1 March 2019

ONLINE APPENDIX

Table A1: Selected Related Studies

Study	Focal Independent Variables	Level of Analysis	Dependent Variables	Type of Study and Data	Key Finding
Balijepally, Mahapatra, Nerur, and Price (2009)	Pair programming	Dyad	Software quality, satisfaction, confidence	Lab experiment with students working on a software development task	Pairs showed higher satisfaction than developers working alone, but the quality produced by pairs and the confidence of pair programmers did only surpass that of the second-best member of a "nominal pair."
Mangalaraj, Nerur, Mahapatra, and Price (2014)	Pair programming	Dyad	Quality of solution, task completion time, task satisfaction	Lab experiment with software professionals working on a software design task	Paired software designers had higher task satisfaction than solo designers. The quality of the solution of paired members did only surmount the second-best member of a nominal pair and the completion time was lower than any member of a nominal pair.
Porter, Hollenbeck, Ilgen, Ellis, West, and Moon (2003)	Legitimacy of need, personality dimensions	Team	Backup behavior	Lab experiment with university students in a non-software development context (teams of four in a combat simulation)	The legitimacy of the need for backup behavior affects the backup behavior provided in teams. This effect is moderated by the personality traits of backup providers and receivers.
Porter (2005)	Learning orientation, performance orientation	Team	Backup behavior, task performance, efficacy, commitment	Lab experiment with university students in a non-software development context (teams of four in a combat simulation)	Learning orientation has a positive effect on backup behavior, efficacy, and commitment. Performance orientation interacts with task performance to positively effect efficacy.
This Study	*Pair programming*	*Team*	*Team performance*	*Software developers, Scrum master, and product owners of 62 software development teams working on software products in one of the largest enterprise software firms worldwide*	*The practice of pair programming helps teams establish backup behavior through contributing to shared mental models. The negative effect of task novelty on team performance is attenuated by backup behavior.*

Notes:

The selection of studies in this table is not meant to be representative but illustrative. The studies by Balijepally et al. (2009) and Mangalaraj et al. (2014) are, to our knowledge, the only empirical IS studies in top-tier outlets focusing on performance effects of pair programming. The papers by Porter et al. (2003) and Porter (2005) are examples for organizational behavior/social psychology studies on the antecedents of backup behavior in teams. Other studied antecedents of backup behavior include team training (Fiscella, Mauksch, Bodenheimer, and Salas 2017; Shuffler, DiazGranados, and Salas 2011), and team composition and climate (Kozlowski and Ilgen 2006; Smith-Jentsch, Kraiger, Cannon-Bowers, and Salas 2009). These studies are often lab experiments and rely on teamwork simulations.

Related work includes the following three types of studies:
(1) The large body of literature reporting experimental studies on pair programming at the dyadic level (see e.g., Di Bella, Fronza, Phaphoom, Sillitti, Succi, and Vlasenko 2013; Dybå, Arisholm, Sjoberg, Hannay, and Shull 2007; Hannay, Dybå, Arisholm, and Sjøberg 2009; Vanhanen and Mäntylä 2013).
(2) Quantitative studies on team-level performance effects of agile software development in general (Lee and Xia 2010; Maruping, Venkatesh, and Agarwal 2009). These studies either lump together several software development practices, including pair programming, or studied agility as a capability as opposed to a behavior. Also, these studies focused on direct and moderated performance effects of agile development and did not examine the mechanisms through which these effects are carried.
(3) Studies reporting analytical modelling related to dyadic-level effects of pair programming (Dawande, Johar, Kumar, and Mookerjee 2008).

We conceptualize backup behavior following Porter et al. (2003). Prior work suggests two different conceptualizations of backup behavior. On the one hand, Marks, Mathieu, and Zaccaro (2001) conceptualize backup behavior in a broader sense as "assisting team members to perform their task" (Marks et al. 2001, p. 367), which may occur by providing a teammate feedback and coaching, assisting a teammate in carrying out actions, and completing tasks for a teammate (Marks et al. 2001). On the other hand, Porter et al. (2003) provide a narrower definition of backup behavior as "the discretionary provision of resources and task-related effort to another member of one's team that is intended to help that team member obtain the goals as defined by his or her role when it is apparent that the team member is failing to reach those goals" (Porter et al. 2003, pp. 391-392). Thus, whereas Marks et al. (2001) include feedback on a teammate's (i.e., somebody else's) work outcomes as a component of backup behavior, Porter et al. (2003) focus on the actual resource provision and task-related effort, i.e., the teammate's own constructive contributions. Given our focus on task novelty, which implies that team members lack required capabilities and processes, we follow Porter and others, and focus on backup behavior in the sense of supporting teammates with task-related effort (Barnes, Hollenbeck, Wagner, DeRue, Nahrgang, and Schwind 2008; Burke, Stagl, Salas, Pierce, and Kendall 2006; Porter, Gogus, and Yu 2010). Also, considering that software development teams often draw on a broad set of capabilities (Faraj and Sproull 2000), we do not view backup behavior with a sole focus on overcoming workload issues, and instead consider it more generally to apply to situations where a teammate is "unable to fulfill his or her role," and where those providing backup would have the "time, resources, and capacity to help their teammates" (Rousseau, Aubé, and Savoie 2006, p. 554).

Table A2: Correlations

Variable	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Architectural modularity	1.00												
2. Backup behavior	-0.14	1.00											
3. Daily meetings	0.09	-0.03	1.00										
4. Number of teams in the same project	-0.27*	-0.03	-0.22	1.00									
5. Pair programming	0.19	0.32*	-0.05	-0.13	1.00								
6. Progr. language: C/C++/C##	0.09	-0.18	-0.04	-0.11	0.04	1.00							
7. Progr. language: Java/JavaScript	0.22	-0.22	0.04	-0.11	0.16	0.11	1.00						
8. Progr. language: Proprietary	-0.14	-0.07	-0.12	0.07	-0.23	-0.20	-0.36**	1.00					
9. Relational team cohesion	-0.02	0.35**	-0.14	-0.04	0.10	0.07	0.06	-0.08	1.00				
10. Shared mental models	0.06	0.43**	-0.17	0.07	0.32*	0.10	0.11	-0.06	0.37**	1.00			
11. Task novelty	0.09	-0.02	-0.29*	0.06	0.09	-0.16	0.04	0.04	-0.05	-0.21	1.00		
12. Team performance	0.37**	-0.03	-0.08	-0.20	-0.03	0.00	0.22	0.04	0.38**	0.13	-0.17	1.00	
13. Team member turnover	0.11	-0.08	-0.03	0.00	-0.07	0.14	0.21	-0.02	0.01	-0.10	0.24	0.12	1.00
14. Team size	0.00	-0.12	0.03	0.03	-0.03	0.00	0.06	0.28*	0.13	0.02	0.11	0.02	0.40**

*Notes: ** p<0.01, * p<0.05*

Table A3: Simultaneous Equation Models Using 3SLS (n=62)

	(1) Pair programming	(2) Shared mental models	(3) Backup behavior	(4) Team performance
Pair programming		4.47**	4.05**	-2.73
		(0.73)	(1.20)	(2.97)
Shared mental models			1.44**	0.46
			(0.36)	(0.97)
Backup behavior				-0.32
				(0.41)
Task novelty				-0.22
				(0.14)
Backup behavior × task novelty				0.62**
				(0.24)
Daily meetings	-0.01	0.01	0.13	
	(0.02)	(0.06)	(0.07)	
Number of teams in the same project	-0.01	0.03	-0.00	
	(0.01)	(0.02)	(0.03)	
C/C++/C#	0.00	0.16	-0.65*	
	(0.08)	(0.25)	(0.32)	
Java/JavaScript	0.03	0.03	-0.56**	
	(0.04)	(0.13)	(0.17)	
Proprietary language	-0.11	0.47*	0.25	
	(0.06)	(0.19)	(0.25)	
Relational team cohesion	0.01	0.14	-0.07	0.58**
	(0.02)	(0.08)	(0.11)	(0.21)
Team member turnover	-0.03	0.05	0.25*	
	(0.02)	(0.08)	(0.10)	
Team size	0.01	-0.03	-0.07	
	(0.01)	(0.03)	(0.04)	
Architectural modularity	-0.01	-0.05		0.47**
	(0.01)	(0.06)		(0.14)
Constant	0.36	3.49**	-3.05	-2.25
	(0.23)	(0.73)	(1.60)	(4.28)
Chi2	7.63	48.89**	112.01**	45.26**

Notes: Standard errors in parentheses; ** $p<0.01$, * $p<0.05$

As is usual in models based on a system of equations, we made some identifying assumptions to estimate this model and to some extent we were assisted by results of our single-equation models for the plausibility of these exclusion restrictions. Note that one of the limitations of these system models is that they require imposing exclusion restrictions for identification (Greene 2000). Nonetheless, the results of 3SLS model are broadly similar to the results reported earlier.

REFERENCES

Balijepally, V., Mahapatra, R., Nerur, S., and Price, K. H. 2009. "Are Two Heads Better Than One for Software Development? The Productivity Paradox of Pair Programming," *MIS Quarterly* (33:1), pp. 91-118.

Barnes, C. M., Hollenbeck, J. R., Wagner, D. T., DeRue, D. S., Nahrgang, J. D., and Schwind, K. M. 2008. "Harmful Help: The Costs of Backing-Up Behavior in Teams," *Journal of Applied Psychology* (93:3), pp. 529-539.

Burke, C. S., Stagl, K. C., Salas, E., Pierce, L., and Kendall, D. 2006. "Understanding Team Adaptation: A Conceptual Analysis and Model," *Journal Of Applied Psychology* (91:6), pp. 1189-1207.

Dawande, M., Johar, M., Kumar, S., and Mookerjee, V. S. 2008. "A Comparison of Pair Versus Solo Programming Under Different Objectives: An Analytical Approach," *Information Systems Research* (19:1), pp. 71-92.

Den Hartog, D. N., De Hoogh, A. H. B., and Keegan, A. E. 2007. "The Interactive Effects of Belongingness and Charisma on Helping and Compliance," *Journal of Applied Psychology* (92:4), pp. 1131-1139.

Di Bella, E., Fronza, I., Phaphoom, N., Sillitti, A., Succi, G., and Vlasenko, J. 2013. "Pair Programming and Software Defects--A Large, Industrial Case Study," *IEEE Transactions on Software Engineering* (39:7), pp. 930-953.

Dybå, T., Arisholm, E., Sjoberg, D. I. K., Hannay, J. E., and Shull, F. 2007. "Are Two Heads Better than One? On the Effectiveness of Pair Programming," *IEEE Software* (24:6), pp. 12-15.

Edmondson, A. 1999. "Psychological Safety and Learning Behavior in Work Teams," *Administrative Science Quarterly* (44:2), pp. 350-383.

Faraj, S., and Sproull, L. 2000. "Coordinating Expertise in Software Development Teams," *Management Science* (46:12), pp. 1554-1568.

Fiscella, K., Mauksch, L., Bodenheimer, T., and Salas, E. 2017. "Improving Care Teams' Functioning: Recommendations from Team Science," *Jt Comm J Qual Patient Saf* (43:7), pp. 361-368.

Greene, W. H. 2000. *Econometric Analysis*, (4th ed.). Upper Saddle River, NJ: Prentice Hall.

Guinan, P. J., Cooprider, J. G., and Faraj, S. 1998. "Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach," *Information Systems Research* (9:2), pp. 101-125.

Hannay, J. E., Dybå, T., Arisholm, E., and Sjøberg, D. I. 2009. "The Effectiveness of Pair Programming: A Meta-Analysis," *Information and Software Technology* (51:7), pp. 1110-1122.

He, J., Butler, B. S., and King, W. R. 2007. "Team Cognition: Development and Evolution in Software Project Teams," *Journal of Management Information Systems* (24:2), pp. 261-292.

Kozlowski, S. W., and Ilgen, D. R. 2006. "Enhancing the Effectiveness of Work Groups and Teams," *Psychological Science in the Public Interest* (7:3), pp. 77-124.

Lau, A. K. W., Yam, R. C. M., and Tang, E. 2009. "The Complementarity of Internal Integration and Product Modularity: An Empirical Study of their Interaction Effect on Competitive Capabilities," *Journal of Engineering and Technology Management* (26:4), pp. 305-326.

Lawler, E. J., and Yoon, J. 1996. "Commitment in Exchange Relations: Test of a Theory of Relational Cohesion," *American Sociological Review* (61:1), pp. 89-108.

Lee, G., and Xia, W. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data," *MIS Quarterly* (34:1), p. 7.

Levesque, L. L., Wilson, J. M., and Wholey, D. R. 2001. "Cognitive Divergence and Shared Mental Models in Software Development Project Teams," *Journal of Organizational Behavior* (22), pp. 135-144.

Mangalaraj, G., Nerur, S., Mahapatra, R., and Price, K. H. 2014. "Distributed Cognition in Software Design: An Experimental Investigation of the Role of Design Patterns and Collaboration," *MIS Quarterly* (38:1), pp. 249-274.

Marks, M. A., Mathieu, J. E., and Zaccaro, S. J. 2001. "A Temporally Based Framework and Taxonomy of Team Processes," *Academy of Management Review* (26:3), pp. 356-376.

Maruping, L. M., Venkatesh, V., and Agarwal, R. 2009. "A control theory perspective on agile methodology use and changing user requirements," *Information Systems Research* (20:3), pp. 377-399.

Nidumolu, S. 1995. "The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable," *Information Systems Research* (6:3), pp. 191-219.

Porter, C. O. 2005. "Goal Orientation: Effects on Backing up Behavior, Performance, Efficacy, and Commitment in Teams," *Journal of Applied Psychology* (90:4), pp. 811-818.

Porter, C. O. H. L., Hollenbeck, J. R., Ilgen, D. R., Ellis, A. P. J., West, B. J., and Moon, H. 2003. "Backing Up Behaviors in Teams: The Role of Personality and Legitimacy of Need," *Journal of Applied Psychology* (88:3), pp. 391-403.

Porter, C. O. L. H., Gogus, C. I., and Yu, R. C.-F. 2010. "When Does Teamwork Translate Into Improved Team Performance? A Resource Allocation Perspective," *Small Group Research* (41:2), pp. 221-248.

Rosen, M. A., Bedwell, W. L., Wildman, J. L., Fritzsche, B. A., Salas, E., and Burke, C. S. 2011. "Managing Adaptive Performance in Teams: Guiding Principles and Behavioral Markers for Measurement," *Human Resource Management Review* (21:2), pp. 107-122.

Rousseau, V., Aubé, C., and Savoie, A. 2006. "Teamwork Behaviors: A Review and an Integration of Frameworks," *Small Group Research* (37:5), pp. 540-570.

Sakka, O., Barki, H., and Côte, L. 2013. "Interactive and Diagnostic Uses of Management Control Systems in IS Projects: Antecedents and Their Impact on Performance," *Information & Management* (50:6), pp. 265-274.

Salas, E., Sims, D. E., and Burke, C. S. 2005. "Is There a “Big Five” in Teamwork?," *Small Group Research* (36:5), pp. 555-599.

Shuffler, M. L., DiazGranados, D., and Salas, E. 2011. "There’s a Science for That: Team Development Interventions in Organizations," *Current Directions in Psychological Science* (20:6), pp. 365-372.

Smith-Jentsch, K. A., Kraiger, K., Cannon-Bowers, J. A., and Salas, E. 2009. "Do Familiar Teammates Request and Accept More Backup? Transactive Memory in Air Traffic Control," *Human Factors* (51:2), pp. 181-192.

Vanhanen, J., and Mäntylä, M. V. 2013. "A Systematic Mapping Study of Empirical Studies on the Use of Pair Programming in Industry," *International Journal of Software Engineering and Knowledge Engineering* (23:9), pp. 1221-1267.

Yates, D., Wagner, C., and Majchrzak, A. 2010. "Factors Affecting Shapers of Oganizational Wikis," *Journal of the Association for Information Science and Technology* (61:3), pp. 543-554.