Fall 9-11-2020

Scaling Agile: Approach for Defining Key Aspects of Multiteam Agile Software Delivery Systems (Research in Progress)

David Marshburn
Cleveland State University / SimVentions, d.marshburn@csuohio.edu

Jaap Dekkinga

Scaling Agile: Approach for Defining Key Aspects of Multiteam Agile Software Delivery Systems (Research in Progress)

David G. Marshburn
Cleveland State University / SimVentions
d.marshburn@csuohio.edu

Jaap V. Dekkinga
Excella
jaap.dekkinga@excella.com

ABSTRACT

The need to scale agile approaches for software delivery within larger organizations and larger projects has led to a proliferation of agile scaling frameworks. Anecdotal evidence of the resultant implementation of these scaling frameworks shows varying degrees of success. Missing from this discourse is a holistic, framework-independent understanding of scaling agility. This research proposes an approach for defining key aspects of agile scaling. Using a Delphi method, we will work with an international panel of agilists representing the major scaling frameworks to determine challenges for scaling agile. These results will then be compared to the existing agile scaling research to determine convergence and identify gaps within the existing research. We will also compare the results to the emerging research that uses multiteam systems to help explain the agile scaling phenomenon. These comparisons will provide a means to gauge the relevance of existing literature to practitioner identified needs. The results of the study will provide practitioners a framework-independent understanding of agile scaling for large organizations and projects and provide scholars a clear direction to support future research.

KEYWORDS

agile, scaling, multiteam systems, SAFe, Scrum @ Scale, Scrum of Scrums, Nexus, LeSS, DAD

INTRODUCTION

Many of today's software delivery organizations and software projects are large, requiring multiple teams working together to deliver a software product or products. The *2019 State of Agile Report* indicates that 73% of organizations have 100 people or more within their software organization (Collabnet VersionOne, 2019). In response to this demand, numerous frameworks have been developed to aid in scaling agile as the original agile frameworks were geared towards small teams and small projects (Dikert, Paasivaara, & Lassenius, 2016). Data indicates that, of the scaled organizations responding to the *2019 State of Agile Report*, 30% use SAFe, 16% use Scrum of Scrums, and 3% use LeSS (Collabnet VersionOne, 2019). A problem with these large-scale agile frameworks is that, in general, they focus on a specific agile framework and only provide guidance on how to use that agile framework with multiple teams, the agile scaling frameworks do not take a holistic look at large-scale agile to determine how large-scale agile is different from smaller agile framework implementations. Prior to selecting a scaling framework, practitioners need a framework-independent, holistic understanding of large-scale agile to better inform their decisions on implementing an agile scaling solution.

Agile research has begun to look at multiteam systems (MTS) to help understand large-scale agile (Bick, Scheerer, Spohrer, Kude, & Heinzl, 2014; Scheerer, Hildenbrand, & Kude, 2014; Scheerer & Kude, 2014). MTS are a way to categorize and study interdependent, cross-functional teams (Mathieu, Marks, & Zaccaro, 2001). Mathieu et al. note that "MTS are different from prior theories and models of system effectiveness (e.g., models found in organizational theory and strategic management literature), because previous conceptual frameworks have not attempted to explain the dynamics of how teams interact with each other to create an effective system" (2001, p. 289). Similar to agile, "MTSs are usually formed or develop naturally to deal with highly turbulent environments that place a premium on the ability to transform work units and to respond rapidly to changing circumstances" (Mathieu et al, 2001, p. 290). To date, however, MTSs have not been used to explain the overall phenomenon of large-scale agile or used to explain why large-scale agile is different than traditional organizational scaling.

The proposed research provides a framework independent lens to answer two basic scaling questions:

Research question 1: What are the core challenges to multiteam agile software delivery systems? As discussed, current large-scale agile research tends to focus on a particular framework. This study explores challenges independent of framework to identify challenges present in any multiteam agile software delivery system.

Research question 2: Does the multiteam systems research better align with the core challenges identified by practitioners than traditional scaling agile research? Finding support in the multiteam systems research for the practitioner defined challenges provides justification for its use in scaled agile research.

The empirical findings of this research will assist software delivery professionals who find themselves needing to scale. Instead of basing their scaling decisions on framework-centric scaling approaches, the research will provide a framework independent understanding of what it means to create a multiteam agile software delivery system. The core challenges provide practitioners with an objective lens through which to view and evaluate the framework-centric scaling approaches to help them better determine the optimal solution for their scaling needs.

The multiteam systems literature supports the research through a better understanding of the differences between agile teams and more standard organizational teams. The concept of multiteam agile software delivery systems highlights where traditional team and organizational research may fall short of explaining the issues and needs of such highly interdependent, cross-functional teams.

Finally, the research supports future studies into multiteam agile software delivery systems by providing a strong foundation, based on practitioner input and based on solid scientific methodologies. The results of the study will serve as the starting point for follow-on research to support the continued growth and acceptance of agile software delivery, specifically in the viewpoint of scaling.

The remainder of this paper is organized as follows. First, we briefly explore the agile scaling literature. Next, we provide an understanding of multiteam systems and their application within the agile scaling discussion. This is followed by our research methodology. The paper concludes with discussion and implications from the research. Due to page limitations and a concentrated focus on methodology, overviews of agile scaling frameworks and traditional project management are not included in this proposal.

BACKGROUND

Following in the footsteps of agile, scaled agile is becoming mainstream. Agile scaling has appeared in several leading practitioner outlets like The Harvard Business Review (Rigby, Sutherland, & Noble, 2018), MIT Sloan (Relihan, 2018), Forbes (Denning, 2016), and CIO (Heusser, 2016). As the original agile frameworks were developed and intended for small projects (Dikert et al., 2016) their suitability for large projects and large teams is in question (Begel & Nagappan, 2007). The agile principles emphasize, among other characteristics, daily interactions between team members and customers, face to face communication, and self-organizing teams (Beck, et al., 2001), principles which become harder to embody as project size and the number of teams working together increase (Ambler, 2009). Although practitioners are scaling agile (Denning, 2016; Heusser, 2016; Relihan, 2018) and transitioning from one scaling approach to another (Sutherland, 2019), research is lagging behind. In their literature review of large-scale agile transformations, Dikert et al. (2016) identified just 52 papers meeting their criteria, and only 6 of these papers contained defined research methodologies.

What exactly is "large-scale" agile?

In their workshop summary, Dingsøyr and Moe (2014) provide a number of participant-generated definitions of large-scale agile development that are based on factors such as number of teams, lines of code, and number of projects. Ambler (2009) identifies eight agility at scale factors: (i) team size, (ii) geographical distribution, (iii) regulatory compliance, (iv) domain complexity, (v) organizational distribution, (vi) technical complexity, (vii) organizational complexity, and (viii) enterprise discipline. Alternatively, Dingsøyr, Fægri, and Itkonen (2014) base their taxonomy of scale on the number of teams involved, ranging from 1 (small scale) to 10 or more (very large scale). Paasivaar, Behm, Lassenius, and Hallikainen (2018) classify the 15-team project in their case study as large-scale. Dikert et al. define large-scale agile as "software development organizations with 50 or more people or at least 6 teams" (2016, p. 88). Practitioners report much larger efforts, with Ambler identifying complex efforts as having "Thousands of Developers" (2009, p. 21). The lack of a clear definition of agile scaling is itself a problem, as Dikert et al. state, "what is seen as large-scale depends very much on the context and the person defining it" (2016, p. 88).

Multiteam Systems

Multiteam systems are defined as:

> *two or more teams that interface directly and interdependently in response to environmental contingencies toward the accomplishment of collective goals. MTS boundaries are defined by virtue of the fact that all teams within the system, while pursuing different proximal goals, share at least one common distal goal; and in doing so exhibit input, process, and outcome interdependence with at least one other team in the system.* - (Mathieu et al, 2001, p. 289)

Shuffler, Jimènez-Rodriguez, and Kramer (2015) inform us that research on MTS is relatively new, only dating back to 2001. Mathieu et al. explain that "Extant theory tells us little about the functioning of collectives such as these because they are neither typically full-fledged organizations nor single teams operating in isolation" (2001, p. 289). Unlike traditional teams, "these collectives comprise multiple teams that form a tightly coupled system, operating in an environment that demands coordinated *interteam* as well as *intrateam* behaviors in order to succeed" (Mathieu et al, 2001, p. 289). Initial MTS research looks at environments ranging from manufacturing to the military to emergency response (Mathieu et al., 2001) and has expanded to include finance, healthcare, and aerospace (Shuffler et al., 2015).

Scheerer et al. introduce MTS as a new perspective to view coordination in large-scale agile development, noting that "The type of large-scale development system examined is one in which several teams have to work together in order to complete a release of a software product" (Scheerer et al., 2014, p. 4781). The authors position their work as "a first step to establish the multiteam level of analysis in studies of software development organizations" (Scheerer et al., 2014, p. 4785) and further posit that "Based on the understanding gathered from these conceptual insights, we see a high value in pursuing an exploratory approach to shed light on coordination in large-scale software development" (Scheerer et al., 2014, p. 4786).

MTS research is focused on exploring aspects of the existing team literature which do not apply within an MTS environment (Shuffler et al., 2015). The MTS environment includes "goal hierarchies, functional interteam interdependencies, performance episodes, and how work is coordinated both within and across teams" (Mathieu et al., 2001, p. 290). Furthermore, "MTSs are usually formed or develop naturally to deal with highly turbulent environments that place a premium on the ability to transform work units and to respond rapidly to changing circumstances" (Mathieu et al., 2001, p. 290). Software development is fraught with change; responding to change is a key aspect of the agile principles (Beck, et al., 2001) and has been discussed in the research as software turbulence (Harris, Webb Collins, & Hevner, 2003; Buganza, Dell'Era, & Verganti, 2009).

RESEARCH DESIGN AND METHODOLOGY

While research has emerged on the majority of the eight agenda items identified by Dingsøyr and Moe's 2013 XP workshop (2013), (i) coordination (see (Bick et al., 2014; Dingsøyr, Moe, & Seim, 2018; Scheerer et al., 2014; Scheerer & Kude, 2014; Xu, 2009), (ii) project organization (see (Bass, 2015; Paasiaara, Lassenius, & Heikkilä, 2012; Paasivaara, Durasiewicz, & Lassenius, 2008)), (iii) release planning (see (Keikkilä, Paasivaara, Lassenius, & Engblom, 2013; Heikkilä, et al., 2015)), (iv) scaling (see (Dikert et al., 2016)), (v) customer collaboration, (vi) large-scale transformations (see (Dikert et al., 2016; Fry & Greene, 2007; Olszewska, Heidenberg, Mikkonen, & Porres, 2016)), (vii) knowledge sharing (see (Dingsøyr et al., 2018)), and (viii) contracts, there is not a definitive resource that looks holistically at scaled agile. Even the research on success factors (see (Dikert et al., 2016)) is limited as it relies on literature reviews and does not provide a systematic collection of practitioner expertise. With so many frameworks and limited understanding of the holistic issues, the choice of research methodologies is critical.

The chosen research methodology must consider the purveying sentiments of the differing scaling frameworks and the practitioners who employ them. If we consider these differing viewpoints as similar to differing cultures, then we can leverage past psychological research to inform our research methodology. Brislin (1976) warns us of the dangers of creating survey instruments from the perspective of one culture and using them to evaluate persons of another culture. The cross-cultural concern is that what is important in the first culture may be of no importance in the other and may, in fact, overlook more important concerns of the second culture (Brislin, 1976). A proposed solution to this dilemma is to create a set of core items, "items that are relevant to all cultures" (Brislin, 1976, p. 219). Considering Brislin's (1976) position, one could reasonably assert that Dingsøyr and Moe's (2014) items may not represent a valid starting point, depending on the representation each of the scaling frameworks had in the discourse. For example, if Dingsøyr and Moe's (2014) workshop only included proponents of Scrum @ Scale, then the items developed in the

workshop may be of no value to proponents of Nexus. This results in serious issues of generalizability which would dramatically hamper a framework-independent view of agile scaling.

Brislin (1976) suggests that researchers should work with others deeply familiar with the individual cultures in order to develop "general theoretical statements (etics) which summarize the data" (1976, p. 220). In our case, this requires gaining input from proponents of each of the scaling frameworks to distill into agile scaling etics. To achieve this, we will use a two-phase Delphi study.

Delphi Method

Schmidt, Lyytinen, Keil, and Cule use a Delphi study to develop "an authoritative list of common risk factors" (2001, p. 6) across multiple cultures to resolve the issue that "no validated lists are available to help the project manager understand the nature and types of risks typically faced in a software project" (Schmidt et al., 2001, p. 6). The authors use the Delphi study as it provides a process "designed to elicit and organize opinions of a panel of experts through iterative, controlled feedback" (Schmidt et al., 2001, p. 6).

The Delphi method can be used to create new information or to revise existing information. Schmidt et al. conducted their Delphi in three rounds beginning with a "brainstorming round" (2001, p. 11) in order to "elicit as many items as possible from the panel(s)" (2001, p. 11). Similarly, Brancheau, Janz, and Wetherbe (1996) conduct a Delphi study to determine key issues within the management information systems (MIS) fields during the 1990s and also use three rounds; however, the authors begin the first round with an existing data set identified in previous years. The processes used by both Schmidt et al. (2001) and Brancheau et al. (1996) are used to facilitate this research. First, as this research attempts to identify a non-framework centric, or etic, set of core items, existing items are not used as a starting point and so it is necessary to structure phase 1 of the Delphi study following Schmidt et al.'s (2001) example. However, as Schmidt et al. (2001) results in within-group rankings, phase 2 of the Delphi study follows the process identified by Brancheau et al. (1996) as the focus is develop the between-group rankings.

Delphi Study Phase 1

While the process identified by Schmidt et al. (2001) is used as a guide, several modifications are necessary. As the panelists are not expected to be co-located, the methodology must allow for more distributed data collection. To facilitate this, panelists work independently in each of the Delphi rounds for this study while Schmidt et al. (2001) allow for more of a collaborative effort.

Members of an international panel of agile practitioners representing the agile scaling frameworks identified in the *2019 State of Agile Report* (Collabnet VersionOne, 2019) are used in the first round of the Delphi study to create a master list of challenges that affect scaled agile implementations

In following the process described by Schmidt et al., all panelists "are asked to submit at least six factors, and to provide short descriptions of the factors to aid the researchers in their collation effort" (2001, p. 11). Panelist responses are collected digitally with each panelist working independently. Responses are then aggregated, first by two researchers independently, then by the researchers working together, to produce a single list of factors and definitions (Schmidt et al., 2001). The list is then distributed back to the individual panelist to gain their concurrence (Schmidt et al., 2001).

A second round of the Delphi study is conducted to identify items that are important to each group (Schmidt et al., 2001), to develop lists of emic items, items that "describe behavior in any one culture under study, taking into account what the people themselves value as meaningful and important" (Brislin, 1976, p. 217). In this case, to develop emic items that represent what is meaningful to proponents of each of the scaling frameworks.

Instead of creating panels that represent subject groups in accordance with Schmidt et al. (Schmidt et al., 2001), the panelists are provided the master list of factors and definitions and asked to identify those items that are important to them independently. The researchers then collate the factors using the panelists self-identified scaled agile framework preference. All factors that are common among at least half the group is kept (Schmidt et al., 2001), resulting in emic factors.

Using multiple iterations for the third round, panelists work to rank the factors based on priority (Schmidt et al., 2001). After each iteration, the factor ranking is analyzed using Kendall's coefficient of concordance (W) (Schmidt et al., 2001) which to measure "the degree of consensus among the panelists" (Schmidt et al., 2001, p. 13). Iterations continue until "either: (1) the coefficient of concordance indicate[s] a strong consensus (W>0.70), or (2) the level of consensus for the panel level[s] off in two successive rounds" (Schmidt et al., 2001, p. 13). During the first iteration, factors are

scrambled to help eliminate bias in ranking and are listed in rank order in all subsequent iterations (Schmidt et al., 2001). Additionally, during the first iteration panelists are asked to "rate the risk factors according to their relative importance" (Schmidt et al., 2001, p. 13) for successful agile scaling.

Delphi Study Phase 2

The results of the phase 1 Delphi process provide agile scaling factors ranked by framework panels (e.g., factors ranked by proponents of SAFe, factors ranked by proponents of LeSS). However, the goal of the current research is to provide framework-independent scaling factors, to achieve this requires a second phase in the Delphi study. Brancheau et al. (1996) provide a methodology for this in their research looking at key issues in the MIS field. Similar to Schmidt et al. (2001), the authors (Brancheau et al., 1996) also followed a three-round approach but leveraged the existing key issues identified in previous studies instead. By leveraging a new panel whose composition is comparable with the first panel, and by reusing a Delphi method helps to improve the quality of the study (Brancheau et al., 1996).

A new panel, similar in structure to the first panel is used for Phase 2. In accordance with Brancheau et al., "To reduce any bias inherent in a particular randomized sequence, four different randomized version of the surveys" (1996, p. 227) are distributed to the panelists. The surveys request each panelist to rank the most important factors to scaling agile based on the 10-point scale identified by Brancheau et al. "where 10 indicated their highest priority issues(s) and 1 indicated their lowest priority issue(s)" (1996, p. 227). The surveys also provide a write-in capability where panelist can add factors not already on the list (Brancheau et al., 1996).

For round two, summarized results are redistributed to the panel, including items added in response to the write-in capability of the first round (Brancheau et al., 1996). Brancheau et al. (1996) outline a process for determining the write-in responses to include based on clustering of similar items and a threshold number of write-ins; a similar process is followed in this study. Round 2 surveys provide the factors in round 1 rank order and panelists are again asked to rank the factors but are not provided the ability to write-in new items (Brancheau et al., 1996). While Brancheau et al. (1996) dropped items that were ranked significantly lower than previously ranked, the current study drops all items that are not ranked above a 5. Additionally, unlike Brancheau et al. (1996), respondents' previous responses are not provided to them for rounds 2 and 3.

In round three, factors are reordered based on the round 2 results (Brancheau et al., 1996) and again, factors are dropped that do not receive rankings above 5. The survey is provided to panel members who provide a final 10-point ranking of each item.

DISCUSSION AND IMPLICATIONS

The determination of a framework-independent, holistic understanding of large-scale agile challenges is of value to practitioners and academics alike. For practitioners, a better understanding of the challenges facing scaling efforts will better inform organizations as they elect to scale their agile implementations. Instead of framework-centric approaches, organizations will be able to better assess the challenges they may face and therefore make more informed decisions on which framework, if any, to choose when scaling.

For academics, the research provides multiple benefits. At the conclusion of the Delphi studies, the agile scaling factors will be compared to the extant research leveraging the work of Dikert et al. (2016). As the work completed by Dikert et al. (2016) is comprised of a literature review, comparing the practitioner-generated agile scaling factors to the academic research will help to identify where the literature supports practitioner efforts and where the research falls short of supporting practitioners. Procedures for this portion of the research will be detailed in future versions of this study due to space limitations.

Additionally, the results will also be compared to the existing research in multiteam systems. This comparison will not only provide validation of using multiteam systems research to help explain issues in scaling agile, but also help to identify future research needs within the multiteam systems research. Procedures for this portion of the research will be detailed in future versions of this study due to space limitations.

REFERENCES

1. Ambler, S. W. (2009). The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments. *Environments*, 1-35.

2. Bass, J. M. (2015). How Product Owner Teams Scale Agile Methods to Large Distributed Enterprises. *Empirical Software Engineering, 20*(6), 1525-1557.
3. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Principles Behind the Agile Manifesto*. Retrieved 10 19, 2018, from http://agilemanifesto.org/
4. Begel, A., & Nagappan, N. (2007). Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. *First International Symposium on Empirical Software Engineering and Measurement.* IEEE.
5. Bick, S., Scheerer, A., Spohrer, K., Kude, T., & Heinzl, A. (2014). Software Development in Multiteam Systems: Longitudinal Study on the Effects of Structural Incongruences on Coordination Effectiveness. *International Research Workshop on IT Project Management*, 48-56.
6. Brancheau, J. C., Janz, B. D., & Wetherbe, J. C. (1996). Key Issues in Information Systems Management: 1994-95 SIM Delphi Results. *MIS Quarterly, June*, 225-242.
7. Brislin, R. W. (1976). Comparative Research Methodology: Cross-Cultural Studies. *International Journal of Psychology, 11*(3), 215-229.
8. Buganza, T., Dell'Era, C., & Verganti, R. (2009). Exploring the Relationships Between Product Development and Environmental Turbulence: The Case of Mobile TLC Services. *Journal of Production Innovation Management, 26*, 308-321.
9. CollabNet VersionOne. (2019). *The 13th Annual State of Agile Report.*Retrieved 1 7, 2020, from CollabNet: https://www.stateofagile.com/?_ga=2.109536069.1962839535.1578432574-808394331.1578432574#ufh-i-521251909-13th-annual-state-of-agile-report/473508.
10. Denning, S. (2016, 4 15). *What Does it Mean to Scale Agile?* Retrieved 12 23, 2019, from Forbes: https://www.forbes.com/sites/stevedenning/2016/04/15/what-does-it-mean-to-scale-agile/#3b13926478b9
11. Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review. *The Journal of Systems and Software*, 87-108.
12. Dingsøyr, T., & Moe, N. B. (2013). Research Challenges in Large-Scale Agile Software Development. *SIGSOFT Software Engineering Notes, 38*, 38-39.
13. Dingsøyr, T., & Moe, N. B. (2014). Towards Principles of Large-Scale Agile Development: A Summary of the Workshop at XP2014 and a Revised Research Agenda. In T. Dingsøyr, N. B. Moe, S. Counsell, R. Ronelli, C. Gencel, & K. Petersen (Eds.), *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation* (Vol. 199, pp. 1-8). Berlin: Springer.
14. Dingsøyr, T., Fægri, T. E., & Itkonen, J. (2014). What is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development. In A. Jeditschka, P. Kuvaja, M. Kuhrmann, T. Männistö, M. Jürgen, & M. Raatikainen (Eds.), *Product-Focused Software Process Improvement* (Vol. 8892, pp. 273-276). Springer.
15. Dingsøyr, T., Moe, N. B., & Seim, E. A. (2018). Coordinating Knowledge Work in Multi-Team Programs: Findings From a Large-Scale Agile Development Program. *arXiv preprint arXi*.
16. Fry, C., & Greene, S. (2007). Large scale agile transformation in an on-demand world. In Agile 2007 (AGILE 2007) (pp. 136-142). IEEE.
17. Harris, M. L., Webb Collins, R., & Hevner, A. R. (2003). Control of Flexible Software Development Under Uncertainty. *Information Systems Research, 20*(3), 400-419.
18. Heikkilä, V. T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., & Järvinen, J. (2015). Operational Release Planning in Large-Scale Scrum with Multiple Stakeholders - A Longitudinal Case Study at F-Secure Corporation. *Information and Technology, 57*, 116-140.
19. Heusser, M. (2016, 01 19). *Can Agile Scale and Does it Matter?* Retrieved 12 23, 2019, from CIO: https://www.cio.com/article/3023903/can-agile-scale-and-does-it-matter.html
20. Keikkilä, V. T., Paasivaara, M., Lassenius, C., & Engblom, C. (2013). Continuous Release Planning in a Large-Scale Scrum Development Organization at Ericsson. *International Conference on Agile Software Development.* Berlin.
21. Mathieu, J. E., Marks, M. A., & Zaccaro, S. J. (2001). Multiteam Systems. In N. Anderson (Ed.), *Handbook of Industrial, Work & Organizational Psychology* (Vol. 2, pp. 289-313).

22. Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2016). Quantitatively measuring a large-scale agile transformation. Journal of Systems and Software, 117, 258-273.
23. Paasiaara, M., Lassenius, C., & Heikkilä, V. T. (2012). Inter-Team Coordination in Large-Scale Globally Distributed Scrum: Do Scrum-of-Scrums Really Work? *Proceedings of the ACM-IEEE Interational Symposium on Empirical Software Engineering and Measurement.*
24. Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2018). Large-Scale Agile Transformation at Ericsson: A Case Study. *Empirical Software Engineering, 23*, 2550-2596.
25. Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). DIstributed Agile Development: Using Scrum in a Large Project. *Global Engineering, 2008.*
26. Relihan, T. (2018, 7 6). *Agile at Scale, Explained.* Retrieved 12 23, 2019, from MIT Sloan: https://mitsloan.mit.edu/ideas-made-to-matter/agile-scale-explained
27. Rigby, D. K., Sutherland, J., & Noble, A. (2018, May). *Agile at Scale.* Retrieved 12 23, 2019, from Harvard Business Review: https://hbr.org/2018/05/agile-at-scale
28. Scheerer, A., & Kude, T. (2014). Exploring Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective. *Thirty-Fifth International Conference on Information Systems*, (pp. 1-11). Auckland.
29. Scheerer, A., Hildenbrand, T., & Kude, T. (2014). Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective. *47th Hawaii International Conference on System Science*, (pp. 4780-4788).
30. Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems, 17*(4), 5-36.
31. Shuffler, M. L., Jimènez-Rodriguez, M., & Kramer, W. S. (2015). The Science of Multiteam Systems: A Review and Future Research Agenda. *Small Group Research, 46*(6), 659-699.
32. Sutherland, J.J. (2019). *Scrum at John Deere: Continuing the Evolution Of A Smart Industrial*. Retrieved from ScrumInc 1 7, 2020: https://www.scruminc.com/scrum-john-deere-continuing-the-evolution-of-a-smart-industrial/.
33. Xu, P. (2009). Coordination in Large Agile Projects. *Review of Business Information Systems, 13*(4), 29-44.