software development" *The journal of Systems and Software*. 85. pp. 1213-1221.

Donaldson, S. E., and Siegel, S. G. 2001. Successful Software Development: Prentice Hall PTR. Retrieved from https://books.google.co.nz/books?id=lrix5MNRiu4C

Dyer, L., and Shafer, R. A. 2003. "Dynamic organizations: Achieving marketplace and organizational agility with people.", *CAHRS Working Paper Series*

Farcic, V., and Garcia, A. 2015. *Test-Driven Java Development*: Packt Publishing.

Giesecke, J., and McNeil, B. 2004. Transitioning to the learning organization.

Goldman, S. L., Nagel, R. N., and Preiss, K. 1995. *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer*: Van Nostrand Reinhold.

Gunasekaran, A. 1998. "Agile manufacturing: Enablers and an implementation framework." *International Journal of Production Research,* 36:5, pp. 1223-1247.

Hair, J. F., Ringle, C. M., and Sarstedt, M. 2011. "PLS-SEM: Indeed a silver bullet." *Journal of Marketing theory and Practice,* 19:2, pp. 139-152.

Highsmith, J. A. 2002. *Agile Software Development Ecosystems*: Addison-Wesley.

Hoda, R., Noble, J., and Marshall, S. 2013. "Self-Organizing Roles on Agile Software Development Teams". *IEEE Transactions on Software Engineering,* 39:3, pp. 422-444.

Hunt, J. (2006). *Agile Software Construction*: Springer London.

Iivari, J., and Iivari, N. 2011. "The relationship between organizational culture and the deployment of agile methods." *Information and Software Technology,* 53:5, pp. 509-520.

Jacobson, I. and Spence, I. 2009. Why we need a theory for Software engineering. Dr Dobb's Journal.

Lal, R. and Clear, T. 2018. "Enhancing Product and Service Capability Through Scaling Agility in a Global Software Vendor environnement." *13th ICGSE International Conference on Global Software engineering 2018 conference. Sweden*.

Lal, R. 2011. "Strategic factors in Agile Software Development method Adaptation: A Study of Market-Driven Organisation", (Doctoral dissertation, Massey University, Auckland, New Zealand). Retrieved from http://mro.massey.ac.nz/handle/10179/2496.

Lal, R. and Clear, T. 2017. "Scaling Agile at the Program Level in an Australian Software Vendor Environment: A Case Study." *The 28th Australasian Conference on Information Systems, University of Tasmania in Hobart, Australia*.

Lee, G., and Xia, W. 2010. "Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility." *Mis Quarterly,* 34:1, pp. 87-114.

Leffingwell, D. 2010. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*: Pearson Education.

Lewis, W. E. 2016. *Software Testing and Continuous Quality Improvement, 3rd Edition*. CRC Press.

Lin, C.-T., Chiu, H., and Chu, P.-Y. 2006. "Agility index in the supply chain." *International Journal of production economics, 100*:2, pp. 285-299.

Mason-Jones, R., Naylor, B., and Towill, D. R. 2000. "Engineering the leagile supply chain." *International Journal of Agile Management Systems, 2*:1, pp. 54-61.

Mathiassen, L., and Pries-Heje, J. 2006. Business agility and diffusion of information technology: Springer.

Monochristou, V., and Vlachopoulou, M. 2007. "Requirements specification using user stories." *Agile Software Development Quality Assurance, 71*.

Moreira, M. E. 2013. *Being Agile: Your Roadmap to Successful Adoption of Agile*: Apress.

Pichler, R. 2010. *Agile Product Management with Scrum: Creating Products that Customers Love (Adobe Reader)*: Pearson Education.

Plonka, F. E. 1997. Developing a lean and agile work force. *Human Factors and Ergonomics in Manufacturing and Service Industries, 7*:1, pp. 11-20.

Pollard, N. 2016. *Getting Started with Agile Software Development*: BookRix.

Schmidt, R., Lyytinen, K., and Mark Keil, P. C. 2001. "Identifying software project risks: An international Delphi study." *Journal of management information systems, 17*:4, pp. 5-36.

Schwaber, K., and Sutherland, J. 2012. *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*: Wiley.

Shalloway, A., Beaver, G., and Trott, J. R. 2009. *Lean-Agile Software Development: Achieving Enterprise Agility*: Pearson Education.

Sharifi, H., and Zhang, Z. 1999. "A methodology for achieving agility in manufacturing organisations: An introduction". *International Journal of production economics, 62*:1, pp. 7-22.

Sherehiy, B., Karwowski, W., and Layer, J. K. 2007. "A review of enterprise agility: Concepts, frameworks, and attributes." *International Journal of industrial ergonomics,* 37: 5, p. 445-460.

Sivanantham, V. 2012. Knowledge Management in Agile Projects. *Cognizant 20-20 Insights*.

West, D., and Grant, T. 2010. Agile development: Mainstream adoption has changed agility.

Yusuf, Y. Y., Sarhadi, M., and Gunasekaran, A. 1999. Agile manufacturing:: The drivers, concepts and attributes. *International Journal of production economics,* 62:1, p. 33-43.