12-13-2014

Software Development in Multiteam Systems: A Longitudinal Study on the Effects of Structural Incongruences on Coordination Effectiveness

Saskia Bick
University of Mannheim, Mannheim, Germany, bick@uni-mannheim.de

Alexander Scheerer
University Mannheim, Mannheim, Germany., scheerer@uni-mannheim.de

Kai Spohrer
University of Mannheim, spohrer@uni-mannheim.de

Thomas Kude
University of Mannheim, kude@uni-mannheim.de

Armin Heinzl
University of Mannheim, heinzl@uni-mannheim.de

Software Development in Multiteam Systems: A Longitudinal Study on the Effects of Structural Incongruences on Coordination Effectiveness

Saskia Bick
University of Mannheim
bick@uni-mannheim.de

Alexander Scheerer
University of Mannheim
scheerer@uni-mannheim.de

Kai Spohrer
University of Mannheim
spohrer@uni-mannheim.de

Thomas Kude
University of Mannheim
kude@uni-mannheim.de

Armin Heinzl
University of Mannheim
heinzl@uni-mannheim.de

ABSTRACT

This study examines structural incongruences between organizational and product domains and their implications for coordination effectiveness in large-scale software development. We use the ongoing shift from on-premise to cloud-based software solutions to examine longitudinal effects of structural incongruences, i.e. the mismatch between organizational structures, including knowledge and task dependencies, and product structures, including technical dependencies, and how they resolve. We integrate extant literature in this field with literature on multiteam systems (MTS) and team composition to guide our longitudinal case study of one particular MTS within a large software organization. First insights from an initial case study of three development teams of different MTSs show how high level structural incongruences emerge on a team-level, providing a foundation for our subsequent study. By exploring the effects of structural incongruences over time, we expect to contribute to existing literature on organizational and product structure alignment as well as on MTS coordination effectiveness research.

Keywords

Organizational structure, product structure, structural incongruence, multiteam systems, dependencies, coordination effectiveness, longitudinal case study, process theory

INTRODUCTION

In recent years, the ever increasing focus on faster delivery cycles and quicker reaction to customer feedback led to profound changes for software development organizations, such as the widespread change in the provisioning mode of software solutions (Shetty, 2013; Stuckenberg, Kude and Heinzl, 2014). Many large software development organizations changed their license model by shifting from developing and selling on-premise software solutions, i.e. software which is hosted and run by the customer, to offering on-demand software solutions, i.e. software which is hosted and run by the vendor. Such a fundamental strategic change implies new requirements and challenges for the entire organization, which needs to realize a fast delivery of new features for its on-demand solutions (Dillon, Wu and Chang, 2010).

The modern large-scale software development organization tries to cope with these changing requirements by establishing a particular, team-oriented organizational structure. In most large-scale IT projects, several teams work together to implement a software system and a team of teams, or multiteam, setup has become widely accepted practice (Scheerer, Hildenbrand and Kude, 2014). Typically, the multiteam system (MTS) structure has been based on the modularized components of the software architecture (Keith, Demirkan and Goul, 2013; Subramanyam, Ramasubbu and Krishnan, 2012). In such a setting, each team owns the development and maintenance responsibility for one component. Modularized by means of the software architecture stack, i.e. the technological layers of a software product, this could mean that team A owns the responsibility for the user interface layer, whereas team B is responsible for the database layer of the software. Urged by customers to provide more frequent functional updates, companies are currently focusing on alternative ways of structuring their development organization. Instead of keeping a long-lasting and specialized software component responsibility, development teams repeatedly take on responsibility for small parts of functionality, i.e. features, which deliver visible business value across various components (Larman and Vodde, 2010). In complex enterprise software, however, single features necessarily span multiple modular components and different layers of the existing technology stack (see Figure 1).

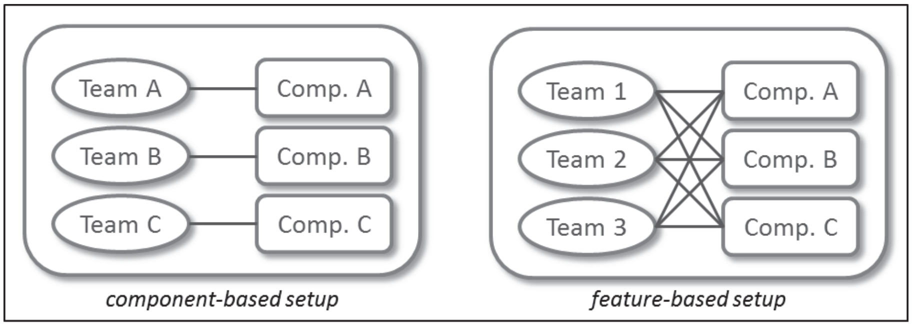

Figure 2: Component Responsibility

As a consequence, responsibilities and dependencies, both within and between teams, are undergoing major shifts. Due to the lack of a clear component ownership (represented through continuous lines in the figure above), organizational ties between teams are subject to change, as each specific team is now working on features across different components and is always peripherally interfering with the work of neighboring teams. Furthermore, dependencies both between and within teams are changing as the distribution of specialized knowledge and expertise is altered (Larman and Vodde, 2010). Henceforth, the coordination between teams in an MTS needs to be managed more coherently as to avoid conflicts due to misalignment of work. While this new form of organizational structure is expected to increase delivery frequency, flexibility, and customer-centricity, the above mentioned changes in responsibilities and dependencies, and more importantly the resulting effects on the organizational and product structure, are still widely unknown. This issue is of particular interest when taking into consideration the fact that the product structure is characterized by existing legacy code (Coplien and Harrison, 2004). Yet, to successfully deliver a product in scope and in time, these dependencies require all actors in the development organization to coordinate and adapt (Strode, Hope, Huff and Link, 2011).

Previous research in organizational design suggests that organizational structures, such as the division of labor, e.g. task assignments or organizational and communication links, influence the structure of the developed product, i.e. the technical architecture and its dependencies between design components (Colfer and Baldwin, 2010; MacCormack, Baldwin and Rusnak, 2008; Sosa, Eppinger and Rowles, 2004). The reported congruence between these structures does not imply any specific directionality, and evidence exists for both directions of influence as well as for a reciprocal influence. Furthermore, several studies have shown that this mirroring relationship between organizational and product structure is desirable, however, typically difficult to achieve and maintain (cf. Colfer and Baldwin, 2010). So far, this research domain has mostly perceived organizational structures as links between individual developers, i.e. looking at the phenomenon from the perspective of an individual as unit of analysis.

The increasingly important MTS structure in large-scale software development organizations has this far been disregarded in the organizational and product alignment literature. Yet, literature on MTS and MTS coordination describes highly complex communication and coordination processes between multiple teams, suggesting a very different structural complexity than in a single team setting (Scheerer et al., 2014; Strode et al., 2011). The new ways of structuring multiteam systems described above appear to disrupt the mapping of organizational and product structure. Lacking a clearly defined responsibility of teams within an MTS for specific components, the development organization is now characterized by a structure of shared responsibilities among multiple actors. Changing the original task assignments and communication links which existed between specialized experts for individual components, and instead sharing responsibilities for the latter may affect the coordination processes within the MTS, and also weaken a sound product modularity. A strongly modularized product structure, however, has been shown to be the source for development and coordination time reduction (Gomes and Joglekar, 2008; Yoo, Henfridsson and Lyytinen, 2010).

Beyond the limited view in the organizational structure domain focusing only on links between individuals, research in this area has so far been conducted independent of time. Nonetheless, the alignment between organizational and product structure is to be understood as a continuous evolutionary process of change, which motivates researchers to call for longitudinal studies assessing the effects of this alignment process (Colfer and Baldwin, 2010; Sosa et al., 2004).

In sum, there appear to be important but theoretically unresolved effects between the described structures. We view the concept of structural (in)congruence as the (mis)match between organizational structures, including knowledge and task dependencies, and product structures, including technical dependencies. Embedded in the overarching research question of *how organizational and product structure realign after a disruption of congruence*, we want to find out *how and why the coordination effectiveness within multiteam systems in large-scale agile software development organizations is affected by the structural incongruence between the organizational and product domain*. To do so, we integrate literature on organizational structure, particularly on MTS and team composition, and literature on product structure. Furthermore, pursuing a process-theoretic approach, one objective is to collect rich and insightful data to propose a process model representing the structural alignment between the organization, respectively an MTS, and the according software product under development. We aim to extend the understanding of time-variant dependencies within the organizational and product domain in large-scale development and the resulting coordination effectiveness by investigating it as a multilevel phenomenon in a real-life case study setting. We are engaged in a research project with a large international software vendor which provides us with access to extensive data from multiteam systems, not only in terms of their organizational but also of their product structure. Embedding our study in this industry context adds to the practical relevance and further offers promising new theoretical insights.

THEORETICAL FOUNDATION

Structural Alignment

The mirroring hypothesis, or Conway's Law (Conway, 1968) as it is known in computer science, predicts an alignment between organizational and (software) product structure. According to Colfer and Baldwin (2010), it describes the congruence between the technical architecture, the division of labor and the division of knowledge in a complex system. The technical architecture depicts the decomposition into specific software components and the *technical dependencies* between those (Baldwin and Clark, 2000). The division of labor consists of the task allocation to people or teams as well as the organizational ties, e.g. communication links or work flows between actors. This division of labor results in and can be illustrated as *task dependencies*. The division of knowledge is represented by the allocation of information to teams and how it is shared between them, i.e. *knowledge dependencies* (Strode and Huff, 2012). As mentioned earlier, previous research postulates that this alignment, or mirroring relationship between the organizational and the product structure, which eventually leads to a state of structural congruence, is desirable, however, typically difficult to achieve and maintain in heterogeneous environments (cf. Colfer and Baldwin 2010).

Organizational Structure

The previously mentioned organizational ties are defined by the way in which the software development system is set up. The large-scale context shows particular challenges as large groups of people need to be coordinated, which usually results in a hierarchical team of teams setup (Larman and Vodde, 2008) where several teams have to work closely together in order to release a single software product.

Multiteam Systems

The previously described organizational setup has been defined as a multiteam system (MTS) by Mathieu et al. (2001, p. 290) who state that MTSs are "two or more teams that interface directly and interdependently in response to environmental contingencies toward the accomplishment of collective goals". The collective goal of such systems can be broken down into a goal hierarchy and constitutes a key characteristic of any MTS. The goal hierarchy marks the boundary of an MTS in that all teams within the system share at least a distal goal while the individual teams pursue their more proximal goals. This structure of goals leads to teams displaying input, process and outcome interdependence with at least one other team (Mathieu et al., 2001). Previous studies have viewed the teams in MTS research as a black box. Delving into knowledge dependencies between teams, however, requires looking into those specific teams more closely.

Team Composition

Just like a larger MTS, a single team can be organized in different ways. Being seen as one predictor of the important outcome variable team performance (Levine and Moreland, 1990), team composition is a heavily investigated research area – quite contrary to the rather new field of MTS research. As opposed to surface-level differences such as demographic attributes, deep-level differences, e.g. team cognition, skill and knowledge distribution, promote an increasing heterogeneity among a team (Harrison, Price and Bell, 1998; Mannix and Neale, 2005). In line with this, Bunderson and Sutcliffe (2002) found dominant function diversity, i.e., the extent to which team members differ in individual functional areas of expertise, intra- and interpersonal functional diversity, to influence communication as well as coordination. That is, team communication and coordination are strongly influenced by individuals being narrow functional specialists or broad functional generalists.

Extant research on coordination, i.e. the management of dependencies between individuals or teams (Malone and Crowston, 1994; Mintzberg, 1979), as well as on coordination effectiveness (Strode et al., 2011) differs not only in the respective unit of analysis, but also in its level of maturity (Scheerer et al., 2014). Furthermore, the effects of dependencies, be it knowledge or task dependencies (Malone and Crowston, 1994; Strode and Huff, 2012; Thompson, 1967), may not be generalized from the team to the MTS level without reflection.

Product Structure

The product structure, in our case a software product, is depicted within the software architecture. Here, the software system is decomposed into components and linkages between them. Several architectural styles and patterns have been developed over the years (cf. Clements et al. 2010; Taylor et al. 2009). In general, one can differentiate between integral and modular architectures. Integral architectures exhibit considerable overlap between functional elements and physical components, and display a tight coupling between interfaces (Ulrich, 1995). This leads to the state that a change in one part of the product affects other parts (Yoo et al., 2010). A defining property of modular architectures are the standardized interfaces between components which allow a system to be decomposed into individual components that can be recombined (Schilling, 2000). The technical dependencies between these components signify the relationship in the form that a change in component 1 may lead to a change in component 2 (Baldwin and Clark, 2000).

Summary and Initial Framework

We integrate existing literature on organizational and product structure, as well as literature on their alignment with the aim to carve out the importance of task, knowledge and technical dependencies. Figure 3 illustrates these inherent dependencies within the organizational and product structure. By visualizing them, our initial framework provides the basis for a subsequent process model, which will be one contribution of our longitudinal study. Such a process model will include specific events, which we will extract from our collected case study data. These triggering events will then cause an alignment process to happen, which will result in an outcome of differing states of the previously explained structural congruence. This process outcome shall then be linked to the coordination effectiveness of the respective MTS.

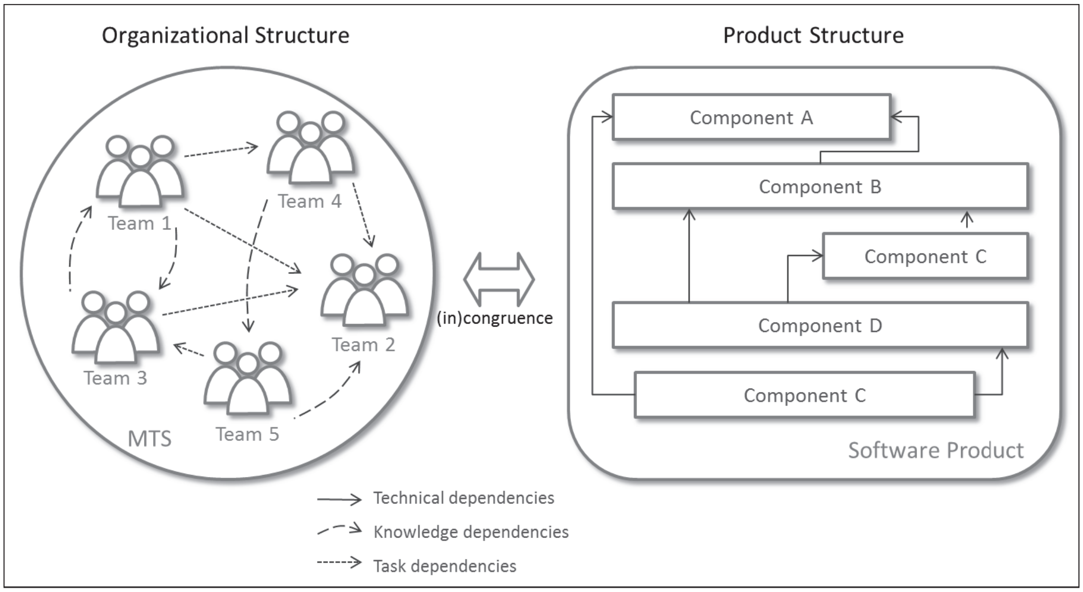

Figure 3: Organizational and Product Structure Dimensions

RESEARCH DESIGN

We follow a multi-stage approach conducting research at one large enterprise software vendor. The company is currently changing its business model from selling licenses for on-premise software installations to providing cloud-based solutions. Thus, several organizational units are changing their structure from a component-focus to a feature-focus. The organization has implemented a large-scale agile development approach to handle the increasing complexity and foster flexibility. This development approach builds upon agile project management frameworks, such as the Scrum approach (Larman and Vodde, 2008; Schwaber and Beedle, 2001), which is practiced throughout the company and suggests task management through prioritized backlogs.

We divided our data collection into two phases. First, two of the authors have been immersed in the company over a period of more than two years with several on-site visits to the company's headquarters per week. Based on their experience coming from unstructured interviews, informal conversations and a general exploration of the software development activities, we conducted three case studies with single development teams (Miles and Huberman, 1994; Yin, 2009). These teams were chosen according to the organizational structure of the particular MTS they were embedded in. Here, we purposefully selected three different MTSs with clearly differing organizational setups regarding their component-, respectively their feature-orientation, to analyze structural effects on the individual development team. The goal of these team-level studies was to gain initial insight into effects of incongruences at different levels of the organizational structure. The first team worked in a traditional component setup, being responsible for one component, team 2 integrated and orchestrated functionality delivered by other teams and team 3 developed end-to-end features across different components in the technology stack. We followed the three teams to retrospectively find out where incongruences led to inhibitions of their daily work as evidenced by so called disruptive events (Kude, Bick, Schmidt and Heinzl, 2014; Strong and Volkoff, 2010). Data collection took place during summer 2013 and included 25 interviews and eleven observations of team meetings. Semi-structured and structured interviews were conducted with each team's product owner and scrum master as well as six developers per team, each taking about 45 minutes. In addition, open interviews with twelve senior developers, team leaders and software architects from other teams helped us better understand the three teams' product contexts. Based on our data, we analyzed 57 events that we used to detect structural incongruences and resulting adaptive behavior which we conceptualized as the structural realignment across teams and products.

Based on these explorative pre-study results, we were able to select the particular MTS of one of the studied teams which promised most revelatory insights into a structural realignment process. MTS *ALPHA* consists of six development teams with more than 50 individuals and has very recently started to move from a component- to a feature-based organizational structure. We have the opportunity to closely accompany and analyze this MTS during the next three years to conduct a longitudinal case study. Such a longitudinal design is in line with prior multilevel research and helps capture not only events and effects that become apparent rapidly but also ones that emerge in the course of action and which are not immediately observable (Lapointe and Rivard, 2005; Strong and Volkoff, 2010). Taking a process-view, we aim to focus on the changes emerging from the division of labor and knowledge, as well as the changes in software architecture to better understand their interrelations and thus observe alignment. In more detail, we will be able to analyze data coming from various sources such as interviews, existing documentation, observation of teams in ALPHA, and historic and current data from the MTS's project management system. This diverse body of data will allow us to gain a multi-faceted picture of the observable events that accompany the process of realignment between organizational and product structure and will help us gain insight into effects on the coordination effectiveness of the teams within the respective MTS.

INITIAL INSIGHTS

Our preliminary findings from the pre-study provide interesting insights into the effects of structural incongruences on development teams. We found team 3 to face a much higher number of disruptive events that were also more severe than in the other teams. These disruptions also spanned a much wider variety of technological aspects. They ranged from breakdowns after an update of graphical user interface components to externally introduced bugs in some important database components. As a consequence, this team was much more often unable to resolve such issues internally than the two other observed teams. Thus, work came to a halt more often and team 3 had to fall back to time-consuming escalation mechanisms on the MTS-level. Before analyzing these events in more detail, we needed to closely examine the working context of team 3. From interviews with managers and the team members we learned that the MTS had been restructured about one year before. After following the typical component-oriented MTS structure for years, a management decision resulted in a restructuring of the organizational unit into feature-oriented teams. Apart from breaking up existing teams and with it the established division of labor, i.e. organizational ties and communication links, the division of knowledge within and between the teams was redistributed entirely. Furthermore, the historic component ownership came to a halt and was replaced by shared responsibilities across various components. Based on these insights, we came to understand that this induced structural incongruence between the existing legacy code and the restructured MTS resulted in responsibility and communication conflicts between the teams which further led to decreased coordination effectiveness within the whole MTS.

Team 3 was required to develop features end-to-end across the technology stack while the existing software architecture was traditionally organized along single components. As peer teams had to work on the same components without knowing much about each other's activities, they frequently produced conflicts in each other's work: they changed parts of components in accordance with their own feature but thereby disrupted functions that other teams relied on. What is more, the dependency on a high number of different components prevented the team members from developing deep and specialized expertise in single components but forced them to become superficially knowledgeable about various parts of the technology stack. As a result, it became impossible to gain sufficient expertise to resolve issues rooted in the depth of a single component internally.

By contrast, team 1 neither experienced as severe disruptive events caused by the product structure nor were its reactions nearly as strong. Although this may well be attributed to various other circumstances, the organizational context of team 1 – namely the MTS ALPHA – was of a much more stable character. From interviews with the MTS's management, however, we learned about the intention to restructure the MTS from a component- to a feature-oriented setting. Giving us access to their entire MTS provides us with the unique opportunity to conduct a longitudinal case study promising very rich data resulting from this structural realignment process.

EXPECTED CONTRIBUTION AND OUTLOOK

Adopting a multilevel perspective on organizational and product structure, our goal is to understand the resolution of structural incongruences and the effects on coordination effectiveness of multiteam systems. Our preliminary results show that there are highly relevant but theoretically disregarded effects caused by such structural incongruences. While these initial and exploratory results provide useful insights into the team-level effects, our next step will be to conduct a longitudinal case study with one particular MTS at our industry partner. This will allow us to examine the intermediate outcome variable coordination effectiveness on a multiteam-level and look at entailing effects on performance indicators such as the time to market. This enables us to better understand the effects of structural incongruences and potential realignment over time through adaptation processes.

Doing so, we aim to contribute our insights from a longitudinal case study to the body of knowledge on organizational and product structure alignment. Answering the call for longitudinal research on this topic, we expect to find out more about the effects of intentionally caused structural incongruences on MTS coordination effectiveness over the course of time. Accompanying one particular MTS in the setting of a large-scale software development organization during the change from one organizational setup to another, we will collect rich data on the shift of task, knowledge and technical dependencies. Whereas recent work in MTS research has so far conceptualized single teams as black boxes, we seek to investigate the role of team composition within an MTS in more detail by looking at the division of labor and the division of knowledge at the team level. Initial insights in this multilevel research study indicate that a redistribution of knowledge and a breakup of organizational ties within individual teams can have severe effects on the overall effectiveness of the entire MTS.

Further, we aim to develop a process model which depicts the alignment process of the organizational and the product structure based on the shift of dependencies between and within teams and their respective product domain. Here, the goal is to identify triggering events, such as for instance a *responsibility conflict*, and further carve out potential outcome patterns of this alignment process.

BIBLIOGRAPHY

Baldwin, C. Y. and Clark, K. B. (2000). Design Rules: The Power of Modularity (1st ed.). *The MIT Press*.

Bunderson, J. S. and Sutcliffe, K. M. (2002). Comparing Alternative Conceptualizations of Functional Diversity in Management Teams: Process and Performance Effects. *Academy of Management Journal*, 45, 5, 875-893.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., … Nord, R. L. (2010). *Documenting Software Architectures: Views and Beyond* (2nd ed.). Addison Wesley.

Colfer, L. and Baldwin, C. Y. (2010). The Mirroring Hypothesis: Theory, Evidence and Exceptions (No. 10-058). *Harvard Business School*, 1-39.

Conway, M. E. (1968). How do committees invent. *Datamation*, 14, 4, 28-31.

Coplien, J. O. and Harrison, N. B. (2004). *Organizational Patterns of Agile Software Development*. Prentice Hall.

Dillon, T., Wu, C. and Chang, E. (2010). Cloud Computing: Issues and Challenges. In *IEEE International Conference on Advanced Information Networking and Applications*, 27-33.

Gomes, P. J. and Joglekar, N. R. (2008). Linking Modularity with Problem Solving and Coordination Efforts. *Managerial and Decision Economics*, 29, 5, 443-457.

Harrison, D. A., Price, K. H. and Bell, M. P. (1998). Beyond Relational Demography: Time and the Effects of Surface- and Deep-Level Diversity on Work Group Cohesion. *Academy of Management Journal*, 41, 1, 96-107.

Keith, M., Demirkan, H. and Goul, M. (2013). Service-Oriented Methodology for Systems Development. *Journal of Management Information Systems*, 30, 1, 227-260.

Kude, T., Bick, S., Schmidt, C. and Heinzl, A. (2014). Adaptation Patterns in Agile Information Systems Development Teams. In *22nd European Conference on Information Systems (ECIS)*, Tel Aviv, 1-15.

Lapointe, L. and Rivard, S. (2005). A Multilevel Model of Resistance to Information Technology Implementation. *MIS Quarterly*, 29, 3, 461-491.

Larman, C. and Vodde, B. (2008). Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum (1st ed.). *Addison-Wesley Professional.*

Larman, C. and Vodde, B. (2010). Practices for Scaling Lean & Agile Development Large, Multisite, and Offshore Product Development with Large-Scale Scrum (1st ed.). Upper Saddle River, NJ: *Addison Wesley*.

Levine, J. M. and Moreland, R. L. (1990). Progress in Small Group Research. *Annual Review of Psychology*, 41, 585-634.

MacCormack, A., Baldwin, C. and Rusnak, J. (2008). Exploring the duality between product and organizational architectures: A test of the “mirroring” hypothesis. *Research Policy*, 41, 8, 1309-1324.

Malone, T. W. and Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26, 1, 87-119.

Mannix, E. and Neale, M. A. (2005). What Differences Make a Difference? The Promise and Reality of Diverse Teams in Organizations. *Psychological Science in the Public Interest*, 6, 2, 31-55.

Mathieu, J. E., Marks, M. A. and Zaccaro, S. J. (2001). Multiteam Systems. In N. Anderson, D. S. Ones, H. K. Sinangil and C. Viswesvaran (Eds.), *Handbook of Industrial, Work and Organizational Psychology*, Vol. 2, London: Routledge, 289-313.

Miles, M. B. and Huberman, A. M. (1994). Qualitative Data Analysis: An Expanded Sourcebook. Thousand Oaks: *Sage Publications.*

Mintzberg, H. (1979). The Structuring of Organizations. Englewood Cliffs. *Prentice-Hall.*

Scheerer, A., Hildenbrand, T. and Kude, T. (2014). Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective. In *47th Hawaii International Conference on System Sciences (HICSS),* Waikoloa, HI, 4780-4788.

Schilling, M. A. (2000). Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity. *Academy of Management Review*, 25, 2, 312-334.

Schwaber, K. and Beedle, M. (2001). Agile Software Development with Scrum. *Prentice Hall.*

Shetty, S. (2013). Gartner Says Cloud Computing Will Become the Bulk of New IT Spend by 2016. *Gartner Report.* Retrieved July 25, 2014, from http://www.gartner.com/newsroom/id/2613015

Sosa, M. E., Eppinger, S. D. and Rowles, C. M. (2004). The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science*, 50, 12, 1674-1689.

Strode, D. E., Hope, B., Huff, S. L. and Link, S. (2011). Coordination Effectiveness In An Agile Software Development Context. In *PACIS Proceedings*.

Strode, D. E. and Huff, S. L. (2012). A taxonomy of dependencies in agile software development. In *23rd Australasian Conference on Information Systems (ACIS), Geelong,* 1-10.

Strong, D. M. and Volkoff, O. (2010). Understanding Organization - Enterprise System Fit: A Path to Theorizing the Information Technology Artifact. *MIS Quarterly*, 34, 4, 731-756.

Stuckenberg, S., Kude, T. and Heinzl, A. (2014). Understanding the role of organizational integration in developing and operating Software-as-a-Service. *Journal of Business Economics*, February, 1–32.

Subramanyam, R., Ramasubbu, N. and Krishnan, M. S. (2012). In Search of Efficient Flexibility: Effects of Software Component Granularity on Development Effort, Defects, and Customization Effort. *Information Systems Research*, 23, 3, 787-803.

Taylor, R. N., Medvidovic, N. and Dashofy, E. (2009). Software Architecture: Foundations, Theory, and Practice (1st ed.). *John Wiley & Sons.*

Thompson, J. D. (1967). Organizations in Action: Social Science Bases of Administrative Theory. *McGraw-Hill.*

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24, 3, 419-440.

Yin, R. K. (2009). Case Study Research: Design and Methods (4th ed.). *SAGE Publications Inc.*

Yoo, Y., Henfridsson, O. and Lyytinen, K. (2010). Research Commentary - The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. *Information Systems Research*, 21, 4, 724-735.