Association for Information Systems
AIS Electronic Library (AISeL)

International Research Workshop on IT Project Management 2014

International Research Workshop on IT Project Management (IRWITPM)

12-13-2014

Measuring Coordination in Agile Software Development

Diane Strode
Whitireia Polytechnic, d.strode@whitireia.ac.nz

Measuring Coordination in Agile Software Development

Diane E. Strode
Whitireia Polytechnic
d.strode@whitireia.ac.nz

ABSTRACT

Coordination has long been recognized as contributing to successful IT projects. Agile software development provides many practices for achieving project coordination in small co-located projects. Given the importance of coordination to successful software development projects and the increasing popularity of agile software development, investigating coordination in this context is timely and potentially useful. This paper takes an existing theory of coordination in co-located agile software development projects developed from case study research and proposes a field test of that theory. The question addressed is *what is the effect of an agile coordination strategy on coordination effectiveness in co-located software development projects?* This paper describes the initial theory of coordination and a research design for field-testing that theory.

Keywords

Agile methods, co-located software development, coordination effectiveness, coordination strategy, coordination theory, explicit coordination, implicit coordination.

INTRODUCTION

Developing an information system is a group effort and effective group efforts require coordination. Coordination has been defined in many domains (Malone and Crowston, 1994). For example, in teamwork *"Coordination means the spatial and temporal synchronization of overt behaviors of two or more people so that those actions fit together into an intended spatial and temporal pattern*" (Arrow, McGrath, and Berdahl, 2000, p. 42). Coordination has long been recognized as contributing to successful large-scale IT projects (Curtis, Krasner, and Iscoe, 1988; Kraut and Streeter, 1995; Nidumolu, 1995) and more recently to projects distributed across countries, continents, and time zones (Cummings, Espinosa, and Pickering, 2009; Kotlarsky, van Fenema, and Willcocks, 2008).

Agile software development is an approach to information systems development that is particularly concerned with group endeavor. Agile software development is an umbrella term for any agile method, such as Scrum, Extreme Programming (XP), or any assemblage of agile practices (Conboy and Fitzgerald, 2007). Originally intended for software projects involving co-located teams, the approach is increasingly adapted for distributed project environments (Jalali and Wohlin, 2012; Sarker, Munson, Sarker, and Chakraborty, 2009). In the last 15 years the world-wide level of adoption has risen to include about 50% of software projects (Stavru, 2014) and shows no sign of abating. Given the importance of coordination to successful systems development and the increasing popularity of agile software development, research to investigate coordination in this context is timely and potentially useful.

Research focusing specifically on coordination in agile software development context is scant (Dingsoyr, Nerur, Balijepally, and Moe, 2012; Strode, Huff, Hope, and Link, 2012), although case studies (Chuang, Luor, and Lu, 2014) and ethnographies have identified coordination as an important element in agile projects (Mishra, Mishra, and Ostrovska, 2012; Pries-Heje and Pries-Heje, 2011; Sharp and Robinson, 2010). This research provides in-depth knowledge about how coordination occurs in selected agile projects but does not explain the relationship between the use of agile practices and effective software project coordination.

There is one theory focusing exclusively on coordination in co-located agile software development projects. This theory was developed inductively and systematically by Strode (2012) from four case studies of agile and non-agile software development projects. A refinement of the theory based on three case studies of agile software development was published by Strode, Huff, Hope and Link (2012). The theory proposes that the coordination effectiveness of an agile software development project is affected by the coordination strategy of the project. This theory, although carefully argued and supported with evidence from in-depth cases studies, has not yet been tested in a field study. This paper describes a research design for operationalizing and testing the concepts and relationships proposed in this theory. The broad research question this research addresses is *what is the effect of an agile coordination strategy on coordination effectiveness in co-located software development projects?*

This paper first summarises the theory proposed by Strode et al. (2012) and then describes the proposed research design for testing the theory in the field. The status of the research, potential contributions, and limitations of the study are discussed, and the paper concludes.

A THEORY OF COORDINATION IN AGILE SOFTWARE DEVELOPMENT PROJECTS

The coordination theory in agile software development projects proposed by Strode et al. (2012) was developed from independent cases of agile software development. Cases were selected because they showed a typical profile for co-located agile projects. Projects had a team size of 5 to 10, and the agile methods used were either Scrum or Scrum with XP practices since these are the most commonly adopted agile methods (Stavru, 2014). The theory has two primary theoretical concepts named coordination strategy and coordination effectiveness.

Coordination Strategy

The coordination strategy concept is concerned with the typical assemblage of agile practices that occurred in the cases. This concept has synchronisation, structure, and boundary spanning dimensions.

Synchronisation is a relation that exists when things occur at the same time, or are simultaneous (Allen, 1990, p. 1236). Synchronisation is achieved with synchronisation activities and synchronisation artefacts. A synchronisation activity involves all team members and brings them together at the same time and place for some pre-arranged purpose. A daily stand-up team meeting, a product demonstration, and a retrospective meeting are examples of synchronisation activities (for detailed descriptions of the most common agile practices see Williams (2010)). In the agile project context, synchronisation activities occur at different frequencies: once per project, once per iteration or sprint, daily, and ad hoc (as and when needed). For example, a project team meeting is held at the start of the project to discuss process, technical, and domain issues. At the start of each sprint, the agile project team holds a planning meeting to discuss progress, update the Scrum wallboard, and to decompose stories into tasks. Meetings are held daily where each project team member individually reports progress and problems. Ad hoc meetings, between the whole team and between sub-groups of the team, are held for planning and discussing issues as they arise. Synchronisation artefacts are produced and consumed (or used up) during synchronisation activities. Any physical thing generated during a synchronisation activity that contains information used by all team members in accomplishing their work is a synchronisation artefact. Typical synchronisation artefacts included story cards, task cards, a Scrum wallboard, and an automated test suite.

Structure is a second dimension of coordination strategy. Structure is defined in its common sense as the arrangement of and relations between the parts of something complex, and has sub-dimensions of proximity, availability, and substitutability. The agile methods Scrum and XP (Beck, 2000; Schwaber and Beedle, 2002) specify that close proximity of the project team, timely response to requests for help from within the team, and the sharing of work tasks among the team, as opposed to intense specialisation, are important for success in agile projects. Each of these sub-dimensions was identified in the agile cases on which the theory was based and they are defined as follows. Proximity is concerned with the physical closeness of individual team members; with adjacent desks providing the highest level of proximity. Availability occurs when team members are continually present and able to respond to requests for assistance or information. Substitutability occurs when team members are able to perform the work of another to maintain time schedules.

Boundary spanning is the third dimension of coordination strategy. Boundary spanning has three sub-dimensions including boundary spanning activities, the production of boundary spanning artefacts, and a coordinator role. In the theory, boundary-spanning activities can occur once per project, per iteration, and ad hoc. A boundary spanning activity is an activity performed by the team or the individual to elicit assistance or information from some unit or organisation external to the project to achieve project goals. A boundary-spanning artefact is produced to enable coordination beyond the team and project boundaries. An example is an email to request a new server from the IT support section. A coordinator role is taken by a project team member specifically to support interaction with people who are not part of the project team but who provide resources or information to the project. The agile methods Scrum and XP do not explicitly provide for boundary spanning (Beck, 2000; Schwaber and Beedle, 2002).

Nevertheless, in the cases on which the theory is based, interaction with other organisations and other business units within the organisation where the project was carried out, were important coordinative activities, therefore boundary spanning was included as a dimension of coordination strategy in agile software development projects.

Coordination Effectiveness

Coordination effectiveness is the outcome of a particular coordination strategy in Strode et al.'s (2012) theory. Coordination effectiveness was found to have an explicit and an implicit dimension. Coordination literature defines explicit coordination as that which occurs when two or more team members use overt mechanisms such as schedules, plans, and procedures, and send communication messages to one another using formal or informal, oral or written, transactions to integrate their work (Espinosa, Lerch, and Kraut, 2004; Nonaka, 1994; Rico, Sanchez-Manzanares, Gil, and Gibson, 2008). Coordination effectiveness was found to have three explicit dimensions: right thing, right place, and right time. That is, when a situation is well coordinated then the right things are in the right place, at the right time.

Implicit coordination occurs when team members anticipate the actions and needs of their colleagues and adjust their behavior accordingly without preplanning or direct communication (Nonaka, 1994; Rico et al., 2008). Five implicit dimensions of coordination effectiveness were identified in the theory. All project members need to 1) know why they are working on a task (or have a shared goal), 2) know what is going on and when, 3) know what to do and when, and, 4) know who is doing what, and 5) know who knows what.

Supporting Research

Prior research supports the existence of each dimension of Strode et al.'s (2012) coordination theory. This research has been carried out in a variety of contexts including agile software development projects. Table 1 shows examples of literature supporting each dimension of the theory and the context in which the research was carried out. Although research supports each dimension in the theory, there does not seem to be any extant theory or model that uniquely combines these concepts and dimensions, and proposes how they are related as Strode et al. (2012) have done in the agile software development context.

Dimension	Sub-dimension	Indicative literature	Context
Synchronisation	Synchronisation activity	Schmidt and Simone (1996) Salas, Sims, and Burke (2005) Arrow et al. (2000)	Cooperative work Teamwork Small groups
	Synchronisation artefact	Ren, Kiesler, and Fussell (2008)) Sharp and Robinson (2008)	Hospital settings Agile software development
Structure	Proximity	Hoegl and Proserpio (2004) Teasley, Covi, Krishnan, and Olson (2002)	Software development teams Software development
	Availability	Weick and Roberts (1993) Matook and Kautz (2008)	Flight operations Agile software development
	Substitutability	Salas et al. (2005)	Teamwork
Boundary spanning	Boundary spanning activity	Levina and Vaast (2005)	Information systems (IS) development
	Boundary spanning artefact	Levina and Vaast (2005)	IS development
	Coordinator role	Hoda, Noble, and Marshall (2010)	Agile software development
Implicit coordination	Know why/shared goal	Parolia, Goodman, Li, and Jiang (2007) Salas et al. (2005)	IS development Teamwork
	Know what is going on and	Yang, Kang, and Mason	Software development teams

	when	(2008)	
	Know what to do and when	Salas et al. (2005)	Teamwork
	Know who is doing what	Moe, Dingsoyr, and Dyba (2010)	Agile software development
	Know who knows what	Faraj and Sproull (2000) Lin, Hsu, Cheng, and Wu (2012)	Software development teams IS development teams
Explicit coordination	Right thing Right place Right time	Crowston and Osborn (2003)	Process mapping

Table 1 Supporting literature for the coordination concepts

The Relationship between Coordination Strategy and Coordination Effectiveness

A theory has theoretical concepts, associations between concepts, boundaries, and system states (Weber, 2012). Strode et al.'s (2012) theory of coordination has two associated concepts, and proposes that the coordination strategy of an agile project is associated with a level of coordination effectiveness. The most salient boundary of this theoretical system is the co-located agile software development project with additional restrictions, based on the cases informing the theory, as follows.

1. A software development project using practices from Scrum, or Scrum and Extreme Programming, with iterations (sprints) of one or two weeks.
2. An identifiable project team of 5 to 10 people working concurrently and full-time on the project, and who are located in close proximity within the same room in direct line of sight of one another.
3. A project with a clear business purpose that is either providing a software product for another business unit within the organisation or for an external organisation.
4. A project with a distinguishable customer or proxy customer. This can be a single person, a group, or groups of people. This customer can work within the team (physically sited with the team and involved in their daily work) or be an external party who is available for consultation.

The coordination strategy concept was found to have two states determined by the relationship of the customer with the team. The customer in an agile project is expected to be a knowledgeable and committed person closely and continuously involved with the project team on a daily basis (i.e. embedded) (Beck, 2000; Schwaber and Beedle, 2002). When the customer is embedded in the project then synchronisation activities and artefacts, and proximity, availability, and substitutability are sufficient to achieve coordination effectiveness. When the customer is external to project team, which is often what occurs in practice (Lohan, Conboy, and Lang, 2011), a project also requires boundary-spanning activities, artefacts, and roles to achieve effective coordination. This is because an external customer needs to be consulted more formally via pre-arranged meetings, and may need to prepare or read project documents to remain involved with and contribute to the project. In addition, a special role of coordinator might be arranged by the project team to facilitate interaction with the customer or other involved parties contributing to the project. These states lead to proposition 1.

Proposition 1. When the customer is embedded within the project team, a coordination strategy that includes synchronisation and structure coordination improves project coordination effectiveness. Synchronisation activities and associated artefacts are required at all frequencies (i.e. per project, per iteration, daily, and ad hoc). When the customer is an external party to the project then boundary spanning coordination is also needed. Boundary spanning activities and associated artefacts are required at all frequencies (i.e. per project, per iteration, and ad hoc). A boundary spanning coordinator role is also necessary.

Proposition 1 treats coordination effectiveness as a unitary concept. Propositions 2, 3, and 4 elaborate on the relationships between the three coordination strategy dimensions and the two coordination effectiveness dimensions. First, synchronisation activities such as iteration zero planning meetings, product demonstrations at the end of each sprint, daily standup meetings, and other meetings held at irregular intervals, serve to increase the project team's implicit coordination. In addition, artefacts produced during these activities such as story cards, task cards, and wallboard displays increase the project team's implicit coordination. That is these meetings and artefacts increase the project team members knowledge of the reasons why they are working on a task, increases their knowledge about what is going on and when, their knowledge about what to do and when, and their knowledge about who is doing what on the project. These meetings also increase their knowledge about who knows what within the project team. This leads to proposition two.

Proposition 2. Synchronisation activities at all frequencies – per project, per iteration, daily, and ad hoc, along with their associated synchronisation artefacts, increase implicit coordination effectiveness.

Similarly, the structural coordination mechanisms of close proximity, high availability, and high substitutability also increase each sub-dimension of implicit coordination. This leads to proposition three.

Proposition 3. Structural coordination mechanisms including close proximity, high availability, and high substitutability increase implicit coordination effectiveness.

Boundary spanning increases explicit coordination effectiveness, which is when the right things are in the right place at the right time. Boundary-spanning activities include activities such as holding a meeting with the customer or their representative. Boundary spanning artefacts include documents such as plans, requirements documents, specifications, and formal requests for resources. In an agile project, these activities occur once per project usually at project initiation, once per iteration such as when a product demonstration is held with invited customers, and ad hoc which means as and when needed. The leads to proposition 4.

Proposition 4. Boundary spanning coordination mechanisms including boundary-spanning activities at all frequencies, i.e. per project, per iteration, and ad hoc, their associated boundary-spanning artefacts, and a coordinator role, increases explicit coordination effectiveness.

RESEARCH DESIGN

To test the theory of coordination proposed by Strode et al. (2012) in the field using quantitative methods, each complex proposition stated in the theory (proposition 1 to 4) was decomposed into a series of simple testable hypotheses. Table 2 shows each proposition and its related hypotheses. The model for this system of hypotheses is shown in Figure 2.

	Propositions	Hypotheses
1	When the customer is embedded within the project team, a coordination strategy that includes synchronisation and structure coordination improves project coordination effectiveness. Synchronisation activities and associated artefacts are required at all frequencies (i.e. per project, per iteration, daily, and ad hoc). When the customer is an external party to the project then boundary spanning coordination is also needed. Boundary spanning activities and associated artefacts are required at all frequencies (i.e. per project, per iteration, and ad hoc). A boundary spanning coordinator role is also necessary.	H1. Customer involvement is positively related to implicit coordination. H2. Customer involvement is positively related to explicit coordination. H3. Customer involvement is negatively related to boundary spanning activities. H4. Customer involvement is negatively related to boundary spanning artefacts. H5. Customer involvement is negatively related to the boundary spanning coordinator role.
2	Synchronisation activities at all frequencies – per project, per iteration, daily, and ad hoc, along with their associated synchronisation artefacts, increase implicit coordination effectiveness.	H6. The frequency of synchronisation activities is positively related to implicit coordination. H7. The frequency of production of synchronisation artefacts is positively related to implicit coordination.
3	Structural coordination mechanisms including close proximity, high availability, and high substitutability increase implicit coordination effectiveness.	H8. Proximity is positively related to implicit coordination. H9. Availability is positively related to implicit coordination. H10. Substitutability is positively related to implicit coordination.

4	Boundary spanning coordination mechanisms including boundary-spanning activities at all frequencies, i.e. per project, per iteration, and ad hoc, their associated boundary-spanning artefacts, and a coordinator role, increases explicit coordination effectiveness.	H11. Boundary spanning activities are positively related to explicit coordination H12. Boundary spanning artefacts are positively related to explicit coordination H13. The boundary spanning coordinator role is positively related to explicit coordination.

Table 2 Mapping of propositions to hypotheses

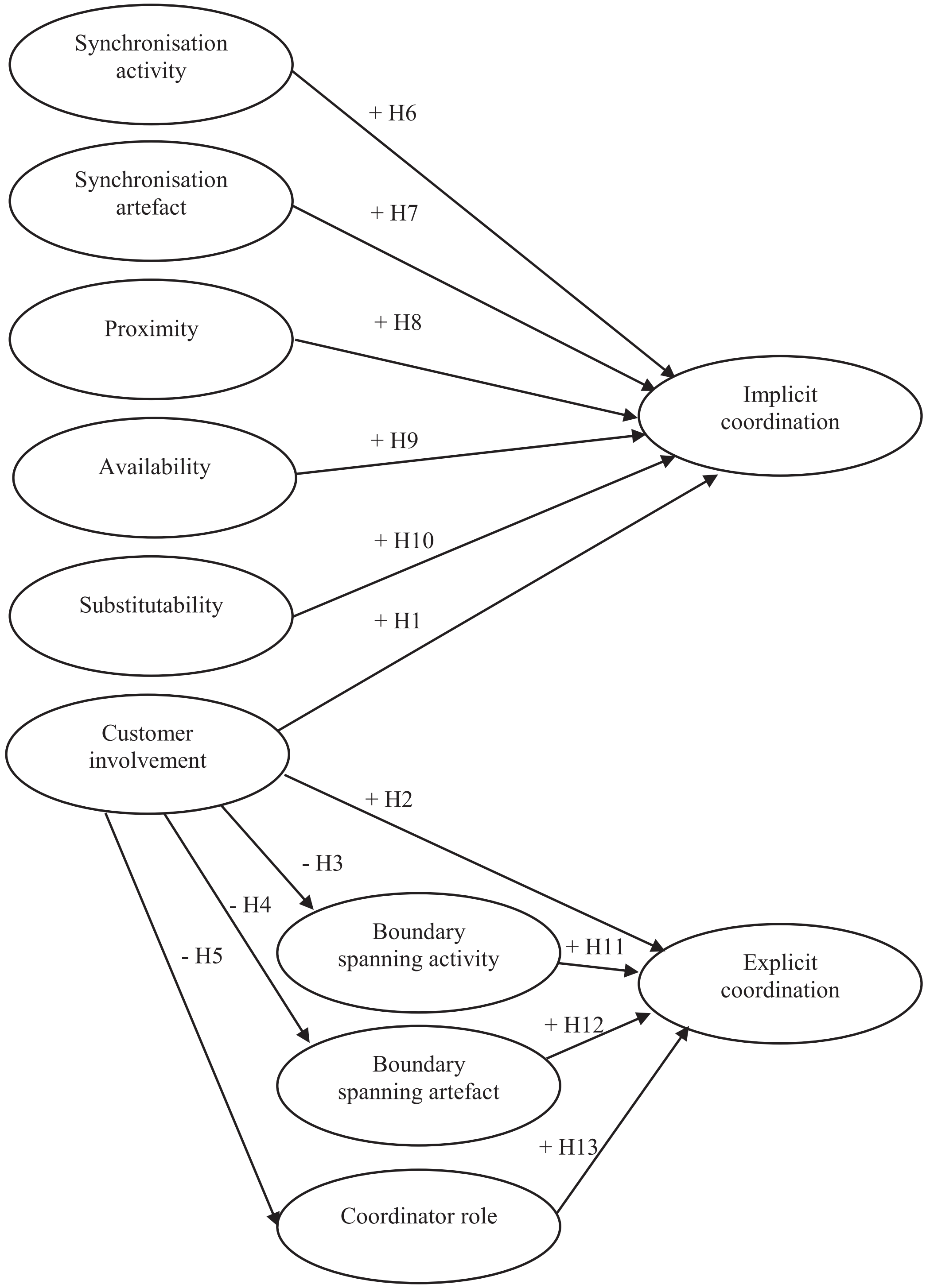

Figure 2 Theoretical model

This proposed field research has a unit of analysis at the project level. Therefore, field data will be drawn from projects selected to fit the project profile, as listed and described in the previous section. That is, each project will fit the theoretical boundaries and restrictions defined in Strode et al.'s (2012) theory. This will allow for some control over variation in team size and in the assemblage of practices used. This also means that sample selection is purposive rather than random.

Projects will be selected and data collected from each one using three survey instruments. The first instrument will be for project profiling and will be completed by a single knowledgeable person on the project. This instrument captures data such as estimated project duration, length of sprint, and other information likely to remain unchanged for the project duration. The second instrument is for collecting coordination strategy data, and the third instrument is for collecting coordination effectiveness data. Coordination strategy and coordination effectiveness are measured using 7-point scales, coordination strategy items range from "Not followed at all" to "Followed very strictly", and coordination effectiveness items range from "Strongly agree" to "Strongly disagree".

Each project will be visited weekly for up to 10 visits on the same day of the week at an agreed time. At the first visit, the team will be divided into two groups of the same size (or nearly equal for groups of uneven size). One group will complete the second questionnaire on the coordination strategy they used in the previous weeks work. The other group will complete the third questionnaire on the coordination effectiveness of the previous weeks work. In following visits, each group will be offered the alternative questionnaire. Assuming a team size of 10, there will be a maximum of 50 individual data points collected for one project.

The reason for splitting the team into groups with one group completing the strategy instrument and the other completing the effectiveness instrument is to reduce potential common methods variance. We believe this bias would be particularly likely in this context where the respondents would be fully aware of the expected outcomes of the practices they use in their projects. The most problematic source of common methods variance occurs when "the data for both the predictor and criterion variable are obtained from the same person in the same measurement context using the same item context and similar item characteristics" (Chang, van Witteloostuijn, and Eden, 2010, p. 179). To reduce this bias the simplest tactic for the project-level research proposed in this paper is to have different project team members assess different variables in each timeframe. Data analysis will involve aggregating data from a single project. Exploratory factor analysis followed by confirmatory factory analysis is proposed. The theoretical model contains formative constructs therefore PLS-SEM, which is appropriate for this type of model, is planned for data analysis using the SmartPLS™ tool.

STATUS OF THE RESEARCH

The research project is at the stage where preliminary survey instruments are developed (see Appendix for questionnaire items for coordination strategy and effectiveness). The project profile questionnaire has been used in prior projects to assess face validity. Face validity for the coordination strategy and coordination effectiveness instruments has been assessed by 10 people including agile software development professionals, IT students, and non-IT people, and the instruments have been adjusted accordingly. A further assessment of content validity is planned using an item-rating task with agile software development professionals and project managers (MacKenzie, Podsakoff, and Podsakoff, 2011). Negotiations with project teams are underway to collect data for a pretest to assess psychometric scale properties, convergent, discriminant, and nomological validities (Straub, Boudreau, and Gefen, 2004).

POTENTIAL CONTRIBUTIONS

Potentially, this research will provide a better understanding of which agile practices contribute to effective project coordination. This information will be useful to practitioners because currently they select their agile method and adopt individual agile practices based on hard-won experience, the advice of consultants, or they adopt practices without prior understanding of their individual and combined effects.

This research has limitations. The first is that sampling is purposive rather than random. A second limitation is the sample size imposed by surveying only typical agile project teams, which are small, optimally ranging from 4 to 10 developers. These restrictions mean that the range of statistical tests available to analyse the data is limited.

CONCLUSION

This paper describes a proposed field test of a theory developed from case study research. The initial theory focuses on the relationship between coordination strategy, which are the behaviors performed by agile project teams in a co-located projects, and coordination effectiveness. The field test proposes to test four propositions that have been elaborated into hypotheses.

APPENDIX

Coordination Strategy Construct		
Dimension	**Sub-dimension**	**Items**
Synchronisation	Synchronisation activity	**This week we used…** Iteration or sprint planning meeting Story or task breakdown session Team members together selected stories to work on Reflective workshop or retrospective meeting Software release Informal unscheduled whole team meeting Story point prioritising session Product demonstration with whole team and customer Unscheduled meeting of 2 or more people Product backlog maintenance session Daily standup or daily team meeting Progress tracking with user stories Daily build of the complete system Self-assignment of stories Pair programming Cross-team talk Continuous integration and testing Broadcast email Acceptance testing Informal chat using SMS or similar technology
	Synchronisation artefact	**This week we used …** Coding standards A product backlog A continuously updated design document) A working version of the software User stories A burn-down chart A done list A whiteboard sketching or recording ideas A wiki for storing information Avatars on task or work item A software tool to store the backlog of stories Layered architecture (n or 3-tier) A source code control tool A unit test suite Ground rules Work items Work in progress limits A bug tracking tool (e.g. JIRA™) A Scrum wallboard – real or virtual A Kanban wallboard – real or virtual
Structure	Proximity	Customer was co-located with the team Team worked in open work area and could see one another
	Availability	Team had a single assignment and worked full-time
	Substitutability	Team members performed other's tasks (e.g. developer did some testing, BA did some development)
Boundary spanning	Boundary spanning activity	**This week the team and an external party (client, customer, end-user, or group of end-users)…** Shared domain knowledge Worked together to generate a backlog Prioritized user stories together Viewed a software demonstration together Consulted daily together Had an unscheduled formal meeting together (face-to-face or by distance) Had an unscheduled informal face-to-face meeting together
	Boundary spanning artefact	**This week the team …** Prepared documents for external parties
	Coordinator role	**This week the team …** Had a team member act as customer liaison

Table 3 Construct definition: Coordination Strategy

Coordination Effectiveness Construct		
Dimension	**Sub-dimension**	**Items** **In the previous week …**
Implicit coordination	Know why/shared goal	**To what extent do you feel you understood:** The overall project goal How tasks you worked on helped to achieve the project goal
	Know what is going on and when	**To what extent do you feel you understood:** What tasks were underway? Which tasks needed to be performed?
	Know what to do and when	**To what extent do you feel that you knew:** What tasks you should be working on? When the tasks you worked on were required to be finished?
	Know who is doing what	**To what extent do you feel that you knew:** What tasks your project team members were working on?
	Know who knows what	**To what extent do you feel that you knew:** What knowledge your project team members had? What capabilities your project team members had?
Explicit coordination	Right thing Right place Right time	**To what extent do you feel that people, resources, and information you depended on to complete your work:** Were available at the right time Were available at the correct location Were fit for use
Additional question		Do you think the project work was well coordinated this week?

Table 4 Construct definition: Coordination Effectiveness

REFERENCES

Allen, R. E. (1990). The concise Oxford dictionary of current English (8 ed.), Clarendon Press, Oxford.

Arrow, H., McGrath, J. E., and Berdahl, J. L. (2000). Small groups as complex systems: Formation, coordination, development, and adaptation, Sage, Thousand Oaks.

Beck, K. (2000). Extreme Programming Explained: Embrace Change, Addison-Wesley, Boston.

Chang, S., van Witteloostuijn, A., and Eden, L. (2010). From the editors: Common method variance in international business research. *Journal of International Business Studies,* 41, 1, 178-184.

Chuang, S.-W., Luor, T., and Lu, H.-P. (2014). Assessment of institutions, scholars, and contributions on agile software development (2001-2012). *The Journal of Systems and Software*. doi: http://dx.doi.org/10.1016/j.jss.2014.03.006.

Conboy, K., and Fitzgerald, B. (2007). The views of experts on the current state of agile method tailoring, in T. McMaster, D. Wastell, E. Ferneley and J. DeGross (Eds.), *IFIP International Federation for Information Processing* (Vol. 235, pp. 217-234). Springer, Boston.

Crowston, K., and Osborn, C. S. (2003). A coordination theory approach to process description and redesign, in T. W. Malone, K. Crowston and G. A. Herman (Eds.), *Organizing Business Knowledge: The MIT Process Handbook* (pp. 335-370), The MIT Press, Cambridge, Massachusetts.

Cummings, J. N., Espinosa, A. J., and Pickering, C. K. (2009). Crossing spatial and temporal boundaries in globally distributed projects: A relational model of coordination delay. *Information Systems Research,* 20, 3, 420-439.

Curtis, B., Krasner, H., and Iscoe, N. (1988). A field study of the software design process for large systems, *Communications of the ACM,* 31, 11, 1268-1287.

Dingsoyr, T., Nerur, S., Balijepally, V., and Moe, N. (2012). A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software,* 85, 6, 1213-1221.

Espinosa, A. J., Lerch, F. J., and Kraut, R. E. (2004). Explicit versus implicit coordination mechanisms and task dependencies: One size does not fit all. In E. Salas and S. M. Fiore (Eds.), *Team Cognition: Understanding the Factors that Drive Process and Performance* (pp. 107-129), American Psychological Association, Washington DC.

Faraj, S., and Sproull, L. (2000). Coordinating expertise in software development teams, *Management Science,* 46, 12, 1554-1568.

Hoda, R., Noble, J., and Marshall, S. (2010). Organizing self-organizing teams, *Proceedings of the IEEE/ACM International Conference on Software Engineering, ICSE 2010, May 2-9, 2010* (pp. 285-294), ACM, New York, NY.

Hoegl, M., and Proserpio, L. (2004). Team member proximity and teamwork in innovative projects, *Research Policy*, 33, 8, 1153-1165.

Jalali, S., and Wohlin, C. (2012). Global software engineering and agile practices: A systematic review, *Journal of Software: Evolution and Process,* 24, 643-659. doi: 10.1002/smr.561.

Kotlarsky, J., van Fenema, P. C., and Willcocks, L. P. (2008). Developing a knowledge-based perspective on coordination: The case of global software projects, *Information and Management,* 45, 2, 96-108.

Kraut, R. E., and Streeter, L. A. (1995). Coordination in software development, *Communications of the ACM,* 38, 3, 69-81.

Levina, N., and Vaast, E. (2005). The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems, *MIS Quarterly*, 29, 5, 335-363.

Lin, T.-C., Hsu, J. S.-C., Cheng, K., and Wu, S. (2012). Understanding the role of behavioural integration in ISD teams: An extension of transactive memory systems concept, *Information Systems Journal,* 22, 211-234.

Lohan, G., Conboy, K., and Lang, M. (2011). Examining customer focus in IT project management: Findings from Irish and Norwegian case studies, *Scandinavian Journal of Information Systems,* 23, 2, 29-58.

MacKenzie, S. B., Podsakoff, P. M., and Podsakoff, N. P. (2011). Construct measurement and validation procedures in MIS and behavioral research: Integrating new and exisiting techniques, *MIS Quarterly,* 32, 2, 293-334.

Malone, T. W., and Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys,* 26, 1, 87-119.

Matook, S., and Kautz, K. (2008). Mindfulness and agile software development. In S. L. Huff (Ed.), *Proceedings of the 19th Australasian Conference on Information Systems, ACIS 2008* (pp. 638-646). Retrieved from http://aisel.aisnet.org/acis2008/58.

Mishra, D., Mishra, A., and Ostrovska, S. (2012). Impact of physical ambiance on communication, collaboration and coordination in agile software development: an empirical evaluation, *Information and Software Technology,* 54, 10, 1067-1078.

Moe, N., Dingsoyr, T., and Dyba, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project, *Information and Software Technology,* 52, 5, 480-491.

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, 191-219.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science, 5*(1), 14-37.

Parolia, N., Goodman, S., Li, Y., and Jiang, J. J. (2007). Mediators between coordination and IS project performance, *Information and Management,* 44, 7, 635-645.

Pries-Heje, L., and Pries-Heje, J. (2011). Why Scrum works. *Proceedings of the Agile Conference* 2011, Salt Lake City, UT (pp. 20-28): IEEE Xplore digital library. doi: 10.1109/AGILE.2011.34

Ren, Y., Kiesler, S., and Fussell, S. (2008). Multiple group coordination in complex and dynamic task environments: Interruptions, coping mechanisms, and technology recommendations, *Journal of Management Information Systems,* 25,1, 105-130.

Rico, R., Sanchez-Manzanares, M., Gil, F., and Gibson, C. (2008). Team implicit coordination processes: A team knowledge-based approach, *Academy of Management Review,* 33, 1, 163-184.

Salas, E., Sims, D. E., and Burke, C. S. (2005). Is there a 'big five" in teamwork? *Small Group Research,* 36, 5, 555-599.

Sarker, S., Munson, C. L., Sarker, S., and Chakraborty, S. (2009). Assessing the relative contribution of the facets of agility to distributed systems development success: an analytic hierarchy process approach, *European Journal of Information Systems,* 18, 1, 285-299.

Schmidt, K., and Simone, C. (1996). Coordination mechanisms: towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 5, 155-200.

Schwaber, K., and Beedle, M. (2002). *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River, New Jersey.

Sharp, H., and Robinson, H. (2008). Collaboration and co-ordination in mature eXtreme programming teams. *International Journal of Human-Computer Studies,* 66, 7, 506-518.

Sharp, H., and Robinson, H. (2010). Three 'C's of agile practice: collaboration, co-ordination and communication. In T. Dingsoyr, T. Dyba and N. B. Moe (Eds.), *Agile software development: current research and future directions*, Springer-Verlag, Heidelberg.

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage, *The Journal of Systems and Software, In press*. doi: http://dx.doi.org/10.1016/j.jss.2014.03.041.

Straub, D., Boudreau, M.-C., and Gefen, D. (2004). Validation guidelines for IS postivist research. *Communications of the Association for Information Systems,* 13, Article 24, 380-427. http://aisel.aisnet.org/cais/vol13/iss1/24.

Strode, D. E. (2012). *A theory of coordination in agile software development projects* (PhD), Victoria University of Wellington, New Zealand. Retrieved from http://hdl.handle.net/10063/2505.

Strode, D. E., Huff, S. L., Hope, B., and Link, S. (2012). Coordination in co-located agile software development projects. *The Journal of Systems and Software*, 85, 6, 1222-1238. doi: 10.1016/j.jss.2012.02.017.

Teasley, S. D., Covi, L. A., Krishnan, M. S., and Olson, J. S. (2002). Rapid software development through team collocation, *IEEE Transactions on Software Engineering,* 28, 7, 671-683.

Weber, R. (2012). Evaluating and developing theories in the information systems discipline, *Journal of the Association for Information Systems,* 13, 1, 1-30.

Weick, K. E., and Roberts, K. H. (1993). Collective mind in organizations: Heedful interrelating on flight decks. *Administrative Science Quarterly,* 38, 3, 357-381.

Williams, L. (2010). Agile software development methodologies and practices. In M. V. Zelkowitz (Ed.), *Advances in Computers* (Vol. 80, pp. 1-44), Elsevier, Amsterdam.

Yang, D.-H., Kang, H.-R., and Mason, R. M. (2008). An exploratory study on the meta skills in software development teams: Antecedent cooperation skills and personality for shared mental models. *European Journal of Information Systems,* 17, 47-61.