Methodological Support for Coordinating Tasks in Global Product Software Engineering

Carolus B. Widiyatmoko[1(✉)], Sietse J. Overbeek[2], and Sjaak Brinkkemper[2]

[1] PT Telkomunikasi Indonesia, Tbk., Bandung, Indonesia
Carolusbw@telkom.co.id

[2] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
{S.J.Overbeek,S.Brinkkemper}@uu.nl

Abstract. Distributing software processes by software producing organizations (SPOs) is emerging increasingly due to benefits that global software engineering (GSE) brings in terms of cost reduction, leveraging competencies, and market expansion. However, these organizations are facing communication and project control issues that can slow down the overall organization performance. Therefore, SPOs should be able to manage inter-dependencies among the tasks distributed to the globally dispersed teams. We analyze existing works and product software companies' best practices in coordinating tasks in GSE. This paper specifically focuses on constructing methodological support for task coordination that can be influenced by the situational factors at the companies. The support comprises a framework and a method developed by using a method engineering approach. We introduce the framework that depicts the aspects that should be examined by companies and the method that elaborates the practices to guide companies to coordinate tasks in GSE projects. The validation results show that the framework and the method are accepted by experts regarding completeness and applicability to help SPOs in managing coordination among globally distributed teams.

Keywords: Design science · Global software engineering · Method engineering · Software producing organization · Task coordination

1 Introduction

Software producing organizations (SPOs) are companies that focus on developing software as a product to be delivered to a targeted market [31]. The extensive client considerations and technical factors such as platform variability of prospective clients make these companies have more software engineering complexity than software companies who perform on software development projects for specific customers [37].

Some countries (India, China, and parts of eastern European countries to name a few) offer a large number of human resources such as software developers with lower salaries than developed countries such as in Europe and the

H. Panetto et al. (Eds.): OTM 2017 Conferences, Part I, LNCS 10573, pp. 213–231, 2017.
https://doi.org/10.1007/978-3-319-69462-7_14

US [1]. SPOs can get the economic value by utilizing these qualified human resources from other parts of the world. These companies apply what is called global software engineering (GSE) where parts of their engineering processes are conducted collaboratively in remote facilities located in other countries. By doing GSE, SPOs can also gain other benefits such as opportunities for market expansion and focus on product development competencies by leveraging some of the business functions to other organizational units [1,14].

In GSE, teams interact with each other working on tasks in the software engineering cycle and create an internationally distributed collaborative network. Apparently, there are differences in geographical location, time-zone, socio-cultural, and organization among these teams [15,21]. Problems then arise, when the teams begin to have difficulties in organizing tasks to manage dependencies on resources or processes caused by the distribution. To that end, SPOs need to be able to coordinate tasks to build better communication, synchronize work, and balance knowledge among these distributed teams [30].

The literature on GSE is rich as existing frameworks, guidelines, and GSE best practices can be identified [13,24,25]. However, most of the studies are not critically discussing coordination practices in globally distributed teams. It is believed that an approach cannot be easily implemented in the same way on every organization because each organization has different situational factors [4]. To this end, this paper surveys the state of the art of task coordination approaches and proposes methodological support that presents the abstraction of coordination methods for SPOs that perform software engineering globally.

In this paper, our goal is to identify the coordination studies and practices and to present the elements related to task coordination that SPOs must understand and be aware of in the context of GSE. This paper is organized as follows: Sect. 2 illustrates our research methods where the execution will be elaborated in subsequent sections. Section 3 describes the literature review to build the foundation of knowledge in this topic. Next to that, Sect. 4 reports how the research artifacts are constructed and then assessed to ensure that the method meets the expectation of those who need this method. After that, Sect. 5 recounts the research process and provides a critical reflection on the benefits and limitations of the method. Finally, Sect. 6 concludes the paper by discussing our findings and limitations.

2 Research Method: The Design Cycle

This research is conducted by adopting an iterative problem-solving design science method [36]. Design science cycle is a subset of the engineering cycle which is a continuing investigation includes design processes to solve a problem by creating an artifact. The design science approach comprises three main stages which are: Problem investigation, solution development, and solution validation. The problem investigation phase and the solution development phase are executed by using method association technique, and the solution validation is accomplished through expert opinion. The stages are elaborated in the following section.

2.1 The Method Association Approach

For the analysis of the existing frameworks, techniques, and methods in global software engineering, we use the method engineering approach [4]. Where software engineering pays attention to all aspects pertains to software production, method engineering focuses on the construction of method that fall into the software engineering domain. Method engineering is defined as "the engineering discipline to design, construct and adapt methods, techniques, and tools" [4]. Hence, this stage will represent the investigation and the solution development of the design cycle that we follow.

In this section, we present our approach to constructing a reference method for task coordination by adopting the method association approach (MAA) [19]. The MAA is used to create meta-models for the processes and data perspectives constructed from the established methods gathered by studying the state of the art of task coordination from literature and best-practices obtained from companies' experiences. Hence, we choose process-delivery diagram (PDD) [35] as the meta-modeling technique to present the methodological support in a uniform and formal representation. The MAA approach adopted from [19] in this research consists of eight steps as depicted in Fig. 1.

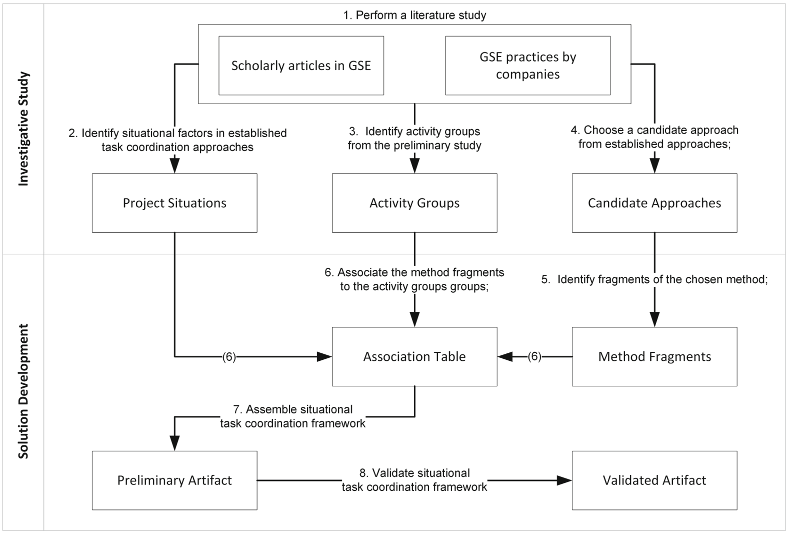

Fig. 1. The MAA design cycle, adapted from [19]

2.2 Method Evaluation Approach

The validation sessions were performed by involving researchers and practitioners to get feedback from a broader perspective. The proposed artifact will be assessed by these experts who reflect how such an artifact will interact with problem contexts and then predict what effects that they think the artifact would have by using expert opinion [36]. Validation of the proposed method by expert opinion will work if the experts understand the artifacts which enables them to imagine problem contexts and predict the effects of the artifacts in the contexts.

The validation strategy is following the Framework for Evaluating in Design Science [32]. We selected the Human Risk and Effectiveness strategy as the proposed artifacts are user oriented that should be evaluated with real users in their real context. Formative assessment starts the evaluation and progressively engages more summative assessment focusing on the applicability of the artifacts. Therefore, we use two-steps validation by involving experts that have scientific backgrounds and experts from business practitioners. The scientific experts will do the criteria-based assessment on the designed method based on meta-modeling criteria which are completeness, consistency, efficiency, reliability, and applicability [5]. Business practitioners will concentrate on usefulness and ease of use of the designed method [16,26].

3 Problem Investigation

3.1 Literature Study

To start the problem investigation phase as is show in Fig. 1, we performed a study by reviewing previous scientific articles in GSE and conducting interviews with several SPOs in the Netherlands to expose the state of the art of task coordination in GSE practices. The literature review is done by reviewing the paper's abstraction, and if it is necessary, we also get deeper by examining contents of the article and do forward and backward snowballing to find more information related to the concepts discussed in the main article [34]. A total of 155 papers were involved in this literature study, and in the end, we found several concepts related to task coordination in GSE as presented in Fig. 2 (The presented semantic network is simplified based on the global software engineering context). Through this semantic network, we continue our study on the concepts directly-related to coordination which are: communication, control, knowledge, tool, and stakeholders. Then, we focus our research on the literature that addresses the practices related to those concepts.

After that, as part of the MAA's first step as well as to understand the situational background of coordination practices as the MAA's second step, we studied the coordination practices from the companies. There are four SPOs which were participating in our research. There are two respondents from each of the companies from various job positions who have experiences in global software engineering. The company's names are replaced with AlphaSoft, BetaSoft, GammaSoft, and DeltaSoft for the reason of confidentiality. The interviews were performed from December 2016 until February 2017. During the interviews, the concepts found in the literature study and the company's practices in coordinating interdependencies during performing GSE were discussed. To this end, the interviewer posed several questions such as: "*What kind of challenges does the*

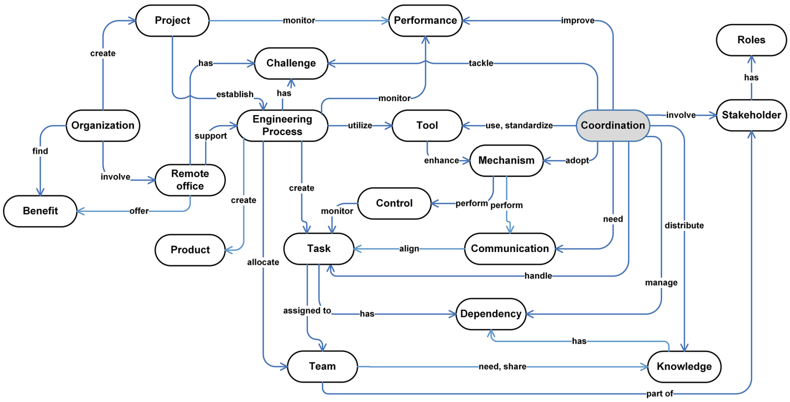

Fig. 2. Semantic net of task coordination at SPO in GSE environment

organization have by doing GSE?" and *"What kind of instruments do you use to support you in managing tasks?"*

The interviews captured several topics: company background, job roles and functions, partners or remote offices profiles, product profiles, company's vision in

Table 1. Task coordination practices comparison

Aspects	AlphaSoft	BetaSoft	GammaSoft	DeltaSoft
Remote office (RO) location	Belgium, Romania, India (susp.)	Malaysia	Mainly in Poland and India	Romania
RO ages	>6 years (Romania)	17 years	>2 years	±2.5 years
RO functions	Development, testing	System design, development, testing	System design, development, testing	Development
Team size	±40	>100	>100	6
Engineering method	Scrum of Scrum	SAFE[a]	Traditional	Scrum[a]
Main market	Dutch companies	Global	Global and internal	Global and internal
Challenges	Communication, trust, timezone difference	Communication tools quality	Communication, expertise, time-zone, org. silos, culture	Lack of explicit knowledge
Communication	Direct	Direct (technical area), indirect (enterprise)	Indirect	Direct
Control	Proactive, mutual adjustment	Reactive, mutual adjustment	Reactive, direct supervision	Proactive, standardization
Knowledge sharing	Document sharing, site visit	Formal training, document sharing	Mentoring (socialization), documentation	Socialization
Tool	Vicon, site visit, skype, TFS	Webex, Skype, Sharepoint	Webex, Skype, OneVision, Sharepoint	Regular site visit, Slack, Skype
Important roles	Scrum master, unit manager	Product manager, feature owner, dev manager	Service coordinator	Team leader

[a]self-build, customized, or similar approach

GSE, challenges in performing GSE, approaches in GSE practices, and stakeholders involved in GSE projects. Each interview was conducted between 45–60 min. The results of the interviews are summarized in Table 1.

3.2 The Global Task Coordination Framework

Before starting to assembly the method, we introduce the Global Task Coordination framework showing that coordinating tasks in GSE comprises three main blocks that build practices to achieve coordinated output: the task coordination mechanisms, the coordination mechanisms supports, and the organization's situational factors that can influence the appropriate coordination mechanisms of the SPOs (Fig. 3). We define coordinated output as a situation where an organization is able to manage task ownerships, synchronize jobs and hand-over to achieve an integrated results among its distributed teams.

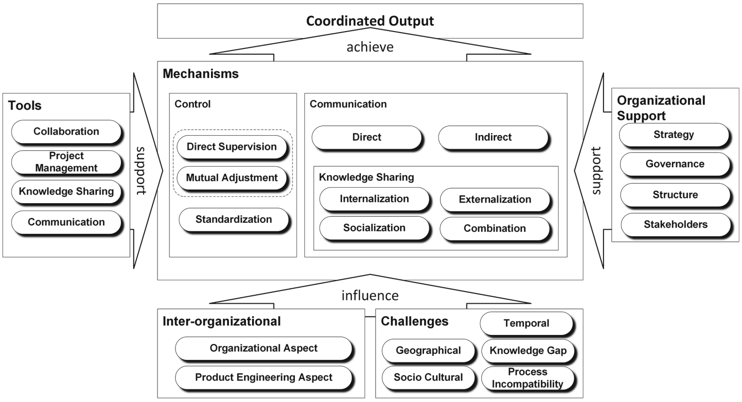

Fig. 3. The global task coordination framework

Based on this framework, SPOs should consider three main elements of task coordination in GSE as follows:

Coordination Mechanism. Managers or leaders in SPOs should be aware of two coordination mechanisms which are the control mechanism and the communication mechanism. Stakeholders who hold the management function have a common function: to manage the dependencies by synchronizing the activities that bring all the team members together at the same time and place for some pre-arranged purpose [9,20]. There are different types of mechanisms perceived from coordination mechanisms by Mintzberg [20]. Distributed teams that can organize dependencies by themselves, managers may perform a mutual adjustment mechanism by helping the teams in managing the work without getting involved directly in the decision making in problem-solving and task distribution. Meanwhile, in some organizations, managers still should supervise directly and take the lead in controlling the task and problem management. Beyond that two practices, the standardization of work and deliverable are the notable practices to ensure the teams have a standard guideline to perform their jobs and the task transition can be achieved smoothly.

Next, regarding the communication mechanism, we suggest SPOs to encourage direct communication among the distributed teams as much as possible by adjusting the working time or providing communication supports (e.g. tools, protocols) to build strong teamness. However, because of the GSE challenges, direct communication will never be enough. Indirect communication in a large company can be in a hierarchical form following the information flow mechanism within the organization top to bottom and vice versa. It is necessary to break the network into smaller groups to facilitate communication [8].

Communication in software engineering is seen as a knowledge-intensive activity [3]. Without an effective knowledge sharing, the project can suffer due to the failure of coordination problems that encourage collaboration [14]. Companies need to recognize the cognitive level of team members to know what kind of knowledge is needed by the team members. Our study identified several mechanisms in disseminating or distributing knowledge based on the knowledge transformation categorization, which are: Socialization, externalization, combination, and internalization [22]. SPOs can create a mapping between the available knowledge and the knowledge required to provide a knowledge gap analysis to determine the need of tools, technologies, or methods can help the transfer of knowledge [18].

Coordination Support. From the literature review and interviews, we found that SPOs aware the importance of coordination in dispersed software engineering teams. Therefore, SPOs facilitate the coordination practices by providing organizational infrastructure and tools. There are organizational supports that can be recognized such as strategy [28,29], governance [2,12], organization structure [29], and clear roles and job functions [28]. To promote task coordination, SPOs should support the practices with tools. The purposes of the tools can vary, such as providing collaboration space, enabling direct communication, amplifying the distribution of knowledge, and enhancing project control [6,27].

Tools can be utilized to provide an ongoing project activities overview at different levels of detail, to support communication and knowledge transfer, and to bring improvements to shared spaces in an integrated development environment [7]. The divergence in tools, the inadequateness of the supporting system, and the imbalance level of expertise in using the tools can limit collaboration and impede communication which ultimately delays the project [15].

Coordination Situational Background. We identified that approaching coordination can be affected by the internal organizational factors and the challenges faced by SPOs. There are several internal factors of the organization that can affect to how the organization prepares and manage task interdependencies. Organization distribution specifies how large the dispersed teams are, how legal relationships between scattered organizations are, and how organizations divide the engineering work. Process distribution describes the relationship among the tasks, the proportion of overlapped tasks, and the process chain between the distributed teams. Dependency shows how the artifacts are shared or transferred among the distributed teams.

Meanwhile, GSE challenges also provide variability in determining appropriate coordination practices. Problems emerge from the incompatibility of processes, tools, and issues related to collaboration bottlenecks because the teams that do not stay in one place are expected to impact job settings and dependencies. The geographical distance shows how teams are distributed in different locations spatially that restrict the organization to have direct communication. Temporal challenges and socio-cultural problems frequently become the communication barriers that inhibit the achievement of mutual understanding.

3.3 Identifying Activity Groups

The third step in the MAA is determining activity groups. An activity group can be seen as a class of similar tasks that represent particular functions or requirements. A chain of activity groups will describe the flow of the method, integrate the involved concepts, and elaborate the detail steps in each of the activity. By using the literature that was used during the preliminary study, we identified all the activities and concepts from the articles on the literature study and practices by the participating companies in the interviews. The study resulted in 46 unique activities and 33 concepts related to the activities. Thereafter, by analyzing the logic and categorizing the associated activities, we circumscribed the activities into 5 activity groups which are: (1) Business Analysis, (2) Situational Factor Analysis, (3) Support Analysis, (4) Task Coordination Mechanisms, and (5) Finalization and Improvement. These activity groups will be used to build the association table to map the candidate activities and concepts as the fragments of the proposed method.

3.4 Identifying Method Fragments

To facilitate the development of the association tables, we reduce the number of articles involved by selecting six articles to represent other articles (The MAA Step 4). These selected articles are chosen subjectively based on several criteria such as the number of citations and depth of discussion related to coordination practices. In the end, we tied our investigation on six papers representing the other articles: [11,17,23,28,29]. The combination of these papers covers the concepts found in the study literature as illustrated in Fig. 2.

The following step of the MAA (Step 5), we identified all the fragments found in the literature and interviews before mapping the fragments to the activity groups [19]. To avoid ambiguity, manage fragments' granularity, and improve clarity, we standardized new terms for the fragment names. For example, "Assign a liaison officer" and "Assign a service coordinator" have two different concepts namely "liaison officer" and "service coordinator." However, these concepts can be understood as a single concept: "On-site Coordinator." Another activity might consist of two activities, such as "Collaboratively develop, communicate, and distribute work plan" should be split into "Develop work plan" and "Distribute work plan."

4 Solution Design: Construction of a Global Task Coordination Method

4.1 Method Association

To start the solution development phase, the sixth step of the MAA requires that each activity and concept found in the preliminary studies are mapped to the activity groups in the association tables as depicted in Fig. 4.

Group	Activities	L1	L2	L3	L4	L5	L6	L7	CA	CB	CC	CD
S1	Product management planning		x							x		
	Reward achievement			x								
	Determine task allocation					x						
	Set goals						x		x	x	x	x
	Determine task schedule						x					
	Define collaboration strategy						x					
	Establish process collaboration						x					
	Identify organization V&M									x		
	Establish SE method						x					
S2	Identify socio-cultural profiles					x	x		x	x		x
	Identify required knowledge					x						
	Identify challenges					x			x	x	x	x
	Identify situational factors									x	x	
S3	Determine organization structure			x		x						
	Role assignment			x		x				x	x	
	Assign onsite coordinator [illegible]											
	Di[illegible]						x					
	[illegible] communication norm									x	x	
	Define work product quality standardization									x	x	
	Standardize process									x	x	x
S5	Organize process improvement						x			x	x	
	Continuous improvement									x		

(a) Activities association table

CONCEPTS	L1	L2	L3	L4	L5	L6	L7	CA	CB	CC	CD
ROLE			x						x		
CHALLENGE	x										
FORMAL TRAINING									x		
KNOWLEDGE BASE				x					x	X	
INDIRECT COMMUNICATION							x				
COMMUNICATION PROTOCOL	x										
SE STRATEGY									x		
WORK PRODUCT											
CORPORATE STRATEGY								x	x	x	x
TEAM								x	x	x	x
SCRUM MASTER, FACILITATOR		x						x			
SOCIO-CULTURAL	x							x	x		
MANAGEMENT BOARD									x		
ORGANIZATIONAL SUPPORT			x								
TECHNOLOGY BOARD									x		

(b) Concepts association table

Fig. 4. Association tables for the Global Task Coordination Method

There are some codes used in Fig. 4. The codes are described as follows:

- L1..L6: Literature selected to represent other literature, which are [11,17,23,25,28,29];
- C1..C4: The codification for the participating companies, C1: AlphaSoft, C2: Betasoft; C3: GammaSoft; C4: DeltaSoft; and
- S1..S5: The five activity groups.

4.2 Method Assembly

To assemble the designed method (step 7), we use PDD language to present the method. That supports assembly-based method engineering approach for constructing situational analysis and design methods [35]. The designed method is derived based on the activity groups as listed in Sect. 3.3 and consists of the steps digested from the association tables. In the end, we present the Global Task Coordination Method. Due to page limitations, we will only present the final validated versions as depicted in Fig. 5. Additional activities are also added based on our subjectivity to maintain the logical order and flow of the activities within the method.

As the method needs to elaborate the coordination practices in a more detail, the method contains three open activities, which are "*Perform routine activities*", "*Determine appropriate control mechanism*", and "*Determine appropriate communication mechanism*". These open activities are presented in the Appendix A.

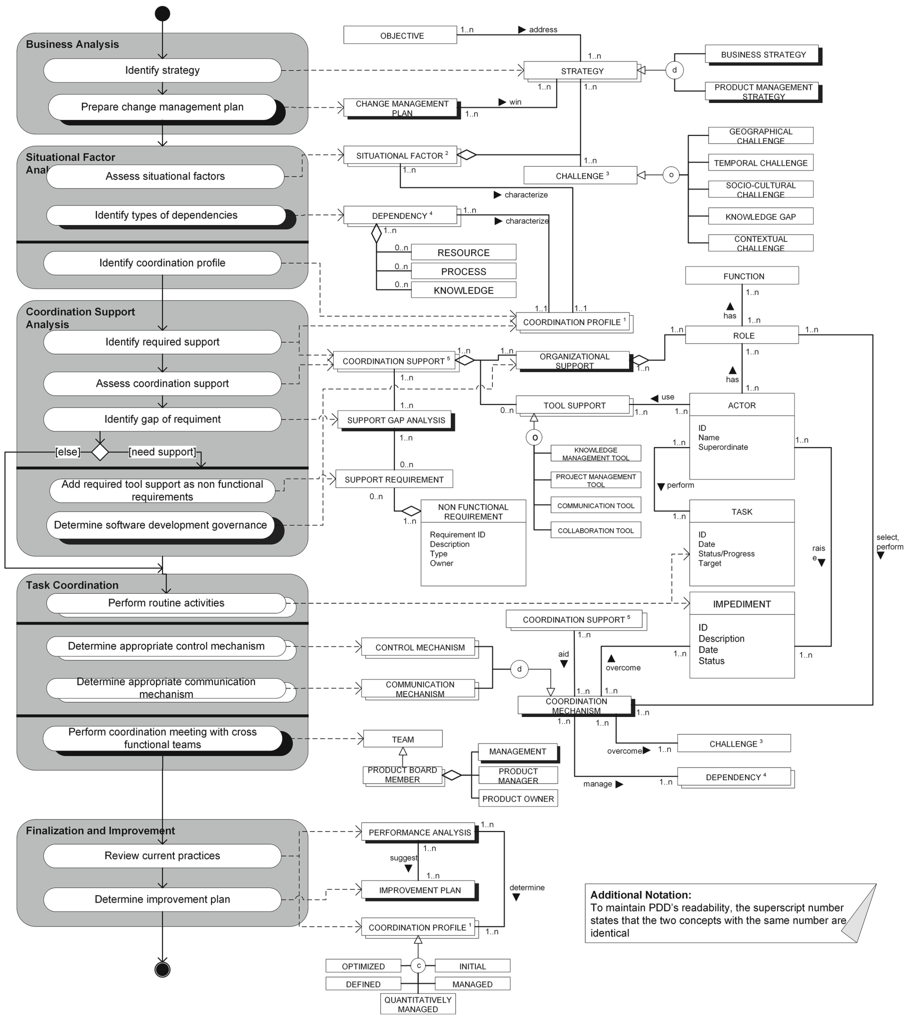

Fig. 5. The Global Task Coordination Method: Main level

4.3 Method Validation

The participating experts consist of scientific experts and business practitioners. The rationale for inviting the researcher is to obtain his feedback and critics from a person who has a broader perspective in global software engineering domain from the scientific standpoint. Other experts would be expected to provide their feedback and critics from their daily practices to assess the usability of the artifacts. The first evaluation adopts the criteria-based approach. We consider evaluating the model based on a set of criteria in assessing a method designed by the method assembly approach, which are: Completeness, consistency, efficiency, reliability, and applicability [5].

For the remaining evaluations, we involve real users to assess our designed artifacts with a natural setting that offers more critical face validity and also assures more rigorous assessment of the acceptance of the artifact [32]. We adopt two constructs from Technology Acceptance Model (TAM) which are Perceived Usefulness and Perceived Ease of Use [10]. TAM usually is used to test the behavioural acceptance or intention of using information technology such an application framework [26], software process engineering tools [33], and a newly designed method in software engineering [16]. Perceived usefulness is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance." Meanwhile, perceived ease of use refers to "the degree to which a person believes that using a particular system would be free of effort" [10, p. 320]. As method engineering is used in the engineering of methods and tools in information system and technology domain [4], we assume that the adoption of TAM will be useful to evaluate the designed artifacts. The results of the validation are summmarized as presented in Table 2.

Table 2. Evaluation session summary

Criteria	Session					Remarks
	#1	#2	#3	#4	#5	
Completeness	±	+	+	+	+	Some details should be added (e.g. socio-cultural level, stakeholders, change management, governance)
Consistency	−	n/a	+	−	+	Overlapped concepts
Efficiency	−	±	+	−	+	Difficult for non-technical users. The method application does not need huge effort
Reliability	−	−	+	−	+	Several unclear terminologies
Applicability	n/a	+	+	−	+	Easy to be followed
Usefulness	n/a	+	+	n/a	+	Covers both theoretical and practical, broad aspects
Ease of use	n/a	+	+	n/a	+	The high-level PDD is useful for higher-level users
Intention to use	n/a	+	+	n/a	+	Practitioners intent to use in different ways

+: satisfied; −: unsatisfied; ±: partly satisfied; n/a: no feedback related to the criteria

Criteria-Based Evaluation. This sub section describes the results of the criteria-based evaluation that consists of five criteria of method engineering [5], which are listed below:

Completeness. The participants were satisfied with the framework and the method. The practitioners indicated that the method covers the real practices and describes the roles who are responsible for the specific activities in managing the distributed work and team members. At the same time, the framework provides a holistic overview of the theoretical perspective of a task coordination approach for PSOs to assess their situational background and the required support, as expressed by the Service Delivery Manager from GammaSoft, "*I like the overview that you have that really helps me. It's more than just theoretical. I've learned a lot.*"

Consistency. The attempt to provide a guideline at more detailed levels threaten the coherence of the developed method. The first-round evaluation directly criticized the consistency issue related to the relationship between communication and knowledge sharing in the domain of software engineering. In the subsequent rounds of assessment, participants found that the concepts and the activities are autonomous and mutually consistent.

Efficiency. The scientific experts argued that the method will not be easy to be followed by non-technical users due to the complexity and granularity. Indeed, as noticed by the practitioners, the artifacts cover all task coordination aspects in global software engineering because the artifacts attempt to cover broad topics. It is a challenge to provide a solution that comprises broad issues, while on the other hand the solution should present a clear explanation and applicative guideline.

Reliability. During the evaluation sessions, some disagreements and suggestions of the terminologies were conveyed by the participants. In the first session, the expert suggests using more specific and general terminologies to avoid misperception and uncertainty, while in the second session the expert suggested that the control mechanism should be elaborated. Then, we find it difficult to keep the method compact. After the fourth session, based on the suggestion from the expert we modified the model and optimized the documentation to make the method more concise. It is easier to maintain the reliability and consistency of concepts and activities presented in the method.

Applicability. The practitioners are satisfied with the designed artifacts and indicated that both the method and the framework could be applied as a reference guideline in their daily practices. The following sub section discusses the applicability from the perspective of behavioral intention to use by discussing the perceived usefulness and the perceived ease of use of the artifacts.

Perceived Usefulness and Perceived Ease of Use Evaluation. The practitioners as the participants of the expert validation sessions indicated to have an intention to use the GSE task coordination method. The participants from

BetaSoft were enthusiastic and considered the usefulness and ease of use of the method even though they have been doing global software engineering for more than ten years. The Product Manager from BetaSoft conveyed to augment the Technology Director comments by reflecting their past experiences, "*The model is useful and we recognize a lot of things... There's part of the method that can help us in different ways (of coordination). We also think that it's easy to use because we already used to it. We still can use the guidelines.*" Meanwhile, the participant from AlphaSoft indicated that the framework could be useful for those who have higher management roles and the detailed guidelines will be helpful for line managers and team leaders. We noticed that the experts preferred to see the method as a set of best practices guidelines where they can come back anytime, assess their current situation to detect the coordination deficiencies while enhancing their coordination practices. The practitioners could see the benefits of the method. They notified that they are very pleased with the method and desire to use the method in their daily practices.

5 Discussion

Our investigative study was completed by a literature study and interviews with SPOs so that we got a full picture from a scientific side as well as a practical side. Similarly, by performing iterative validation in which each update of each validation session becomes the input for the next session, we can assure that the method presented after the last validation is the complete and applicable artifact. The benefits and drawbacks of the method were obtained by evaluating the method based on the FEDS approach. We compare the results with the meta-modeling criteria and the intention to use test.

We incorporated experts' feedback in newer method versions. We observed a number of benefits by the assessment of the method. First, the GTC framework offers the abstraction of coordination elements that can be used as a reference that can help managers in PSOs to analyze the extent to which the organization has been able to prepare for coordination. Second, the method can be used to increase awareness of stakeholders to find out who are involved and when these roles perform particular coordination activities. Third, the method is perceived to improve the effectiveness of communication and project management by showing the variety of problems, best practices that can be emulated, and the resources needed to solve the problem.

This research aims to present the coordinating elements in the GSE comprehensively as well as its best practices to be a reference to all levels of stakeholders in SPOs. The attempt to fulfill both the conceptual as well as the detail presentation becomes an obstacle in maintaining the level of granularity that becomes the main drawback of this method which is suspected can lead to rejection of the application of this method. Although we conduct investigations and validations with rich GSE experienced business-practitioners, there is no evidence to present the applicability of this method in the real situation. However, with a positive response from experts, we perceive that this method is acceptable and has a great opportunity to be applied as a reference in coordinating on the GSE by SPOs.

In order to judge the quality of this research, we used multiple data sources to guarantee the construct validity by using multiple data sources through literature study and investigative interviews. However, expert opinion focuses more on the desire to use the method. This research could not provide an evidence where the method can effectively improve the performance of SPOs in quantitative results. The interviews involve different stakeholders with different perspectives from different companies with different characteristics. We also performed the validation phase by involving both scientific experts and business experts to ensure that the method is built comprehensively examined and gained objective judgments not only from a single point of view to maintain the external validity. Nonetheless, it may be possible that another investigation phase and validation phase at another organization outside the Netherlands yields different results. Last but not least, the limitation regarding the reliability is that the results are heavily dependent on the experience of the experts, which possibly will raise a threat to the reliability of this research.

6 Conclusions and Future Research

Software producing companies involve complex factors in their software engineering processes. As the complexity increases in situations where engineering processes are carried out in a globally distributed environment, the need to coordinate tasks will be influenced by the differentiating factors that make coordination practices unique for each organization. Therefore, we present the Global Task Coordination Method as a guideline for SPOs to determine the appropriate coordination practices based on their situational backgrounds. The method provides a reference for a better understanding of the existing aspects related to task coordination among distributed teams and to suggest adequate proposals to identify and analyze the various alternatives in managing interdependencies distributed teams. During the process, our research develops a comprehensive understanding of existing knowledge base of task coordination methods by elaborating and connecting methods which have been studied and approaches by SPOs on how tasks are allocated in GSE projects. In addition, the methodological support enhance the theoretical base in the software engineering domain by adding sources of knowledge in task coordination regarding project planning and execution management through the MAA as a method engineering approach.

Our proposal is developed based on industry inputs which are headquartered in Netherlands. We tried to maintain the external validity by selecting companies with a different characteristic of global distribution. Nonetheless, it may be possible that another investigation phase and validation phase at another organization outside the Netherlands yields different results.

To further help practitioners, we intend to encourage researchers to conduct a longitudinal multi-case study on the practices of companies. To that way, we will able to provide more objective evidence on related challenges and successful practices when coordinating tasks among distributed teams globally as well as measuring the application of the method quantitatively.

Appendix A The Global Task Coordination Method

See Figs. 6, 7 and 8.

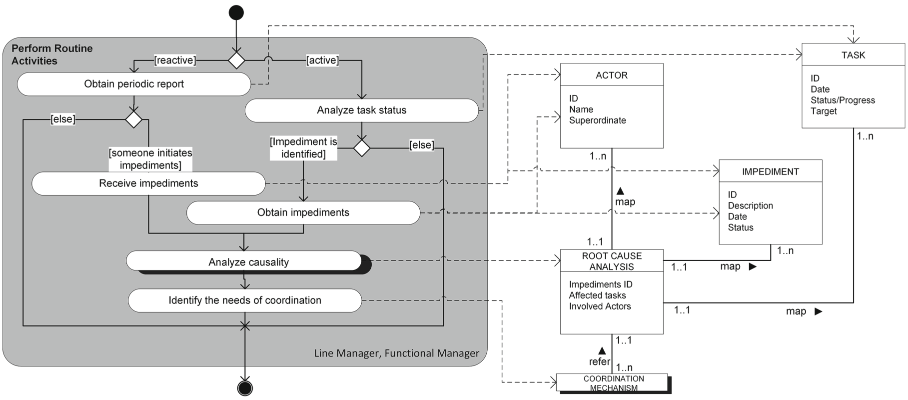

Fig. 6. The Global Task Coordination Method: Daily routines

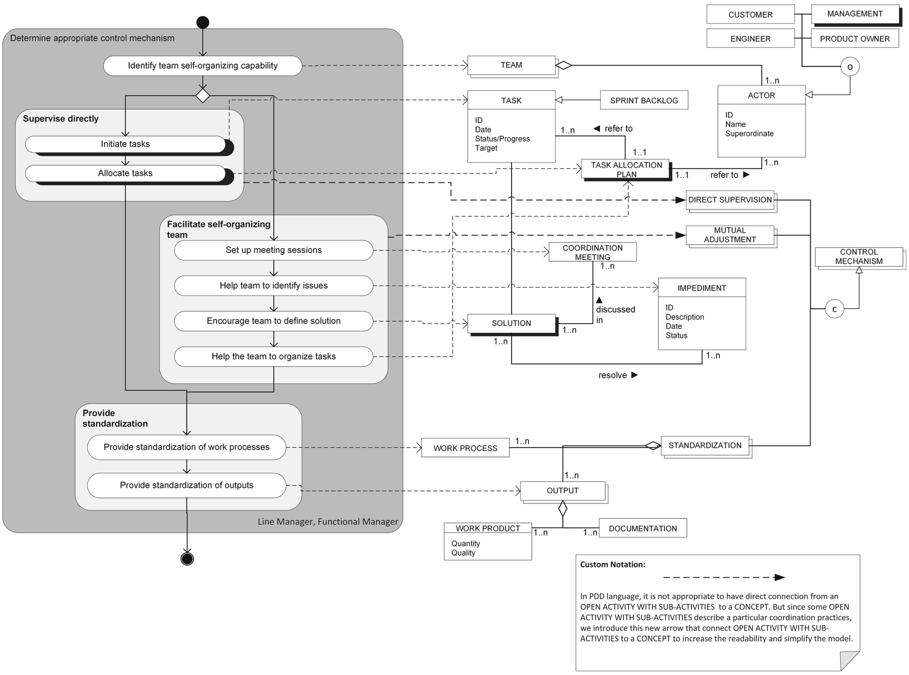

Fig. 7. The Global Task Coordination Method: Control Mechanism

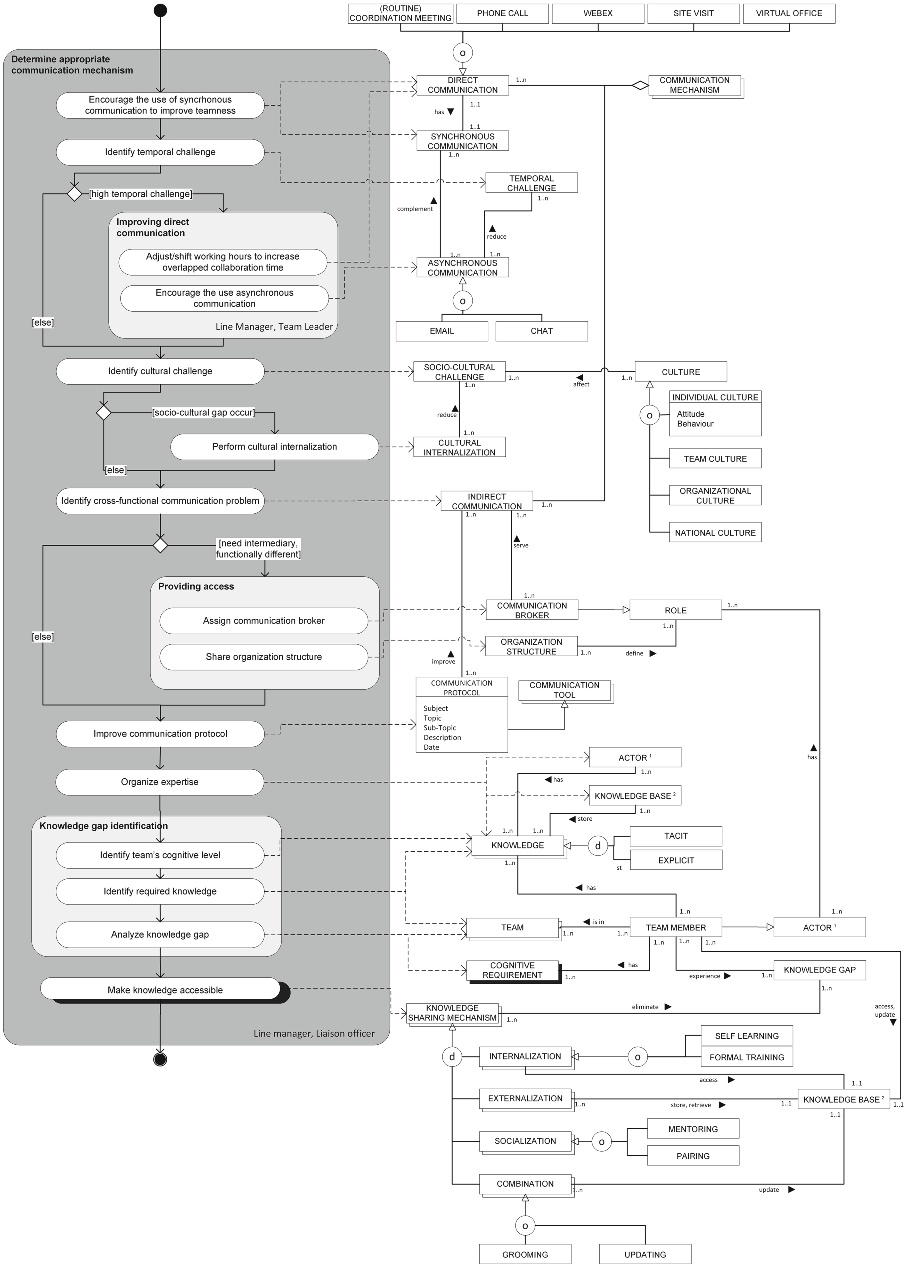

Fig. 8. The Global Task Coordination Method: Communication Mechanism

References

1. Ågerfalk, P.J., Fitzgerald, B., Olsson, H.H., Conchúir, E.O.: Benefits of global software development: the known and unknown. In: Wang, Q., Pfahl, D., Raffo, D.M. (eds.) ICSP 2008. LNCS, vol. 5007, pp. 1–9. Springer, Heidelberg (2008). doi:10.1007/978-3-540-79588-9_1
2. Bannerman, P.L.: Software development governance: a meta-management perspective. In: Proceedings of the 2009 ICSE Workshop on Software Development Governance, pp. 3–8. IEEE Computer Society (2009)
3. Bjørnson, F.O., Dingsøyr, T.: Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used. Inf. Softw. Technol. **50**(11), 1055–1068 (2008)
4. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Inf. Softw. Technol. **38**(4), 275–280 (1996)
5. Brinkkemper, S., Saeki, M., Harmsen, F.: Meta-modelling based assembly techniques for situational method engineering. Inf. Syst. **24**(3), 209–228 (1999)
6. Cataldo, M., Wagstrom, P.A., Herbsleb, J.D., Carley, K.M.: Identification of coordination requirements. In: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work - CSCW 2006, p. 353. ACM Press, New York (2006)
7. Chadli, S.Y., Idri, A., Ros, J.N., Fernández-Alemán, J.L., de Gea, J.M.C., Toval, A.: Software project management tools in global software development: a systematic mapping study. SpringerPlus **5**(1), 2006 (2016)
8. Chiu, M.L.: An organizational view of design communication in design collaboration. Des. Stud. **23**(2), 187–210 (2002)
9. Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Garcia-Penalvo, F.J., Tovar, E.: Project managers in global software development teams: a study of the effects on productivity and performance. Softw. Qual. J. **22**(1), 3–19 (2014)
10. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q. **13**(3), 319 (1989)
11. Deshpande, S., Beecham, S., Richardson, I.: Global software development coordination strategies - a vendor perspective. In: Kotlarsky, J., Willcocks, L.P., Oshri, I. (eds.) Global Sourcing 2011. LNBIP, vol. 91, pp. 153–174. Springer, Heidelberg (2011). doi:10.1007/978-3-642-24815-3_9
12. Ebert, C.: Managing global software projects. In: Ruhe, G., Wohlin, C. (eds.) Software Project Management in a Changing World. LNCS, pp. 223–246. Springer, Heidelberg (2014). doi:10.1007/978-3-642-55035-5_9
13. Espinosa, J.A., Lerch, J., Kraut, R.: Explicit vs. implicit coordination mechanisms and task dependencies: one size does not fit all. Technical report, Carnegie Mellon University, Pennsylvania, USA (2002)
14. Herbsleb, J., Moitra, D.: Global software development. IEEE Softw. **18**(2), 16–20 (2001)
15. Jain, R., Suman, U.: A systematic literature review on global software development life cycle. ACM SIGSOFT SEN **40**(2), 1–14 (2015)
16. Koc, H., Timm, F., Espana, S., Gonzalez, T., Sandkuhl, K.: A Method for Context Modelling in Capability Management. Research Papers 43, January 2016
17. Kotlarsky, J., van Fenema, P.C., Willcocks, L.P.: Developing a knowledge-based perspective on coordination: the case of global software projects. Inf. Manage. **45**(2), 96–108 (2008)

18. Kristjánsson, B., Helms, R., Brinkkemper, S.: Integration by communication: knowledge exchange in global outsourcing of product software development. Expert Syst. **31**(3), 267–281 (2014)
19. Luinenburg, L., Jansen, S., Souer, J., van de Weerd, I., Brinkkemper, S.: Designing web content management systems using the method association approach. In: Proceedings of the 4th International Workshop on Model-Driven Web Engineering, MDWE 2008, pp. 106–120 (2008)
20. Mintzberg, H.: The Structuring of Organizations. Pearson, Upper Saddle River (1979)
21. Niazi, M., Mahmood, S., Alshayeb, M., Riaz, M.R., Faisal, K., Cerpa, N., Khan, S.U., Richardson, I.: Challenges of project management in global software development: a client-vendor analysis. Inf. Softw. Technol. **80**, 1–19 (2016)
22. Nonaka, I., Takeuchi, H.: The knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York (1995)
23. Olsson, H.H., Conchúir, E.Ó., Ågerfalk, P.J., Fitzgerald, B.: Global software development challenges: a case study on temporal, geographical and socio-cultural distance. In: Proceedings of the IEEE International Conference on Global Software Engineering, ICGSE 2006, pp. 3–11. IEEE, Florianopolis, October 2006
24. Olsson, H.H., Fitzgerald, B., Ågerfalk, P.J., Conchúir, E.Ó.: Agile practices reduce distance in global software development. Inf. Syst. Manage. **23**(3), 7–18 (2006)
25. Paasivaara, M., Lassenius, C.: Could global software development benefit from agile methods? ICGSE 2006, pp. 109–113. IEEE, Florianopolis (2006)
26. Polančič, G., Heričko, M., Rozman, I.: An empirical examination of application frameworks success based on technology acceptance model. J. Syst. Softw. **83**(4), 574–584 (2010)
27. Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M., Beecham, S.: Tools used in global software engineering: a systematic mapping review. Inf. Softw. Technol. **54**(7), 663–685 (2012)
28. Richardson, I., Casey, V., Burton, J., McCaffery, F.: Global software engineering: a software process approach. In: Mistrík, I., Grundy, J., Hoek, A., Whitehead, J. (eds.) Collaborative Software Engineering, pp. 35–56. Springer, Heidelberg (2010)
29. Smirnova, I., Münch, J., Stupperich, M.: A canvas for establishing global software development collaborations. In: Dregvaite, G., Damasevicius, R. (eds.) ICIST 2014. CCIS, vol. 465, pp. 73–93. Springer, Cham (2014). doi:10.1007/978-3-319-11958-8_7
30. Smite, D.: Global software development improvement. Ph.D. thesis, University of Latvia (2007)
31. Vähäniitty, J.: Do small software companies need portfolio management? Ph.D. thesis, Helsinki University of Technology (2006)
32. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: A Framework for Evaluation in Design Science Research. Eur. J. Inf. Syst. **25**(1), 77–89 (2016)
33. Wagenaar, G., Overbeek, S., Helms, R.: Describing criteria for selecting a scrum tool using the technology acceptance model. In: Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B. (eds.) ACIIDS 2017. LNCS (LNAI), vol. 10192, pp. 811–821. Springer, Cham (2017). doi:10.1007/978-3-319-54430-4_77
34. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future writing a literature review. MIS Q. **26**(2), xiii–xxiii (2002)

35. van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, pp. 38–58 (2009)
36. Wieringa, R.J.: Design Science Methodology: For Information Systems and Software Engineering. Springer, Heidelberg (2014)
37. Xu, L., Brinkkemper, S.: Concepts of product software. Eur. J. Inf. Syst. **16**(5), 531–541 (2007)