DOI: 10.5748/19CONTECSI/PSE/ESD/7037

AGILE ROLES ADOPTED FOR LARGE-SCALE AND DISTRIBUTED TEAMS: A SYSTEMATIC LITERATURE REVIEW

Ivaldir Farias Junior
Universidade Rural Federal De Pernambuco - UFRPE

rafael marques miranda
Universidade Rural Federal De Pernambuco - UFRPE

Marcelo L. M. Marinho
Universidade Rural Federal De Pernambuco - UFRPE

AGILE ROLES ADOPTED FOR LARGE-SCALE AND DISTRIBUTED TEAMS: A SYSTEMATIC LITERATURE REVIEW

ABSTRACT

Companies have concentrated on the challenge of developing agile methods and establishing large-scale and global software development (GSD) teams. Research is scarce on which agile roles have been adapted. In this paper, we sought to identify which roles for large-scale and GSD teams had been adapted to agile methods and how the adaptations could be represented to assist researchers and practitioners in analyzing the adapted agile roles. We conducted a systematic review of the initial 1,520 studies, and 81 primary studies were chosen to be analyzed and evaluated for quality. The findings present a synthesis of 17 distinct relevant agile roles that organizations can adopt, such as ``System Managers'', ``cultural ambassadors'', and ``proxy product owners''.

Keywords: Agile roles, Agile methods, Large-scale, Global Software Development

1. INTRODUCTION

In large projects, global software development (GSD) is a reality. Large-scale development poses the challenge of maintaining quality, but global organizations must increase productivity. The industry's desire for greater productivity and quality has led it to seek agile methods on a large scale to achieve these goals, as agile practices can improve software quality and team productivity [19]. Large-scale distributed projects prioritize and specify requirements, as well as emphasize rich requirements documentation, whereas agile teams do not prioritize documentation [14].

1.1 Background

GSD is rapidly expanding and is quickly becoming the norm [7,23]. GSD is a term used to describe organizations that distribute their software development across multiple countries. Many industry organizations have turned to GSD for low-cost, high-quality software with a short development [13]. There are numerous advantages and motivations for using GSD, including access to a global talent pool, cost savings, and advancements in infrastructure and software development tools [8]. Because of the transition from traditional single-site development to a networked development environment, product development is becoming a complex, global enterprise with multiple stakeholders and activities involved [29].

Large organizations typically carry out large projects through large distributed development teams, needing a framework to scale agile methods. According to Leffingwell [21], in addition to the issues encountered in regular global projects, agile scaling entails several challenges, including coordination among multiple agile teams and the lack of an initial architecture and requirement analysis.

Agile methods appeal to large corporations due to their benefits, and potential [3, 25], and agile practices have been deemed very suitable for distributed projects distributed [24]. A large-scale agile organization can be defined by its number of software teams. Dingsøyrr et al. [10] taxonomy was used in this study to define the scale of agile software development projects based on the number of teams involved. ``Small-scale" projects should only have one team. ``Large-scale" projects can have 2 to 9 teams, requiring scaling methods for coordination. Finally, projects with more than ten teams are considered "very large-scale", and such projects should use a scaling framework.

1.2 Related work

Secondary studies on large-scale agile methods have investigated topics such as ontologies to support distributed software development [6], the global software development life cycle [16], the Scaled Agile Framework (SAFe) adoption [27], and agile adaptation impact factors in related studies such as those on agile practices in GSD [1].

Studies on roles in agile methods for GSD teams and large-scale agile are available [4, 5, 28]. Although significant publications and results have been obtained, they are tangential to it and sometimes address particular questions only in a correlational way, failing to achieve the same goal as the current study.

Ivarsson and Gorschek [15] presented the significant changes in Scrum roles and responsibilities as an integral part of a project's Agile-Scrum adaptation journey. This study was restricted to a single project that was solely focused on Agile Scrum adaptations.

Bass [4] mapped the scrum master adaptation’s role in large corporate projects. The author carried out case studies to present the various adaptations suffered by the scrum-master. In [5], Bass mapped how Product Owner (PO) teams scale agile methods for large distributed enterprises. The author conducted 45 interviews in eight companies and showed the adaptations in the PO paper industry.

Unger-Windeler et al. [28] realized a systematic mapping to provide insights into the PO's role in the industry in the form of an overview of the results pointing to the PO's role in large-scale projects in the industry, as well as to identify future research directions. %Our study investigated which roles are adapted in agile methods and how they are adapted in various scaling frameworks or methodologies.

Gustavsson [11] conducted a review in two large-scale agile development frameworks: Scaled Agile Framework (SAFe) and Large-Scale Scrum (LeSS). The author prescribed coordination practices and inter-team coordination roles and checked for additional roles.

Thus, researchers have confirmed the presence of adaptations in several agile roles for large companies in general. Additionally, secondary research on which and how agile roles are adopted and adapted by GSD teams in large-scale environments.

1.3 Objective and contribution

This research aimed to analyze the literature to close the gap in which the current literature does not provide a cohesive picture of which adaptations in agile roles are required in scaled-agile and GSD projects. We conducted a Systematic Literature Review (SLR) from 2001 to 2021 to identify research approaches that highlighted the adoption and adaptation of agile roles in the large-scale agile and GSD context. Throughout our investigation, we further developed the two key research questions: RQ1: What are the agile roles adopted by GSD and large-scale teams? and RQ2: How do GSD teams adapt agile roles according to project scale level?

A total of 81 primary studies were considered for this SLR. Coding based on the thematic synthesis of [9] was used to review and synthesize our qualitative research.

The remainder of this paper is organized as follows: Section 2 describes the method that we applied. Section 3 presents the results and an overview of the main agile roles adaptations. Section 4 presents the conclusions and directions for future research and finally Section 5 study recommendations.

2. RESEARCH METHOD

We conducted an SLR based on the procedures and guidelines proposed by Kitchenham and Charters \cite{keele2007guidelines} to identify which agile roles are adopted and how GSD teams in large-scale environments adapt agile roles. Our aim was to identify and choose a group of recurring adoptions and adaptations of studies rather than to find all adopted and adapted roles.

We created a review protocol, which can be found at https://tinyurl.com/2s3adnft, and we included the following steps to identify and select primary studies: 1) developing inclusion and exclusion criteria, 2) searching, 3) assessing quality, and 4) extracting and synthesizing data. The steps are detailed in the sections that follow.

2.1 Inclusion and exclusion criteria

The following criteria guided the selection of papers that helped us address the research questions.

We excluded: (i) Studies not written in English; (ii) Documents that are books, short papers (<=4 pages), theory papers, work- shop papers, technical reports; (iii) Studies that present personal viewpoints or specialists' opinions; (iv) Studies related to regular software development instead of distributed software development, scaling agile; (v) Studies related to education matters on distributed software development, scaling agile and large-scale software development; (vi) Studies not related to Software Engineering and computer science.

We included: (i) Studies that were published in journals and peer- reviewed conferences; (ii) Studies directly related to the research questions; (iii) Studies related to

distributed software development, scaling agile, agile roles, and large-scale software development.

2.2 Search process

We used an automatic search in five large bibliographic databases to identify a set of relevant papers that should match the research goals. By combining the keywords “global software development”, “scaling agile”, “large-scale”, and “roles”. The search string aimed to collect the keywords of each research theme. We used the following boolean search string to ensure that we captured a wide variety of papers: ("distributed software development" OR "distributed Software Engineering" GSE OR GSD OR "distributed teams" OR "dis-tributed team" OR "global software development" OR "global Software Engineering" OR "global team" OR "global teams" OR "offshore" OR "outsource" OR "DSD" OR "DSE") AND ("scaling agile" OR "scaled agile framework" OR SAFe OR Spotify OR Scrum@Scale OR Scrum OR Kanban OR Lean OR Nexus OR "large Scale Scrum" OR LeSS OR "agile programmer management" OR AgilePgM OR XP OR "Extreme Programming" OR "feature driven development" OR fdd OR "agile scale" OR "scale agile") AND (large OR scale OR "large-scale" OR "large scale" OR "large-scale development") AND (role OR roles OR “active roles” OR “active role” OR function OR functions OR assignments OR assignment OR responsibilities OR responsibility).

We used this string to search the IEEEXplore, ACM Digital Library, SpringerLink, Scopus, and Wiley bibliographic databases for metadata relating to journals and conference proceedings.

From 2001 to 2021, the search yielded 1520 references (IEEE =31, ACM = 836, Springer = 191, Scopus = 404, and Wiley = 58). Two researchers divided the papers considered for inclusion in the selection process into two stages: Phase 1: an initial screening of the search results based on reading the papers’ titles and abstracts eliminated papers that met some exclusion criteria or did not meet the inclusion criteria; and Phase 2: a final list of articles chosen based on reading their introductions and conclusions following the inclusion and exclusion criteria.

When this selection diverged, a third researcher arbitrated. The search string can be found in our full review protocol. Table 1 shows how many studies were extracted from each engine using the search string and how many were accepted in each phase of the extractions.

Table 1: Papers by engine

Engine	Selection	Phase 1	Phase 2
ACM	836	69	24
Scopus	404	78	16
Wiley	58	9	2
Spring	191	59	24
IEEE	31	26	16
Total	1520	241	81

2.3 Data extraction and synthesis

The extracting primary data process for the SLR is known as data extraction. We used it to extract data for answering research questions and then exported it to a spreadsheet for further data analysis.

We extracted the following information for each primary study: year, geographical distribution methods, study type, study method, scaling framework/method, distribution type, and company domain.

We also classified the studies using Dingsøyrr et al. [10] scaling taxonomy of agile software development projects. We classified each article according to one of the research types identified by Wieringa et al. [30]. Furthermore, all reviewed studies were classified using the contribution type aspects from Petersen et al. [26] study.

We used the study of [9] to classify and synthesize the relevant data using 'thematic synthesis', which is based on the principles of thematic analysis. In the previous systematic reviews, we identified recurring themes or issues from various studies for interpretation and explanation to draw conclusions, which we used to identify, analyze, and report patterns in the data.

2.4 Study Evaluation

We based the studies evaluation process on the method of rigor and relevance evaluation assessment proposed by Ivarsson and Gorschek [15]. The rigor aspects are evaluated on a scale of 0 (``weak''), 0.5 (``medium''), and 1 (``high''), and it has three dimensions which are (i) context, (ii) study design, and (iii) threats to validity. The maximum value of rigor is 3.

The industry relevance, on the other hand, is concerned with the impact a study can have on industry and academia, taking into account relevant research topics and real-world industry settings. The relevance aspect has a binary score, with 1 being present and 0 being absent. The aspects are (i) The subjects used in the assessment are representative of the intended users of the technology, ie, industry professionals; (ii) the context in which the study was conducted, such as industrial settings; (iii) the scale of the applications used in the study, such as realistic industrial applications; and (iv) the research method used. The maximum relevance value is 4.

3. RESULTS

3.1 Overview of the Primary Studies

Appendix A presents the list of 81 papers that passed through the different SLR phases. The primary studies are numbered in square brackets with S and a number. Figure 1 depicts the frequency distribution based on the year in which the 81 studies were published. It is worth noting that there was a gradual increase in occurrences from 2001 to 2012, indicating a trend toward a gradual increase in studies on the research topic. There were many occurrences from 2013 to 2014, which may have been reflected in the following year with the lowest contribution. Furthermore, there was a significant increase in the occurrences of studies from 2016 to 2019, demonstrating the study’s relevance to the academic environment and thus providing substantial inputs for the SLR.

Furthermore, multiple methods were used in the primary studies, with Case Study being the most commonly used method (43 papers), followed by grounded Theory (12 papers). Table 2 lists the set of study methods by article.

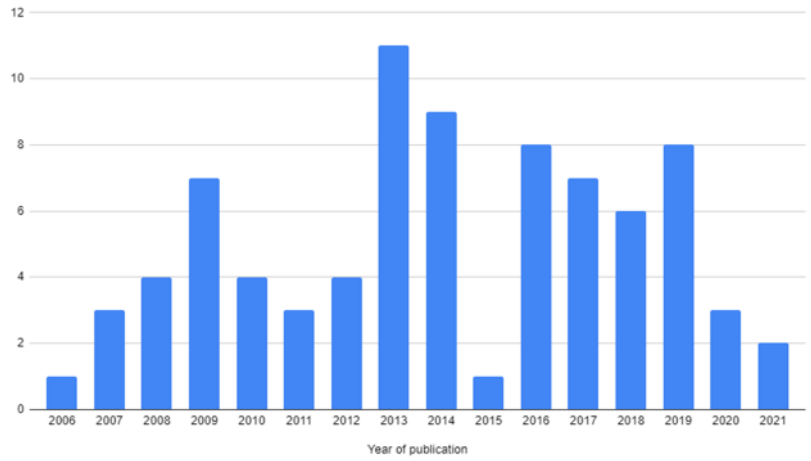

Figure 1: Number of Papers Included in RSL Over Time

Table 3 displays the data collection methods. Because some pa- pers used more than one method, the number of methods did not correspond to the number of papers. Most studies relied on interviews and observations, which is understandable given a large number of case studies.

Table 2: Study Methods by Papers

Study Method	ID_Paper	Occorrences
Case Study	[S2],[S3],[S4],[S5],[S9],[S10],[S11],[S12],[S13],[S14],[S15],[S16],[S17],[S19],[S20],[S21] [S22],[S23],[S24],[S27],[S28],[S32],[S33],[S38],[S39],[S42],[S46],[S47],[S48],[S49] [S50],[S55],[S53],[S61],[S62],[S69],[S70],[S71][S72],[S75],[S76],[S79],[S80]	43
Grounded Theory	[S14],[S16],[S23],[S26],[S32],[S33],[S43],[S44],[S48],[S60],[S73],[S74]	12
Experience report	[S26],[S29],[S30],[S34],[S35],[S36],[S45],[S51],[S54]	9
Multiple-Case Study	[S25],[S40],[S41],[S57],[S65],[S66],[S73],[S78]	8
Survey	[S6],[S18],[S22],[S53],[S68]	5
Exploratory Research	[S8],[S53],[S59],[S63]	4
Literature Review	[S16],[S56],[S75]	3
Theory	[S31],[S77],[S81]	3
Action Research	[S13],[S37]	2
Ethnography	[S38],[S47]	2

Table 3: Data Collection Methods

Methods	Paper ID	Occorrences
Interviewees	[S4],[S5],[S9],[S10],[S11],[S12],[S14],[S25],[S16],[S38],[S19],[S20],[S21],[S22],[S23],[S25],[S27], [S28],[S32],[S33],[S38],[S40],[S41],[S42],[S43],[S44],[S46],[S48],[S49],[S52],[S54],[S55],[S56], [S57],[S58],[S60],[S61],[S62],[S64],[S65],[S66],[S67],[S69],[S70],[S71],[S72],[S73],[S74],[S75] [S76],[S78],[S79],[S80]	53
Observations	[S40],[S41],[S21],[S23],[S27],[S28],[S32],[S37],[S38],[S40],[S42],[S43],[S44],[S47],[S41],[S55], [S62],[S66],[S72],[S79],[S80]	21
Questionnaires	[S39],[S45],[S49],[S53],[S58],[S63],[S65],[S68]	8
Projects documentation	[S42],[S44],[S55],[S69],[S70],[S73],[S75],[S76]	8
Workshops	[S25],[S26],[S38],[S41],[S21],[S39],[S63]	7
Literature	[S75],[S29],[S30],[S31],[S77],[S81]	6
Documents	[S43],[S37],[S40],[S56],[S66]	5
Survey	[S53],[S27],[S40],[S53],[S79]	5
Focus Group	[S53],[S40],[S55],[S65]	4
Documents	[S38],[S28],[S38]	3
Others	[S1],[S2],[S4],[S6],[S7],[S8],[S26],[S34],[S35],[S36]	10

Regarding the study design, the vast majority of articles (66 papers) used a qualitative approach, followed by a mixed approach (11 papers), and finally, a quantitative method (4 papers).

In Figure 2, we present the research facets. Many evaluation studies (57 papers) were identified, indicating that evaluating implementation requires more than just a demonstrative case study. Following experience (14 papers), philosophical (6 papers), and solution techniques (4 papers) were observed.

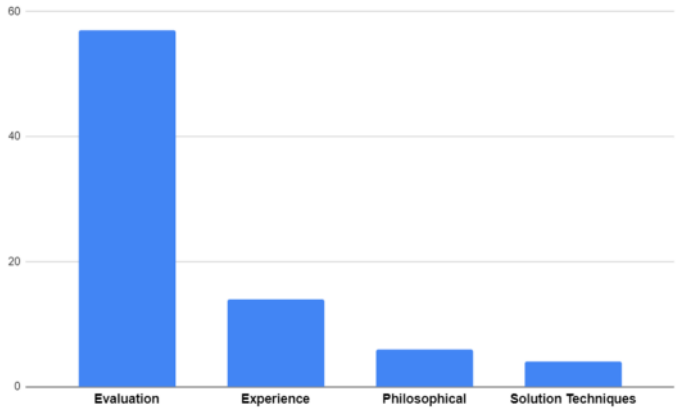

Figure 2: Research type facets

Figure 3 presents the most common contribution type was lessons learned (57 papers), followed by model and framework (8 papers each), guideline and theory (3 papers each), and finally, advice (3 papers) (2 papers).

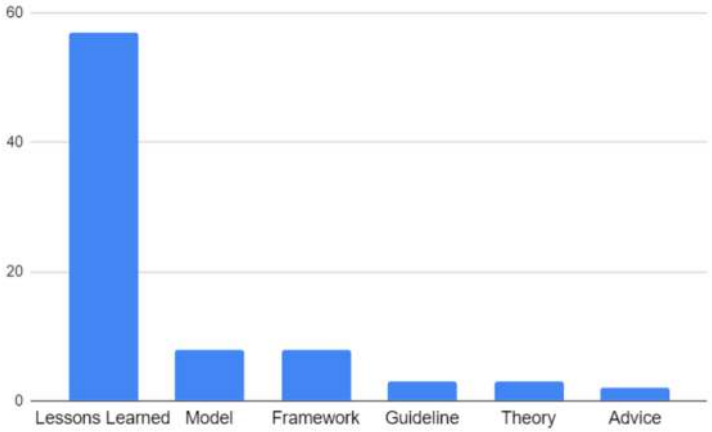

Figure 3: Contribution type facets

3.2 Rigor and relevance

We calculated the rigor and relevance of the studies according to Ivarsson and Gorschek [15]. Thus, we identified: lowest 20%, poor 1 paper; 20% to 40%, fair (4 papers); 40% to 60%, average (6 papers); 60% to 80%, good (26 papers; and highest 80%), AND excellent (44 papers). We realized that most of the papers exhibited relevance of the topic to the industry.

3.3 RQ1: What are the agile roles adopted by GSD and large-scale teams?

We followed the recommendations of [9] for data synthesis, which defines the recommended steps based on thematic analysis principles. These suggestions assisted us in identifying the agile roles adopted by the SLR papers.

We found numerous agile roles adopted for large-scale agile methods in GSD, with 17 being identified repeatedly in multiple studies. Table 4 summarizes the roles according to the scaling frame- work/method.

We summarize, according to the primary studies, and present below the description of the roles found in GSDs and large projects.

- Program Manager: guide and coordinate teams linked to var- ious projects in the portfolio. Typical activities include supporting release planning, tracking program

dependencies and risks, and assisting with contract management with vendors and external contractors.

- Agile Team/Development team: is a cross-functional group of individuals who define, build, test, and deliver an increment of value. 4.3.3 Agile project manager: act as consultants, allocating the right people, processes, and resources to reinforce team effectiveness and efficiency.
- Product Owner: is an agile team member responsible for defining Stories and prioritizing the Team Backlog. As a project grows to include multiple teams, ideally, a new product owner is found for each team. So, there are multiple POs in a large project with multiple teams.
- 4.3.5 Scrum Master/Agile Coach/Team Leader: is a servant leader and coach for an agile team. It helps educate the team on Scrum, Extreme Programming (XP), and Kanban, ensuring that the agreed agile process is being followed.
- Business Owners: a small stakeholder's group with primary technical and business responsibility for governance, compliance, and return on investment (ROI) for a team-built solution.
- System Architect: is responsible for defining and communicating a shared technical and architectural vision to teams to ensure that the system or solution under development is fit for purpose.
- Area Product Owner: specializes in a customer-centric area and acts as Product Owner with the teams in that area. An Area Product Owner (APO) does the same job as a PO, but with a more limited but still customer-centric perspective.
- Cultural Ambassador: is one member from each of the dis- tributed teams' cultures to serve as a "cultural ambassador" who can interpret the remote team's communication and actions and support the remote teams.
- Proxy Product Owner: it is an intermediary role between the people who make decisions about a product, the sometimes very busy Product Owner, and the people who develop it.
- Chief Product Owner: is responsible for having an overview of the entire product or set of products.
- Tech lead: The technical lead must work with the PO to help ensure the viability of the customer's ideas. They are responsible for adequately prioritizing the resolution of bugs and other technical debts.
- Product Team: involves specialists and managers for a new product, process, or project team who have the authority to make important decisions about the product.
- Solution Architect: is responsible for defining and communicating a shared technical and architectural vision of various pro- grams to help ensure that the system or solution under development is fit for purpose.
- Lean Agile Portfolio Manager: is responsible for providing oversight for value-driven governance at the portfolio level for product deliveries across the organization. Paper is crucial to ensuring that all portfolio investments are aligned with business strategy and meet short/long-term business needs and value expectations.
- System Architect: is a technical expert who supports architectural decisions for the team and facilitates the creation and evolution of the overall solution design.
- Release Train Engineer: a server leader makes it easy to run programs, remove impediments, manage risks and dependencies, and continually improve,

Table 4: The most recurrent roles in agile methods for large-scale and GSD teams

Roles	Scaling Framework / Method	Papers	Occurrences
Agile team/ Development Team	Scrum, LeSS, Agile in General, SAFe, XP, kanban, Lean;	[S3],[S5],[S7],[S10],[S11],[S12],[S13],[S16],[S24], [S25],[S27],[S30],[S32],[S35],[S44],[S50],[S51], [S57],[S60],[S61],[S64],[S65],[S67],[S68],[S69], [S70],[S73],[S76],[S79],[S80],[S81]	31
Agile Project Managers	Scrum, Agile in general,XP, Kanban, LeSS, SAfe;	[S2],[S5],[S7],[S11],[S22],[S24],[S25],[S26],[S28], [S29],[S30],[S34],[S36],[S37],[S38],[S48],[S49], [S51],[S52],[S55],[S56],[S57],[S60],[S65],[S68], [S69],[S73],[S76],[S77],[S79]	30
Scrum Master/ Agile Coach/ Team Leader	Scrum, SAFe, XP, Agile in general, Kanban;	[S2],[S3],[S7],[S15],[S20],[S22],[S28],[S30],[S32], [S34],[S35],[S49],[S50],[S51],[S60],[S61],[S64], [S68],[S70],[S72],[S73],[S74],[S76],[S78],[S79]	25
Program Manager	LeSS, DAD, Scrum, Agile in General, SAfe, Kanban, XP;	[S3],[S4],[S5],[S7],[S11],[S16],[S22],[S25], [S26],[S28],[S30],[S34],[S37],[S38],[S50], [S53],[S58],[S59],[S61],[S65],[S66],[S68]	23
Product Owner	Scrum, SAFe, DAD, Agile in general, XP;	S1],[S5],[S6],[S10],[S12],[S15],[S16], [S30],[S32],[S44],[S47],[S53],[S57],[S61], [S64],[S66],[S71],[S72],[S73],[S75]	20
Business Owner	LeSS, XP, Scrum, Agile in general, SAFe, Kanban;	[S1],[S32],[S38],[S51],[S53],[S60],[S68], [S73],[S76]	9
System Architect	Agile in general, Scrum, LeSS, Kanban, SAFe, XP;	[S1],[S10],[S11],[S16][S22],[S65],[S68],[S69],[S73]	9
Area Product Owner	LeSS,Scrum,XP;	[S1],[S10],[S11],[S22],[S44], [S51]	6
Cultural Ambassador	Scrum, XP;	[S12], [S24],[S60],[S64],[S79],[S81]	6
Proxy Product Owner	Scrum,Less;	[S3],[S5],[S44],[S10],[S30]	5
Chief Product Owner	Scrum, LeSS;	[S10],[S20],[S34],[S71]	4
Tech Lead	Scrum, Agile in general;	[S20],[S66],[S71],[S73]	4
Product Team	Scrum;	[S34],[S37],[S50],[S71]	4
Solution Architect	LeSS, Scrum;	[S1],[S10],[S11]	3
Lean Agile Portfolio Manager	Scrum, Agile in general, DAD;	[S7],[S28] [S73]	3
Release Train Engineer	SAFe;	[S2],[S26],[S4]	3

3.4 RQ2: How do GSD teams adapt agile roles according to project scale level?

We identified scale agile frameworks and agile methodologies that had adapted roles in GSD and large-scale environments in the selected papers (See Table 5). By classifying only recurring roles in the various studies, we ensured that only roles not specific to the organization's context were included in the final set. In the following section, we observed some projects of small-scale, large-scale, and very-large-scale organizations in agile and GSD contexts.

Table 5: Role's adaptation by Scaling Framework/Methodology

Scaling Framework / Methodology	Reference	Occurrences
Scrum	[S3],[S5],[S7],[S9],[S11],[S12],[S15],[S17],[S22],[S25],[S30],[S32],[S33],[S34],[S35],[S36],[S37],[S38], [S39],[S40],[S41],[S42],[S44],[S46],[S47],[S48],[S49],[S50],[S52],[S54],[S56],[S57],[S58],[S60], [S61],[S64],[S67],[S68],[S70],[S71],[S72],[S73],[S74],[S76],[S78],[S79]	46
SaFE	[S2],[S4],[S6],[S13],[S18],[S26],[S36],[S53],[S66]	9
XP	[S10],[S32],[S33],[S38],[S44],[S51],[S60],[S68],[S74]	9
Agile in General	[S26],[S27],[S49],[S55],[S62],[S65],[S69],[S73]	8
NA	[S24],[S29],[S40],[S63],[S77],[S80]	6
Kanban	[S33],[S35],[S43],[S44],[S68]	5
DAD	[S28],[S31],[S62],[S66]	4
LeSS	[S1], [S11]	2
Spotify	[S22],[S23]	2

3.4.1 Small-scale agile in GSD context.

In the context of small-scale, defined by Dingsøyr et al. [10] we found articles that addressed the adaptations of the Product Owner and the Team.

In [S47] it was observed that the Product Owner (PO) should have a strong knowledge of the business and the technology to support the distribution of activities in the teams.

In this setting, the Agile Team or Development Team was divided into micro-teams, for instance, 2 to 3 developers and a tester [S13]. According to [S13], the daily meetings must ensure the right amount of information is shared. The scrum master must strike a balance between the project team and business partner leadership, as well as distance, trust, and cultural diversity [S51]. Another critical aspect of this relationship is that there are agile project managers; however, they should not take on all of the responsibility; instead, teams must establish their leaders, and everyone on the team needs to know that they can participate in decision-making.

3.4.2 Large-scale agile and GSD context.

Regarding the Large-scale agile projects and GSD context, we have identified the following adapted roles: cultural ambassador, Scrum Master or agile coach or team Leader, systems architect and agile project manager. According to [S61], Product Owner (PO) is shared with multiple PO's. In [S72], it is reported that POs were not considered team members, possibly because they traveled a lot collecting requirements, but were customer representatives who occasionally attended sprint planning and review meetings to manage improvements for proper implementation.

The roles of cultural ambassador, business owner and agile coach should also be mentioned. The cultural ambassador visits the team to report lessons learned and plan future actions for the project [S64]. They participate in the daily meetings and retrospectives of the visited team. Cultural ambassadors provide the offshore team with information about the business context [S12]. The primary responsibility of cultural ambassadors is to understand team members at the location they are sent to [S60]. The business owner is responsible for handling the Scrum team's issues and requirements [S1], as well as being the first point of contact for external parties to document the business process and data requirements in the form of customer stories. users [S1]. The Scrum Master or agile coach or team Leader facilitates communication between teams, with coaches subdivided into other teams to moderate work towards a common project goal [S30].

The systems architect has been repeatedly discovered as a member of the group in charge of implementing the changes and requirements of the product backlog, ensuring that everything is organized and agreed before the start of the next Sprint [S76]. The agile project manager's responsibilities include coordinating the development team and planning the budget and capabilities [S1]

3.4.3 Very large-scale agile in GSD context.

Regarding the very large-scale agile projects and GSD context, we have identified the following adapted roles: product team, Lean agile portfolio manager, program manager, agile project management, product owner, Proxy Product Owner (Proxy PO), Area PO (APO), Solution Architect, system architect, chief Product owner, Scrum Master/ Agile Coach/Team Leader, Release Train Engineers (RTE) and tech lead.

The product team's daily duties include identifying market opportunities, prioritizing based on customer needs, building ideas to address these opportunities, and testing the solution's efficiency. In contrast, an integrated product team includes strategic development, competitive analysis, defining roadmaps, and observing product backlogs to reap benefits from previous investments [S34], [S37], [S50], [S71].

Management roles are another recurring role that is adapted on a large scale. The Lean agile portfolio manager is in charge of a group of client projects managed as a portfolio [S73]. The program manager updates the backlog, and the system managers work on the overall system level [S11]. Furthermore, agile project management can define the project's schedule and scope while balancing this with timely and regular value deliveries and organizing and leading working and project status meetings [S11].

In large-scale agile, the product owner role evolves into a product owner team [S1, S32, S34]. A product sponsor needs to surround itself with a team to liaise with the many stakeholders in the development program. A Proxy Product Owner (Proxy PO) is an intermediary between the people who make decisions about a product and those who develop it. A PO Proxy usually performs activities that a Product Owner usually does. However, a Proxy PO is an incomplete version of a Product Owner. In other words, Proxy PO plans to carry out the Backlog with the team; however, it does not define either the product's vision or its strategy [S44].

In [S10] is presented the Area PO (APO) who is responsible for a subset of the product features. APO roles subdivide into two roles: a systems architect and a Solution Architect who is a product management representative who could have commercial or technical experience [S11].

The Solution Architect is responsible for specifying and communicating a shared technical and architectural vision across multiple "time groups" to help ensure that the system or solution under development is fit for purpose.

The system architect activity is used to support the product owner's team members who are conventionally more business- oriented. Architectural patterns and reference architecture need technical knowledge to coordinate agile teams [S1, S16, S22]. The system architect establishes and refines conventions for structuring large-scale software systems, supporting team members and helping to disseminate best practices [S10, S65, S73].

Each APO pair is in charge of resources in a specific product area, such as billing and policy application, and collaborates with the teams that create these resources [S11]. The chief Product owner, on the other hand, has the final say in prioritization decisions [S10].

We verified additional roles for the Scrum Master/ Agile Coach/Team Leader in terms of how to verify the assignment of user stories in order to minimize interference between different teams, which still plays an essential role during the scrum of scrums meeting [S4]. Each scrum master will report the status of their team [S5], and this meeting is used to tactically manage and coordinate iteration progress across the various Scrum teams [S4] [S6]. Thus, the most important activity for a scrum master is the process's anchor activity: owning and disseminating the Scrum process within the development team [S4].

As a result, while scaling up, the scrum master will devote more time to communication and coordination between teams for dependencies [S8], such as resolving conflicts between merged code modules. Moreover, Release Train Engineers (RTE) had a significant overlap and were prepared considering these role' SAFe guidelines and business focus [S26].

Furthermore, we find in evidence the tech lead role, who is re- sponsible for guiding the team in completing its tasks and is the first to respond to the technical obstacles raised by the team. In addition, the tech lead can provide technical standards for the team.

4. RECOMMENDATIONS

This section presents recommendations from the data collected during the systematic literature review.

Therefore, the recommendations below serve as a framework for consideration when adapting. Together with the data collected in the RSL are inputs that can help improve adoption and better deal with possible adaptations of agile roles in large-scale or distributed projects.

- Recommendation 1: Level of scaling is Important: When adopting agile methods consider the scaling level because it impacts how these agile roles can be used. The company needs to verify how the role is adapted in the distributed team context and agile scaling level.
- ● Recommendation 2: Open to new responsibilities: The distributed agility and large team should not act following the book; instead, they should be open to new

attributes. Recommendation 3: Agile team change planning and organization: Changes must be carefully planned, implemented in stages, and avoided altogether. Allowing time for training, comprehension, testing of the new structures, tasks, division of responsibilities, or combination of duties, and support of the required tools.

- Recommendation 4: Adapting to the context: Changes should be carefully planned, implemented in stages, and avoided altogether.

5. CONCLUSIONS

Large-scale agile methods are used where various teams collaborate on a shared development program over a long period. Large-scale development programs often involve a complex mix of technologies and a wide range of stakeholders. Furthermore, large scale often goes hand in hand with geographic distribution.

Initially, agile methods were associated with small co-located teams. However, the compelling ability of agile methods to respond to changing priorities and mitigate risk spurs adoption in large-scale, distributed team settings. While agile teams are self- organizing [5], collaborative, agile teams need to sacrifice some autonomy to work with each other.

There are some frameworks for large-scale agile, such as Scaled Agile Framework (SAFe) [22], Large Scale Scrum (Less) [20] and Disciplined Agile (DA) [2]. However, we observed that Scrum is the method that has its most adopted roles. In the distributed con- text, the team undergoes some adaptations to deal with disparate members and to deal with several teams, so roles arise in order to orchestrate the team development. Furthermore, more mature adopters often adapt and evolve their unique approach.

Agile enthusiasts identify roles as defining aspects of software development processes [12]. Development roles need to evolve when carrying out large-scale development programs [10].

This paper's aim was to assist researchers and professionals in the analysis of agile roles in large-scale GSD environments and how these roles were adapted according to the project scale level. For this, we carried out a systematic literature review from 2001 to 2021. Thus, 81 studies were selected to answer the research questions.

We showed a set of 17 roles used in this context and that the most frequent adaptations occur in very-large scale agile context. More- over, we identify that adaptations for agile roles include adding new roles to deal with distributed and global teams. We verified that the PO role appears quite frequently in papers. We identified that in the context of large-scale and distributed teams are led by several POs and found some new adaptations for the PO role, such as the APO and the proxy PO.

This study has presented an overview of what adopted agile roles and how the agile roles are adapted for large-scale teams and GSD. A total of 81 research studies published from 2001 to 2021 were selected. The selected studies were subsequently classified and analyzed according to the SLR protocol. We presented a set of adapted agile roles that were observed to be recurring in different project contexts. It was revealed that Product Owner and Scrum Master roles are adapted in large-scale GSD teams.

The results provided 17 distinct roles that organizations can implement, such as agile project managers, cultural ambassadors, and lean agile portfolio manager. In a future study, we intend to evaluate the mapping of role adaptations in a real industrial development project. Furthermore, we plan to survey distributed agile teams to determine whether the typical GSD roles are being used in large-scale distributed projects. Further research is required to address the challenges of adapting agile roles in large-scale and GSD teams, identify best practices in organizations, contribute to better adaptation to agile, and investigate whether a correlation exists between roles and scaling frameworks.

5.1 Limitations

In this study, a strict SLR was used. The main limitation of such studies is the possibility of bias, although the systematic process of writing SLRs is designed to avoid bias [17]. To avoid this bias, we defined and refined appropriate inclusion and exclusion criteria, and the authors decided whether or not to include the studies at each stage of the SLR. When necessary, the third researcher resolved the disagreements. A search was conducted using five different sources to cover all relevant studies: ACM Digital Library, IEEE Xplore, Science Direct, Springer, and Wiley Online Library, which included one more than mentioned in the studies of [18]. We also evaluated and discussed the quality/validity of the included studies. The contexts of the studies were carefully extracted into a spreadsheet and peer-reviewed

6. REFERENCES

[1] Areebah Altaf, Urooj Fatima, Wasi Haider Butt, Muhammad Waseem Anwar, and Maryum Hamdani. 2019. A systematic literature review on factors impacting agile adaptation in global software development. In Proceedings of the 2019 7th International Conference On Computer and Communications Management. 158–163.

[2] Scott W Ambler and Mark Lines. 2012. Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise. IBM press.

[3] Julian M Bass. 2012. Influences on agile practice tailoring in enterprise software development. In 2012 Agile India. IEEE, 1–9.

[4] Julian M Bass. 2014. Scrum master activities: process tailoring in large enterprise projects. In 2014 ieee 9th international conference on global software engineering. IEEE, 6–15.

[5] Julian M Bass. 2015. How product owner teams scale agile methods to large distributed enterprises. Empirical software engineering 20, 6 (2015), 1525–1557.

[6] Alex Borges, Sérgio Soares, Silvio Meira, Hilário Tomaz, Rodrigo Rocha, and Catarina Costa. 2013. Ontologies supporting the distributed software development: a systematic mapping study. In Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering. 153–164.

[7] Rafael Camara, Annelyelthon Alves, Iury Monte, and Marcelo Marinho. 2020. Agile global software development: A systematic literature review. In Proceedings of the 34th Brazilian Symposium on Software Engineering. 31–40.

[8] Erran Carmel and Paul Tjia. 2005. Offshoring information technology: Sourcing and outsourcing to a global workforce. Cambridge university press.

[9] Daniela S Cruzes and Tore Dyba. 2011. Recommended steps for thematic synthesis in software engineering. In 2011 international symposium on empirical software engineering and measurement. IEEE, 275–284.

[10] Torgeir Dingsøyr, Tor Erlend Fægri, and Juha Itkonen. 2014. What is large in large-scale? A taxonomy of scale for agile software development. In International Conference on Product-Focused Software Process Improvement. Springer, 273–276.

[11] Tomas Gustavsson. 2017. Assigned roles for Inter-team coordination in Large-Scale Agile Development: a literature review. In Proceedings of the XP2017 Scientific Workshops. 1–5.

[12] Rashina Hoda, James Noble, and Stuart Marshall. 2012. Self-organizing roles on agile software development teams. IEEE Transactions on Software Engineering 39, 3 (2012), 422–444.

[13] Steinar Hole and Nils Brede Moe. 2008. A case study of coordination in distributed agile software development. In European Conference on Software process improvement. Springer, 189–200.

[14] Syeda Sumbul Hossain. 2019. Challenges and mitigation strategies in reusing requirements in large-scale distributed agile software development: a survey

result. In Intelligent Computing-Proceedings of the Computing Conference. Springer, 920–935.
[15] Martin Ivarsson and Tony Gorschek. 2011. A method for evaluating rigor and industrial relevance of technology evaluations. Empirical Software Engineering 16, 3 (2011), 365–395.
[16] Ritu Jain and Ugrasen Suman. 2015. A systematic literature review on global software development life cycle. ACM SIGSOFT Software Engineering Notes 40, 2 (2015), 1–14.
[17] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. Keele, UK, Keele University 33, 2004 (2004), 1–26.
[18] B. Kitchenham and S. Charters. 2007. Guidelines for performing systematic literature reviews in software engineering. Technical Report. EBSE Technical Report.
[19] HELMUT Kopka and Patrick W Daly. 1999. Advances in Knowledge-Based and Intelligent Information and Engineering Systems, Part1, Harlow, England.
[20] Craig Larman and Bas Vodde. 2016. Large-scale scrum: More with LeSS. Addison-Wesley Professional.
[21] Dean Leffingwell. 2007. Scaling software agility: best practices for large enterprises. Pearson Education.
[22] Leffingwell, Dean. 2022. Scaled Agile Framework. https://www.scaledagileframework.com/. [Online; accessed 16-July-2022].
[23] Marcelo Marinho, John Noll, and Sarah Beecham. 2018. Uncertainty management for global software development teams. In 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC). IEEE, 238–246.
[24] Maria Paasivaara, Sandra Durasiewicz, and Casper Lassenius. 2008. Distributed agile development: Using scrum in a large project. In 2008 IEEE International Conference on Global Software Engineering. IEEE, 87–95.
[25] Maria Paasivaara and Casper Lassenius. 2016. Scaling scrum in a large globally distributed organization: A case study. In 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE). IEEE, 74–83.
[26] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering. In 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12. 1–10.
[27] Abheeshta Putta, Maria Paasivaara, and Casper Lassenius. 2018. Adopting scaled agile framework (SAFe) a multivocal literature review. In Proceedings of the 19th International Conference on Agile Software Development: Companion. 1–4.
[28] Carolin Unger-Windeler, Jil Klünder, and Kurt Schneider. 2019. A mapping study on product owners in industry: identifying future research directions. In 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP). IEEE, 135–144.
[29] Antti Välimäki and Jukka Kääriäinen. 2008. Patterns for distributed scrum—a case study. In Enterprise interoperability III. Springer, 85–97.
[30] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements engineering 11, 1 (2006), 102–107.

7. APPENDICE A

A.1 Primaries Study Set

[S1]Uludağ, Ömer, et al. Ïnvestigating the adoption and application of large-scale scrum at a German automobile manufacturer.2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE). IEEE, 2019.

[S2]Ebert, C.,Paasivaara, M. (2017). Scaling agile. Ieee Software, 34(6), 98-103.

[S3] Bass, Julian M. Scrum master activities: process tailoring in large enterprise projects.2014 ieee 9th international conference on global Software Engineering. IEEE, 2014.

[S4] Paasivaara, Maria. Ädopting SAFe to scale agile in a globally distributed organization.2017 IEEE 12th International Conference on Global Software Engineering (ICGSE). IEEE, 2017.

[S5] Bass, Julian M. Ägile method tailoring in distributed enterprises: Product owner teams.2013 IEEE 8th International Conference on Global Software Engineering. IEEE, 2013.

[S6] Razzak, Mohammad Abdur, et al. SScaling agile across the global organization: an early stage industrial SAFe self-assessment.2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE). IEEE, 2018.

[S7] Jha, Madan Mohan, Rosa Maria Ferrer Vilardell, and Jai Narayan. SScaling agile scrum software development: providing agility and quality to platform development by reducing time to market.2016 IEEE 11th international conference on global Software Engineering (ICGSE).IEEE, 2016

[S8] Noll, John, et al. Ägile practices for the global teaming model.2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW). IEEE, 2016.

[S9] Bass, Julian M. Ïnfluences on agile practice tailoring in enterprise software development.2012 Agile India. IEEE, 2012.

[S10] Paasivaara, Maria, Ville T. Heikkilä, and Casper Lassenius. Ëxperiences in scaling the product
owner role in large-scale globally distributed scrum.2012 IEEE Seventh International Conference on Global Software Engineering. IEEE, 2012.

[S11] Paasivaara, Maria, and Casper Lassenius. SScaling scrum in a large globally distributed organi-zation: A case study.2016 IEEE 11th International Conference on Global Software Engineering (ICGSE). IEEE, 2016.

[S12] Paasivaara, Maria, Sandra Durasiewicz, and Casper Lassenius. "Distributed agile development: Using scrum in a large project.2008 IEEE International Conference on Global Software Engineering. IEEE, 2008.

[S13] Vallon, Raoul, et al. Ädapting to changes in a project's dna: A descriptive case study on the effects of transforming agile single-site to distributed software development.2014 Agile Conference. IEEE, 2014.

[S14] Salameh, Abdallah, and Julian Bass. SSpotify tailoring for B2B product development.2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 2019.

[S15] Fitzgerald, Brian, et al. SScaling agile methods to regulated environments: An industry case study.2013 35th International Conference on Software Engineering (ICSE). IEEE, 2013.

[S16] Martini, Antonio, and Jan Bosch. Ä multiple case study of continuous architecting in large agile companies: current gaps and the CAFFEA framework.2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA). IEEE, 2016

[S17] Bass, Julian M. "Large-scale offshore agile tailoring: exploring product and service organisa-tions. Proceedings of the Scientific Workshop Proceedings of XP2016. 2016.

[S18] Bass, Martini, Antonio, Lars Pareto, and Jan Bosch. "Communication factors for speed and reuse in large-scale agile software development.Proceedings of the 17th international software product line conference. 2013.

[S19] Tendedez, Helena, Maria Angela Felicita Cristina Ferrario, and Jonathan Nicholas David Whittle.SSoftware development and CSCW: standardization and flexibility in large-scale agile devel-opment.Proceedings of the ACM on Human-Computer Interaction-CSCW 2.CSCW (2018).

[S20] Nyrud, Helga, and Viktoria Stray. Ïnter-team coordination mechanisms in large-scale agile.Proceedings of the XP2017 scientificworkshops. 2017.

[S21] Rolland, Knut H. SScaling across knowledge boundaries: A case study of a large-scale agile software development project.Proceedings of the Scientific Workshop Proceedings of XP2016. 2016.

[S22] Moe, Nils Brede, et al. "Networking in a large-scale distributed agile project.Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. 2014.

[S23] Salameh, Abdallah, and Julian M. Bass. "Heterogeneous tailoring approach using the spotify model.Proceedings of the Evaluation and Assessment in Software Engineering. 2020. 293-298

[S24] Batra, Dinesh. "Modified agile practices for outsourced software projects.Communications of the ACM 52.9 (2009): 143-148.

[S25] Hossain, Emam, Paul L. Bannerman, and Ross Jeffery. "Towards an understanding of tailoring scrum in global software development: a multi-case study.Proceedings of the 2011 International Conference on Software and Systems Process. 2011.

[S26] Pandya, Alaukikdev, V. S. Mani, and Ansuman Pattanayak. Ëxpanding the responsibility of an offshore team and sustainably increasing business value using SAFe.Proceedings of the 15th International Conference on Global Software Engineering. 2020.

[S27] Lous, Pernille, et al. "Virtual by design: How a work environment can support agile distributed software development.2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE). IEEE, 2018.

[S28] Lal, Ramesh, and Tony Clear. Ënhancing product and service capability through scaling agility in a global software vendor environment.Proceedings of the 13th International Conference on Global Software Engineering. 2018.

[S29] Gupta, Rajeev Kumar, and Tulasi Anand. "Knowledge transfer for global roles in GSE.2017 IEEE 12th International Conference on Global Software Engineering (ICGSE). IEEE, 2017.

[S30] Gupta, Rajeev Kumar, Prabhulinga Manikreddy, and K. C. Arya. "Pragmatic Scrum Transformation: Challenges, Practices Impacts During the Journey A case study in a multi-location legacy software product development team." Proceedings of the 10th Innovations in Software Engineering Conference. 2017.

[S31] Brown, Alan W., Scott Ambler, and Walker Royce. Ägility at scale: economic governance, measured improvement, and disciplined delivery.2013 35th International Conference on Software Engineering (ICSE). IEEE, 2013.

[S32] Hoda, Rashina, et al. Ägility in context.Proceedings of the ACM international conference on Object oriented programming systems languages and applications. 2010.

[S33] Hoda, Rashina, et al. Ägility in context.Proceedings of the ACM international conference on Object oriented programming systems languages and applications. 2010

[S34] Gupta, Rajeev Kumar, Shivani Jain, and Bharat Singh. "Challenges in scaling up a globally distributed legacy product: A case study of a matrix organization.Proceedings of the 13th International Conference on Global Software Engineering. 2018

[S35] Godoy, Cristiano P., et al. "Blueprint model: A new approach to SCRUM agile methodology.2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE). IEEE, 2019.

[S36] Gupta, Rajeev Kumar, et al. "Key factors in scaling up agile team in matrix organization.Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference). 2019.

[S37] Gupta, Rajeev Kumar, Mekanathan Venkatachalapathy, and Ferose Khan Jeberla. "Challenges in adopting continuous delivery and devops in a globally distributed product team: a case study of a healthcare organization. 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE). IEEE, 2019.

[S38] Matthiesen, Stina, and Pernille Bjørn. "When distribution of tasks and skills are fundamentally problematic: A failure story from global software outsourcing.Proceedings of the ACM on Human Computer Interaction 1.CSCW (2017): 1-16.

[S39] Laukkanen, Eero, et al. "Bottom-up adoption of continuous delivery in a stage-gate managed software organization. Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. 2016.

[S40] Garbajosa, Juan, Agustín Yagüe, and Eloy González. "Communication in agile global software development: An exploratory study.OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, Berlin, Heidelberg, 2014.

[S41] Paasivaara, Maria, and Casper Lassenius. Üsing Scrum practices in GSD projects.Agility across time and space. Springer, Berlin, Heidelberg, 2010. 259-278.

[S42] Vallon, Raoul, et al. Ïnter-organizational co-development with scrum: experiences and lessons learned from a distributed corporate development environment.International Conference on Agile Software Development.Springer, Berlin, Heidelberg, 2013.

[S43] Rahy, Scarlet, and Julian Bass. Ïnformation flows at inter-team boundaries in agile information systems development.European, Mediterranean, and Middle Eastern Conference on Information Systems. Springer, Cham, 2018. SBQS ’22, November 07–10, 2022, Curitiba, PR MIRANDA, R. M et. al

[S44] Bass, Julian M. "How product owner teams scale agile methods to large distributed enterprises.Empirical Software Engineering 20.6 (2015): 1525-1557.

[S45] Kommeren, Rob, and Päivi Parviainen. "Philips experiences in global distributed software development. Empirical Software Engineering 12.6 (2007): 647-660.

[S46] Korkala, Mikko, Minna Pikkarainen, and Kieran Conboy. "Distributed agile development: A case study of customer communication challenges.International Conference on Agile Processes and Extreme Programming in Software Engineering. Springer, Berlin,Heidelberg, 2009.

[S47] Koch, Christian, et al. "We All Know How, Don’t We? On the Role of Scrum in IT-Offshoring.International Working Conference on Transfer and Diffusion of IT. Springer, Berlin, Heidelberg, 2014.

[S48] Badampudi, Deepika, Samuel A. Fricker, and Ana M. Moreno. "Perspectives on productivity and delays in large-scale agile projects.International Conference on Agile Software Development. Springer, Berlin, Heidelberg, 2013. .

[S49] Sekitoleko, Nelson, et al. "Technical dependency challenges in large-scale agile software development.International conference on agile software development. Springer, Cham, 2014.

[S50] Lee, Seiyoung, and Hwan-Seung Yong. "Distributed agile: project management in a global environment.Empirical Software Engineering 15.2 (2010): 204-217.

[S51] Wildt, Daniel, and Rafael Prikladnicki. "Transitioning from Distributed and Traditional to Distributed and Agile: An Experience Report.Agility Across Time and Space. Springer, Berlin, Heidelberg, 2010. 31-46.

[S52] Hossain, Syeda Sumbul. "Challenges and mitigation strategies in reusing requirements in large-scale distributed agile software development: a survey result.Intelligent omputing-Proceedings of the Computing Conference. Springer, Cham, 2019.

[S53] Lautert, Tatiane, Adolfo Gustavo Serra Seca Neto, and Nádia P. Kozievitch. Ä survey on agile practices and challenges of a global software development team.Brazilian Workshop on Agile Methods. Springer, Cham, 2019.

[S54] Kussmaul, Clifton. Önshore and offshore outsourcing with agility: Lessons learned." Agility Across Time and Space. Springer, Berlin, Heidelberg, 2010. 91-106.

[S55] Korhonen, Kirsi. "Migrating defect management from waterfall to agile software development in a large-scale multi-site organization: A case study.International Conference on Agile Processes and Extreme Programming in Software Engineering. Springer, Berlin, Heidelberg, 2009.

[S56] Hossain, Emam, Muhammad Ali Babar, and June Verner. "Towards a framework for using agile approaches in global software development.International Conference on Product-Focused Software Process Improvement. Springer, Berlin, Heidelberg, 2009.

[S57] Hole, Steinar, and Nils Brede Moe. Ä case study of coordination in distributed agile software development. European Conference on Software process improvement. Springer, Berlin, Heidelberg, 2008.

[S58] Välimäki, Antti, and Jukka Kääriäinen. "Patterns for distributed scrum—a case study.Enterprise interoperability III. Springer, London, 2008. 85-97.

[S59] Thomson, Esmeralda, and Richard Vidgen. "Balancing the Paradox of Formal and Social Governance in Distributed Agile Development Projects.Information Systems Development. Springer, New York, NY, 2013. 155-166.

[S60] Dorairaj, Siva, James Noble, and Petra Malik. Ünderstanding team dynamics in distributed agile software development.International conference on agile software development. Springer, Berlin, Heidelberg, 2012.

[S61] Lehtinen, Timo OA, et al. "Why the development outcome does not meet the product owners' expectations?." International Conference on Agile Software Development. Springer, Cham, 2015

[S62] Paasivaara, Maria, et al. SSupporting a large-scale lean and agile transformation by defining common values. International Conference on Agile Software Development. Springer, Cham, 2014.

[S63] Välimäki, Antti, Jukka Kääriäinen, and Kai Koskimies. "Global software development patterns for project management.European Conference on Software Process Improvement. Springer, Berlin, Heidelberg, 2009.

[S64] Paasivaara, Maria, Sandra Durasiewicz, and Casper Lassenius. Üsing scrum in a globally distributed project: a case study.Software Process: Improvement and Practice 13.6 (2008): 527-544.

[S65] Sablis, Aivars, Darja Smite, and Nils Moe. "Team-external coordination in large-scale software development projects.Journal of Software: Evolution and Process 33.3 (2021): e2297.

[S66] Beecham, Sarah, et al. "Do scaling agile frameworks address global software development risks? An empirical study. Journal of Systems and Software 171 (2021): 110823.

[S67] Agile management in the large: An exploratory case study onlarge-scale interdisciplinary projects

[S68] Vithana, V. N., David Asirvatham, and M. G. M. Johar. Än empirical study on using agile methods in global software development.2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2018.

[S69] Usman, Muhammad, et al. Ëffort estimation in large-scale software development: An industrial case study.Information and Software technology 99 (2018): 21-40.

[S70] Rolland, Knut, et al. "Problematizing agile in the large: alternative assumptions for large-scale agile development.39th International Conference on Information Systems. Association for Information Systems (AIS), 2016.

[S71] Scheerer, Alexander, and Thomas Kude. Ëxploring coordination in large-scale agile software development: A multiteam systems perspective." (2014).

[S72] Vallon, Raoul, et al. Ïdentifying Critical Areas for Improvement in Agile Multi-site Co-development.ENASE. 2013

[S73] Daneva, Maya, et al. Ägile requirements prioritization in large-scale outsourced system projects: An empirical study.Journal of systems and software 86.5 (2013): 1333-1353.

[S74] Dorairaj, Siva, James Noble, and George Allan. Ägile software development with distributed teams: Senior management support.2013 IEEE 8th International Conference on Global Software Engineering. IEEE, 2013.

[S75] Ralph, Paul, and Petr Shportun. SScrum Abandonment in Distributed Teams: A Revelatory Case.PACIS. 2013.

[S76] Noordeloos, Ramon, Christina Manteli, and Hans Van Vliet. "From RUP to Scrum in global software development: A case study.2012 IEEE Seventh International Conference on Global Software Engineering. IEEE, 2012.

[S77] Batra, Dinesh, Debra VanderMeer, and Kaushik Dutta. Ëxtending agile principles to larger, dynamic software projects: A theoretical assessment.Journal of Database Management (JDM) 22.4 (2011): 73-92.

[S78] Paasivaara, Maria, Sandra Durasiewicz, and Casper Lassenius. Üsing scrum in distributed agile development: A multiple case study.2009 Fourth IEEE International Conference on Global Software Engineering. IEEE, 2009.

[S79] Cho, Juyun. "Distributed Scrum for large-scale and mission-critical projects." (2007).

[S80] Korkala, Mikko, and Pekka Abrahamsson. "Communication in distributed agile development: A case study. 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007). IEEE, 2007.

[S81] Batra, Dinesh. "Modified agile practices for outsourced software projects.Communications of the ACM 52.9 (2009): 143-148