On the Nature of GSE Organizational Social Structures: an Empirical Study

Damian A. Tamburri, *Dept. of Computer Science, VU University, Amsterdam, Netherlands*, *d.a.tamburri@vu.nl*

Patricia Lago, *Dept. of Computer Science, VU University, Amsterdam, Netherlands*, *p.lago@vu.nl*

Hans Van Vliet, *Dept. of Computer Science, VU University, Amsterdam, Netherlands*, *hans@vu.nl*

Elisabetta di Nitto, *Dip. di Elettronica ed Informazione, Politecnico di Milano, Milano, Italy*, *dinitto@polimi.it*

***Abstract*—In Global Software Engineering (GSE), people are organized in teams, distanced in space, time and culture. Organizational research calls this interplay of people an Organizational Social Structure (OSS). Previous literature in GSE shows that its OSS is highly dynamic and unpredictable. This paper presents a mapping of OSS types onto GSE organizational factors, based on empirical evidence. We made two observations: first, current OSS types don't support factors related to GSE process management and organizational efficiency (e.g. risk management, language, etc.). Second, OSSs in GSE have attributes which don't map onto any GSE factor, rather introduce a new one, awareness management (e.g. awareness of skills to others, awareness of tasks, tasks (re-)localization, etc.). Our conclusions are twofold. First, OSSs for GSE should focus on increasing support to process management and organizational efficiency. Lastly, research in GSE should include factors focusing on awareness management.**

***Keywords*-Global Software Development, Social Computing, Social Structures, Requirements Engineering, Human Factors, Empirical Study**

I. Introduction

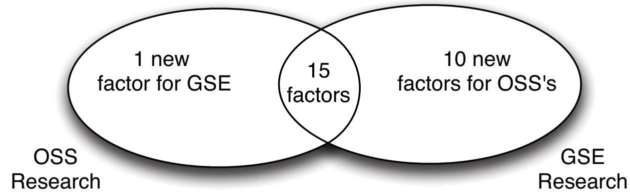
Global Software Engineering (GSE) is a business decision entailing project teams to collaborate globally on the same project, from different timezones [8], [9]. Literature shows how this decision increases failure risks [7]. Different factors cause increased risks. For example, distance in space and time makes (cross-)dependencies among project tasks very tight; in this web of dependencies a single unmet deadline can cause a ripple effect compromising the whole project [19]. Also, the cultural distance among developers in different cultural areas (e.g. Europe vs. India) can cause fear, mistrust or other "social" problems which hinder communication and collaboration [5].

An Organizational Social Structure (OSS) represents the emergent web of (social) ties, practices and cognitive approaches between individuals collaborating towards a common goal [25]. An OSS' purpose is to enable knowledge fluidity between individuals pursuing the same organizational goal (e.g. software in case of GSE) [3]. Within GSE, distributed teams collaborate (i.e. co-create knowledge) on software systems development (i.e. delivery being their final goal). By definition, this constitutes an OSS. Conway's law [16] underpins the importance of studying and supporting OSSs for software engineering. This is especially true for GSE, since global distance complicates social interactions.

In this paper we map the current state of the art in OSSs onto current practice in GSE. The current state of the art in OSSs is derived from a systematic literature review [23], while the current practice in GSE is derived from empirical research reported in [5], [18].

Figure 1 illustrates our two key observations. First, empirical research in [5], [18] identifies 25 organizational factors that an organization has to decide on, when embarking on a GSE project. Ten of these organizational factors are not matched by any OSS discussed in literature. These 10 unmatched factors relate to "process management" and "organizational efficiency". Second, the OSSs that best fit GSE have a number of attributes that address "awareness management" (e.g. awareness of people, awareness of their skills, of their allotted tasks, awareness of tasks (re-)localization as needed, etc.). This factor, namely, "awareness management", is not mentioned in the literature discussing organizational factors in GSE [5], [18].

Figure 1. Our Results: OSSs miss 10 GSE factors, and introduce 1 new factor.



The above observations led us to conclude that research in OSSs for GSE should focus on increasing its support to "process management" and "organizational efficiency" factors. Conversely, research in GSE should include factors focusing on awareness management.

The rest of the paper is structured as follows: section II provides an overview of the materials we used for this study (OSSs and GSE organizational factors) as well as the results we obtained in mapping them. Section III provides discussions and observations on results. Finally, section IV concludes the paper pointing to future work.

978-0-7695-4787-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICGSE.2012.25

II. Research Approach

A. What we used

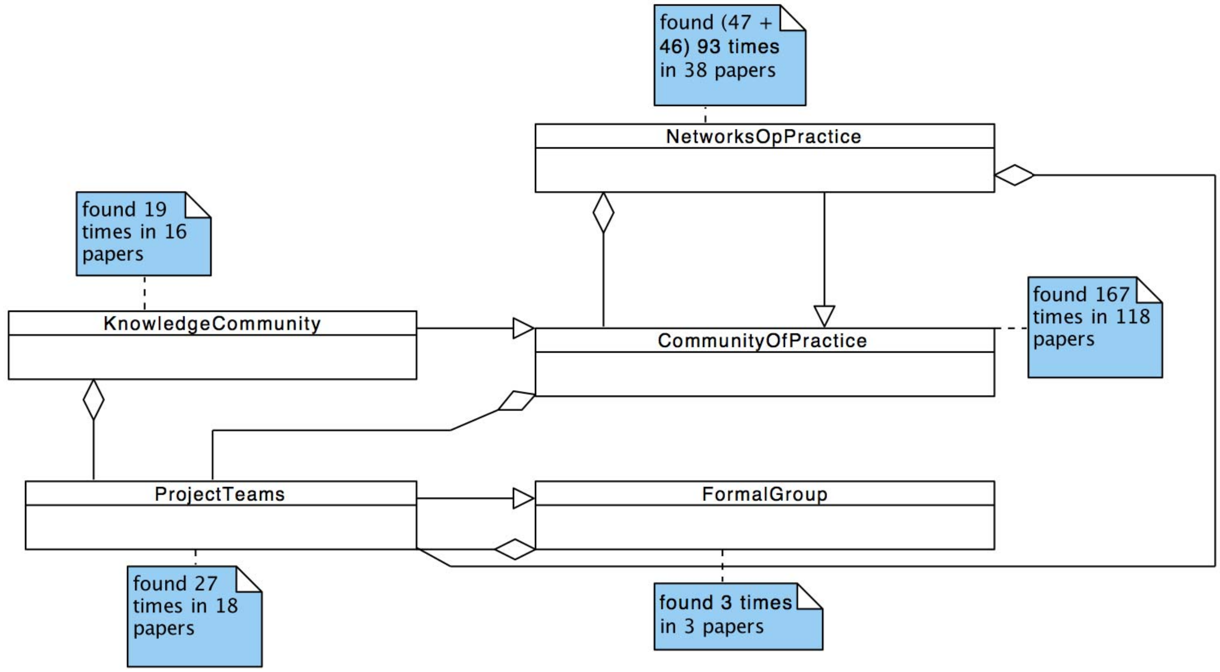
Two key contributions were used for the work in this paper. The first is a systematic literature review (SLR) we conducted into OSSs. The second is published in [5] and [18].

Using a grounded-theory approach [12] to study the literature we obtained 13 types of OSSs (and their defining attributes) [23]. To focus on the types which best reflect the GSE organizational structure, compared the 13 OSS types with definitions of GSE from [9], [8], [19] and [7]. We were left with 5 types: *Communities of Practice, Networks of Practice, Formal Groups, Knowledge Communities and classic Project Teams*. Figure 2 captures their mutual relations. The 5 remaining types are defined as follows:

1) *Communities of Practice (CoP):* quoting from [25] "[CoPs] are groups of people informally bound together by shared expertise and passion for a joint enterprise – engineers engaged in deep-water drilling, for example, consultants who specialize in strategic marketing, or frontline managers in charge of check processing at a large commercial bank". A CoP consists of co-located groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting frequently and in the same geolocation. Therefore CoPs serve as scaffolding for organizational learning in one specific practice. For a CoP to take place, a vital requirement must be satisfied: co-location. All developers have to meet in the same place, at the same time for the dynamics of CoPs to take place. For example, Software Architects' workshops in GSE projects can be seen as an instance of CoPs in GSE, since they entail, co-located professionals to share a common practice (Software Architecture) for the benefit of the project. Instances of this type in practice are (co-located) software architects' meetings: these are common in GSE to synchronize efforts and plan further progress.
2) *Networks of Practice (NoP):* Quoting from [11] "NoP comprises a larger, geographically dispersed group of participants engaged in a shared practice or common topic of interest [...] CoPs and NoPs share the characteristics of being emergent and self-organizing, and the participants create communication linkages inside and between organizations that provide an "invisible" net existing beside the formal organizational hierarchy". A NoP is a networked system of communication and collaboration that connects CoPs (which are localized). In principle anyone can join it without selection of candidates (e.g. an OpenSource forge, like SourceForge, is an instance of NoP). NoPs have a high geodispersion, i.e. they can span geographical and time distances alike. This increases their visibility and the reachability by members. An unspoken requirement for entry is the expected IT literacy of members. IT literacy must be high since the tools needed to take part in NoPs are IT-based (e.g. Micro-blogs, forums, hang-outs, etc.). NoPs are composed of CoPs (which are co-located). They inherit from CoPs the enforcement of shared repositories of knowledge for their members, as well as the presence of a common practice acting as an engagement within the network. Differently than CoPs, NoPs can be seen as IT-enabled global networks, since their chief aim is to allow communication (and collaboration) on the same practice through large geographical distance. For example, each GSE team (e.g. people, skills, documents, etc.) can be seen as a node within a GSE NoP [22]. In the GSE practice, Ericsson is known to use a NoP comprising of its own project teams and open-source communities, for the purpose of supporting its GSE efforts[1].
3) *Formal Groups (FG):* FGs, are exemplified in [10] as "[groups of] teams and/or workgroups [...]. Numerous different definitions of diversity have been put forth; however, they generally distinguish between two main sets of characteristics [for FGs]: 1) diversity of observable or visible detectable attributes such as ethnic background, age, and gender; 2) diversity with respect to non-observable, less visible or underlying attributes such as knowledge disciplines and business experiences". FGs a set of people which is explicitly grouped by corporations to act on (or by means of) them (e.g. governing employees or ease their job or practice, by grouping them in areas of interests). Each group has a single organizational goal (governing boards are groups of executives whose goal is to devise and apply governance practices). In comparison to other types, they seldom rely on networking technologies to link their members, on the contrary, they are local in nature. Moreover, it is very common for organizations to have these groups and extract project teams out of them (and therefore they are composed of project teams). Moreover, since project teams are instances of formal groups but tailored specifically to solve a particular problem, they inherit organizational aspects of formal groups such as clear-cut definition of tasks, complementary set of skills, etc. A perfect example of an FG is the JPL (Jet Propulsion Lab) within NASA. An example in the GSE domain can be seen in the SCR group at Siemens, in which Siemens researchers, work collaboratively to develop best practices [17]. Formal groups are very similar to the organizational units, or "sites", which are used in

[1] http://www.ericsson.com/yourbusiness/developers/open_source

Figure 2. OSS types relevant in GSE domains



GSE.

4) *Knowledge Communities (KC):* Quoting from [6] "Virtual knowledge communities are organized groups of experts and other interested parties, who exchange knowledge on their field of expertise or knowledge domain in cyberspace within and across corporate and geographical borders. Virtual knowledge communities focus on their knowledge domain and over time expand their expertise through collaboration. They interact around relevant issues and build a common knowledge base". Essentially KCs, are groups of people with a shared passion to create, use, and share new knowledge for tangible business purposes (e.g. increased sales, increased product offer, clients profiling, etc.). The main difference with other types is in their specific tie to precise business goals for the organizational sponsor. Moreover, they are not limited to use electronic communication and collaboration means (such as NoPs) but rather they inherit from CoPs the enforcement of co-located meetings or workshops to devise or explore new ideas. Specific industrial groups such as the JDA global alliance program [2], focused on supply-chains, can be seen as a knowledge community for GSE, since they focus on best practices and knowledge interchange around problems specific to (a specific domain of) GSE.
5) *Project Teams (PT):* Lindkvist [13] provides a general definition of PTs with the following words: "[PTs are] temporary organizations or project groups within firms [that] consist of people, most of whom have not met before, who have to engage in swift socialization and carry out a pre-specified task within set limits as to time and costs. Moreover, they comprise a mix of individuals with highly specialized competences, making it difficult to establish shared understandings or a common knowledge base". PTs are made by people with complementary skills who work together to achieve a common purpose for which they are accountable. They are enforced by their organization and follow specific strategies or organizational guidelines (e.g. time-to-market, effectiveness, low-cost, etc.). Their final goal is delivery of a product or service which responds to the requirements provided. Compared to the other OSS types, they are the most formal type of group. PTs are also defined as strict and single-minded aggregates of people, (closely) collaborating on well-defined reification tasks (i.e. tasks which produce a tangible artifact which justifies their effort). Any Scrum project team, e.g. in [26], is a project team.

[2]available at: http://www.jda.com/alliances/partners-index/

Table II (in appendix) compares the OSS types. Column one, contains OSS attribute types. Columns two to six, contain attributes values. Each OSS type is identified by a unique set of attributes' value which, observed empirically for that specific type. For instance the "geodispersion" attribute type, has value "Network-Spanning" in NoPs, since nodes in a NoP are distanced in both time and space (i.e. every one has a different geolocation and their dispersion is network wide). The remaining types from our study in [23] don't comply with GSE definitions in [9], [8], [19] or [7]. This doesn't mean they are not emerging in GSE. For example, *Problem Solving Communities (PSCs)* entail extremely experienced professionals to focus on solving a specific problem to achieve strategic business advantage. PSCs are distant from GSE organizational practice, because single GSE projects don't pursue the strategic advantage of organizational sponsors, but rather *use* the strategy advantage of round-the-clock productivity to deliver products faster. However, PSCs are used in GSE as support communities to solve specific/recurring problems. Other similar examples involve *Learning Communities* or *Strategic Communities*, which are specific to learning and development of best-practices. These communities are not intended to share a practice for a purpose (as in CoPs or NoPs), rather they are bent on pure learning, e.g. to educate beginning GSE practitioners (e.g. an agile methods learning group could be used to speed up the learning curve). Lastly, types such as *Social Networks* and *Informal Networks* are still too generic to be explicitly considered as an organizational unit in GSE.

To understand which (sum) of these types matched GSE, we used empirical evidence from [5] and [18]. In both works, the authors present action research results conducted within large GSE corporations, over a period of nine years. Through action research, 25 organizational factors were derived and refined, based on observed organizational issues. The 25 factors are defined as key decisions that need to be taken for management and governance in GSE. Table IV (in the appendix) shows the 25 factors and exemplifies the decisions to be made for each factor. On Table IV we made a distinction: the top part shows **"socio-organizational factors"** which refer to structure or operational behavior of teams (i.e. relating to people involved); the bottom part, in bold, shows **"process management and efficiency factors"** which refer to aspects of the software process or the efficiency of the GSE organizational structure (i.e. relating to the processes and approaches adopted).

B. Mapping OSS Types to GSE Factors

In the previous section we reported five OSS types which were similar (by definition) to GSE. To understand which combination of these five could effectively support GSE, we did a systematic concept mapping of GSE organizational factors onto OSS types. All 25 factors were mapped with attributes' values from the OSS types in Figure 2, and vice versa. More in particular, the following rule was applied:

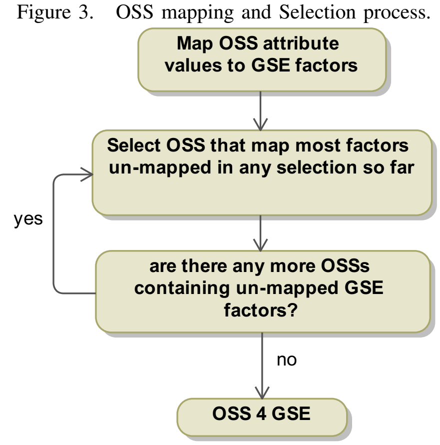
An OSS attribute "X" is mapped to a GSE organizational factor "Y", if and only if X's value is a decision about organizational factor Y

For example, the "Visibility" organizational factor was mapped to the attribute value "Visibility = Highest" from Knowledge Communities. Consequently, the "Knowledge Community" type is supportive to GSE, since it supports its "Visibility" organizational factor. This mapping ensures that GSE organizational factors are mapped to OSS attributes (and the type they represent) which explicitly support these factors.

To select OSS types consistently, we used a "greedy" rule, i.e. we selected a minimal set of OSSs by applying the following rule:

An OSS is selected if and only if its attributes' values, map to a set of organizational factors which were not present in previous OSS selections.

Figure 3. OSS mapping and Selection process.



The mapping process is summarized in the UML-style activity diagram in Figure 3.

As a result of the mapping process, 4 types were selected, namely: *Project Teams, Networks of Practice, Knowledge Communities and Formal Groups*. The resulting OSS composite can be described as follows: *GSE practitioners are organized in project teams. Also, they are forced to collaborate within a network (through the internet, or VPNs, etc.). In this network they share a practice (global software engineering) and the resources relevant to it (e.g. software artifacts being produced or used). Moreover, practitioners in GSE carry out knowledge intensive activities (i.e. preparing documentation, resolving requirement conflicts, making design decisions, etc.) across time, space and culture distance. GSE practitioners' cooperation in knowledge-intensive activities across time and space, makes them similar to knowledge communities. Finally, GSE practitioners are formally acknowledged (and governed) as (de-)centralized groups (e.g. development "sites").*

Table III (in the appendix) is a is a 19 x 25 matrix containing the "OSS attributes' value" vs. "GSE organizational factors". Columns in bold are factors which remain unmatched by any attribute of any of the four selected OSS types.

Every (I x J) cell matches the I-th attribute value (and consequently the OSS type to which it belongs) with the organizational factor in the J-th column. As a result, the OSS composite we selected, has two sets of attributes: (a) attributes that map an OSS type to GSE organizational factors; (b) other attributes of (mapped) OSS types.

III. Discussion

Two key observations were made on our results.

10 out of 25 organizational factors are not covered by any OSS (in bold on table IV and table III). These are *"Project Management"*, *"Efficient Partitioning"*, *"Risk Management"*, *"Language Selection"*, *"Tools"*, *"Culture"*, *"Information"*, *"True Cost"*, *"Reporting"* and *"Process"*. It is noticeable that all of the un-matched attributes are into the **"process management"** and **"organizational efficiency"** category (see section II), except *"Culture"* which can be a considered cross-cutting concern. This suggests OSS literature has not (fully) explored these factors yet. This should not come as a surprise, since OSSs focus on organizing people rather than supporting explicitly business purposes (i.e. intended to produce a tangible output such as software). This notwithstanding, we look at additional ramifications of the OSS meta-model resulting from our systematic literature review [5]. We found that no combination of OSSs can offer support for all these factors: the "Strategic Community" type (a specialization of Communities of Practice) offers support to *"Project Management"*, *"True Cost"*, and *"Fear"* through organizational sponsoring practices, contract value management, as well as team partitioning guidelines (in the form of "previous experience" policies, "personal goals"). This suggests that the OSS type for GSE could be enriched by integrating features of a strategic community. Finally, the "Workgroup" type (which inherits from Networks of Practice) provides cohesion practices that could support both *"Risk Management"* and *"Fear"*.

Observation: *Process management and organizational efficiency in GSE could benefit from support in the GSE OSS. Additional research should be invested in constructing an OSS hybrid which can cover all 25 organizational factors.*

Table I shows the composite OSS for GSE: column 1 contains the OSS types; column 2 contains the attributes' values, rephrased to represent the GSE domain; column 3 provides a label for unique identification; column 4 distinguishes new attributes (with a capital "yes") from others that were previously explored in GSE literature (i.e. are matched by organizational factors). As expected, many of the attributes in Table I (e.g. attributes R2a and R2e) are not new to software engineering practice and, specifically, to GSE (e.g. [7], [4], [9]). This confirms that software engineering research has moved well in coping with many social and organizational factors occurring in GSE. On the other hand many attributes deriving from OSS characteristics, are new.

Attribute R1a states this explicitly, by calling for dynamic indexing and retrieval of professionals (i.e. the management must be aware of "who" is able to do "what", and "where"). Attributes R2b, R2f, R3d, R3e, suggest ways in which people should be modeled or organized so that their skills can be easily retrieved or switched (i.e. people should be organized to support the awareness of their abilities). Finally, attribute R3g, states a way in which people should be formatted in a federated social network to allow for their cooperation. At a first glance, from the description in table I, these new attributes seem to fall under the "skills management" factor. Rather, with the exception of R2i (which can be seen as a concern cross-cutting all factors), they all address a different concern: awareness management (e.g. awareness of skills, awareness of the location and skills of certain people, awareness of task allocations, awareness of possible (re-)allocations of tasks to skills and people etc.).

Observation: *This trend in the attributes indicates an emergent GSE factor focusing on awareness management in GSE. Using the new attributes, ad-hoc support tools could be developed, to support this new concern (e.g. an adaptable and dynamic social network of skills, rather than teams, to allow for their (re-)localization as needed).*

IV. Conclusions and Future Work

This paper offers a profile of the organizational social structure for GSE. Our data and discussions support two key conclusions.

First, current OSSs fail to support *"process management and efficiency"* factors in GSE. Additional investigation should be invested in devising an OSS which matches all 25 organizational factors relevant in GSE. For example, the definition of the complete OSS for GSE could be used to devise ad-hoc support tools to bootstrap GSE projects and monitor them.

Second, current practices in management and governance of GSE, have focused on the process, on coordinating organizations involved, organizing teams into coherent working units: additional effort should be invested in exploring mechanisms for awareness management in GSE. For example, mechanisms to support representation and (re-)localization of skills could be critical when certain project tasks remain dangling (e.g. as a consequence of employee turnover). Also, awareness should be supported at different granularity levels (i.e. skills, people, tasks, etc.).

In the future, we plan to develop a prototype (based on Agile Service Networks and residing in the cloud [21]) to support OSSs emerging in GSE. Future work should also be invested in developing a context-model of the OSS defined in table I (e.g. by refining the one presented in [22]), so that context awareness and adaptation mechanisms can be developed for the GSE OSS. This can be done by investigating further in the OSS state of the art (e.g. as provided in [23]) to identify attributes and characteristics which are part of OSS context, and relevant to GSE.

[5] the diagram is available for reference at http://picfront.org/d/8oRX

OSSs	***attribute for GSE***	***Label***	***new?***
Knowledge Communities	the OSS_{GSE} type must support the application of management practices to index and dynamically retrieve skills from the professionals enrolled in it as needed (i.e. according to business demand); Evidence of this need is the increased usage for agile practices and supporting tools such as FogBugz	R1a	**YES**
	the OSS_{GSE} type should have as prime goal knowledge generation and sharing; This can already be seen in practice, by the increased adoption of big industrial players (e.g. SAP, IBM, etc.) of knowledge management strategies[3]	R1b	
	the OSS_{GSE} type should adopt all possible ways to increase both internal and external visibility (e.g. local promotion, bannering, seminars, ad-hoc trainings, etc.); GSE research in practice had already shown this, e.g. in [18]	R1c	
	the OSS_{GSE} type must envision ways to maintain its visibility at the highest level (e.g. by embedding itself with the technical space of the developers); as presented with specific tools for socio-technical congruence, e.g. [20]	R1d	
Networks of Practice	the OSS_{GSE} type should ensure communication openness; a testimony of the effectiveness of communication openness is the success of agile practices in GSE, as shown in [14]	R2a	
	the OSS_{GSE} type should make explicit the geolocation (e.g. the location in both time and space on the globe) of each node; as exemplified in the context model from [22]	R2b	**YES**
	the OSS_{GSE} type should support fine-grained skills management practices actionable on each node (e.g. it should be able to propose skill alternatives for each node); this strategy was already pointed out in classic software engineering research	R2c	
	the OSS_{GSE} type should clearly define which boundary objects (emails, blogs, RSS feeds, etc.) can be used and how (post-reply, knowledge repository, etc.); e.g. as in agile practices, the scrum coach or scrum master can be used as gateways to other sites, as in [14]	R2d	
	the OSS_{GSE} type should provide a shared repository of knowledge to be maintained (automatically); as already pointed out previously in global software engineering literature	R2e	
	the OSS_{GSE} type should be able to use the shared knowledge repository to tighten the geodispersion of each node from the others (e.g. by using massive geo-coding technologies to locate each and every resource contained); as also represented in practice by the emergence of enterprise social networking tools (e.g. Yammer)	R2f	**YES**
	the OSS_{GSE} type should allow the application of governance practices (e.g. agreed norms, sanctions, automated rewarding mechanisms, emotional management, social events, etc.) on each network node to maintain its high motivation; this requirement is also evident in practice from the strong reliance on coaches (e.g. in scrum)	R2g	
	the OSS_{GSE} type should support integration with the technical space(s) decided for the project it is supporting; again, the emergence of non-invasive enterprise social networks supports this requirement	R2h	
	the OSS_{GSE} type should allow the definition of (and agreement to) organizational practices (i.e. it should support the building of an organizational culture) based on company adopted standards and accepted values (e.g. as an entry pre-requisite to the OSS); as suggests also the increased usage of social capital management approaches	R2i	**YES**
Project Teams	the OSS_{GSE} type should support the guideline of "proficiency diversity = complementary" to support the definition of roles and responsibilities in project teams; this is a commonly acknowledged rule in classic software engineering research and practice	R3a	
	the OSS_{GSE} type should enable and nurture cohesion practices (e.g. proposal of team building initiatives, stand-up meetings, hang-outs, etc.) in project teams to maintain its high motivation; this is represented by the increased effectiveness of informal management techniques in international partnerships [24]	R3b	
	the OSS_{GSE} type should integrate collaborative networking or programming facilities (e.g. CVS, distributed black-board, etc.); again this is a basic rule of textbook GSE, as shown in [19]	R3c	
	the OSS_{GSE} type should allow the definition of an in-team technological gatekeeper, i.e. a person or entity which decides whom to forward certain technological-related issues or solutions; this also emerges from practice as shown in the effectiveness of agile approaches in GSE [14]	R3d	**YES**
	the OSS_{GSE} type should assume each developer (i.e. each node) is weakly tied to the rest of the network, in case seamless switching of skills is needed; as also underlined in [18]	R3e	**YES**
	the OSS_{GSE} type should nurture the creative problem solving abilities of project members (e.g. by integrating mind-mapping facilities); this is also shown in the effectiveness of problem solving communities in large industrial practice (e.g. in IBM[4])	R3f	
	the OSS_{GSE} type should allow the definition of a (federated) social network for local project teams; as the approach in [22] shows, each node in a service network can be a team, which is, by definition, a federation of skills	R3g	**YES**
Formal Groups	the OSS_{GSE} type should integrate governance mechanisms for emotional management to mitigate fear and its negative potentials; this fact is evident by the increased need for governance practices in GSE as shown in [1]	R4a	
	the OSS_{GSE} type should integrate trust-in-members mechanisms (e.g. members trust-estimation) and practices (e.g. team-building); this fact is evident from the increased need of informal bonding exercises and their proven effectiveness in GSE [15]	R4b	

Table I
OSS FOR GSE ATTRIBUTES

Acknowledgment

The authors acknowledge the support of the European Community's Seventh Programme FP7/2007-2013, grant agreement 215483 (S-Cube), for partially funding this project.

References

[1] Raja Bavani. Governance patterns in global software engineering: Best practices and lessons learned. In *ICGSE*, pages 50–54. IEEE, 2011.

[2] Anne Bourhis, Line Duba, and Raal Jacob. The success of virtual communities of practice: The leadership factor. *Electronic Journal of Knowledge Management*, 3(1):23–34, jul 2005.

[3] John Seely Brown and Paul Duguid. Knowledge and organization: A social-practice perspective. *Organization Science*, 12(2):198–213, mar 2001.

[4] E. Carmel and R. Agarwal. Tactical approaches for alleviating distance in global software development. *IEEE Software*, 2(18):22–29, March/April 2001.

[5] Valentine Casey and Ita Richardson. Implementation of global software development: a structured approach. *Software Process: Improvement and Practice*, 14(5):247–262, 2009.

[6] Angela M. Dickinson. Knowledge sharing in cyberspace: Virtual knowledge communities. pages 457–471, 2002.

[7] Christof Ebert and Philip De Neve. Surviving global software development. *IEEE Software*, 18(2):62–69, 2001.

[8] James D. Herbsleb. Global software engineering: The future of socio-technical coordination. In Lionel C. Briand and Alexander L. Wolf, editors, *FOSE*, pages 188–198, 2007.

[9] James D. Herbsleb and Audris Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6):481–94, 2003.

[10] Eli Hustad. Managing structural diversity: the case of boundary spanning networks. *Electronic Journal of Knowledge Management*, 5(4):399–409, dec 2007.

[11] Eli Hustad. Exploring knowledge work practices and evolution in distributed networks of practice. *Electronic Journal of Knowledge Management*, 8(1):69–78, jan 2010.

[12] Michael Jones and Irit Alony. Guiding the use of grounded theory in doctoral studies an example from the australian film industry. *International Journal of Doctoral Studies*, 2011.

[13] Lars Lindkvist. Knowledge communities and knowledge collectivities: A typology of knowledge work in groups. *Journal of Management Studies*, 42(6):1189–1210, sep 2005.

[14] Darja ?mite, Nils Brede Moe, PŁr J gerfalk, Jakub Rudzki, Imed Hammouda, Tuomas Mikkola, Karri Mustonen, and Tarja SystŁ. *Agility Across Time and Space*. Springer Berlin Heidelberg, 2010.

[15] Nils Brede Moe and Darja Šmite. Understanding a lack of trust in global software teams: a multiple-case study. *Softw. Process*, 13(3):217–231, May 2008.

[16] Nachiappan Nagappan, Brendan Murphy, and Victor Basili. The influence of organizational structure on software quality: an empirical case study. In *International conference on Software engineering*, pages 521–530, Leipzig, Germany, May 2008. IEEE.

[17] Hartmut Raffler, Matthias Schneider-Hufschmidt, and Thomas Khme. System ergonomics and human-computer interaction at siemens corporate research and development. In Penny Bauersfeld, John Bennett, and Gene Lynch, editors, *CHI*, pages 65–66. ACM, 1992.

[18] Ita Richardson, Valentine Casey, John Burton, and Fergal McCaffery. Global software engineering: A software process approach. In Ivan Mistrik, J. Grundy, A. van der Hoek, and J. Whitehead, editors, *Collaborative Software Engineering*. Springer, January 2010.

[19] Raghvinder Sangwan, Matthew Bass, Neel Mullick, Daniel J. Paulish, and Juergen Kazmeier. *Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series)*. Auerbach Publications, Boston, MA, USA, 2006.

[20] Anita Sarma, Larry Maccherone, Patrick Wagstrom, and James Herbsleb. Tesseract: Interactive visual exploration of socio-technical relationships in software development. In *Proceedings of the 31st International Conference on Software Engineering*, ICSE '09, pages 23–33, Washington, DC, USA, 2009. IEEE Computer Society.

[21] D. A. Tamburri and P. Lago. Satisfying cloud computing requirements with agile service networks. In *Proceedings of the 2011 IEEE World Congress on Services*, SERVICES '11, pages 501–506, Washington, DC, USA, 2011. IEEE Computer Society.

[22] D.A. Tamburri and P. Lago. Supporting communication and cooperation in global software development with agile service networks. In *Software Architecture*, Lecture Notes in Computer Science, pages Vol. 6903, 236–243. Springer Berlin / Heidelberg, 2011.

[23] Damian A. Tamburri, Patricia Lago, and Hans Van Vliet. Organizational social structures for software engineering. *ACM Computing Surveys - Under Review*, page 35, 2012.

[24] Sharon Watson and Gary R Weaver. How internationalization affects corporate ethics: formal structures and informal management behavior. *Journal of International Management*, 9(1):75 – 93, 2003.

[25] E. C. Wenger and W. M. Snyder. Communities of practice: The organizational frontier. *Harvard Business Review*, 78(1):139, January 2000.

[26] Laurie Williams, Gabe Brown, Adam Meltzer, and Nachiappan Nagappan. Scrum + engineering practices: Experiences of three microsoft teams. In *ESEM*, pages 463–471. IEEE, 2011.

	Knowledge Community	*Project Teams*	*Networks of Practice*	*Communities of Practice*	*Formal Groups*
Visibility	Highest	-	-	-	-
Management Practices	adaptable combination of skills in teams	-	enforce self-management	-	Rigid and inflexible
Proficiency Diversity	-	complementary	-	-	-
Creation Process	-	-	-	-	formal appointment
Members Previous Experience	-	-	-	-	selected through entry requirements
Critical Success Factors	-	in-team social network, Weakly Tied-ness, Creative Problem Solving	-	-	good governance
Members Cohesion	-	Milestone Based .. Social Closeness	-	co-localization	formal assignation
Boundary Spanning	-	-	High	-	-
Members Motivation	-	delivery	High	-	governance-based
Members Selection Process	-	skills	-	self-interest	formal assignation
Support Tool	-	-	Communication Platform, Persistent Computer Network, Interactive Community	-	-
Communication Openness	-	custom	open	open	closed
Knowledge Types	domain-specific	-	Tacit, Embedded	-	-
Geodispersion	-	-	Network-spanning	-	-
Shared Repository	-	-	yes	yes	-
Members Official Status	-	custom	-	participation	-
Membership Creation Process	-	-	-	self-selection, promotion	-
Perceived Competitiveness	-	-	-	None	-
Communication Media	-	-	electronic only	situated practice	-
Context Openness	-	custom	egalitarian	egalitarian	closed
Governance	-	-	-	-	Routinization .. Emotional Management .. Control .. Scientific Management

Table II
OSS COMPARISON TABLE

	Communication	Communication Tools	Temporal Issues	**Effective Partitioning**	**Project Management**	Skill Management	Knowledge Transfer	Defined Roles / Responsibilities	Team Selection	**Risk Management**	**Language**	Motivation	Technical Support	Coordination	Cooperation	**Tools**	**Culture**	Teamness	Visibility	**Information**	Fear	Trust	**True Cost**	**Reporting**	Process
Knowledge Community																									
-Visibility = Highest																			x						
-Management Practices = Combination						x																			
Project Teams																									
-Proficiency Diversity = Complementary								x																	
-Creation Process = Formal Appointment									x																
-Critical Success Factor = Creative Problem Solving									x																
-Members Previous Experience = Cross Functional									x																
-Critical Success Factor = Weakly Tiedness									x																
-Critical Success Factor = Within Team Social Network												x													
-Members Cohesion = Milestone Based .. Social Closeness								x				x						x							
Networks of Practice																									
-Boundary Spanning = High						x									x										
-Members Motivation = High						x						x													
-Members Selection Process = Entry Requirements						x																			
-Support Tool = Communication Platform, Persistent Computer Network, Interactive Community		x											x												
-Communication Openness = Open	x					x								x											
-Management Practices = Strong												x													
-Knowledge Type = Tacit							x																		
-Knowledge Type = Embedded							x																		
-Geodispersion = Whole Network			x																						
Formal Groups																									
-Governance = Routinization .. Emotional Management .. Control .. Scientific Management																					x	x			x

Table III
MATCHING OF OSS ATTRIBUTES TO GSE FACTORS.

Factor	*Example*
Communication	Team A needs to communicate heavily with teams in location B given the dependency between their work packages. Live and efficient mechanisms for communication must be selected.
Communication Tools	Team A and others must agree on a tool that they all should adopt, in order to avoid miscommunication as much as possible
Temporal Issues	Team A and Team B are working on the same work page in two contiguous shifts of 8 hours. Information continuity should be planned.
Effective Partitioning	Team A should be entrusted with continuing the workpackage of B only if Team B can communicate efficiently with A
Skill Management	Team A has a large array of skills, Team B has a specific set of skills. Team A and Team B should not work on the same workpackage
Knowledge Transfer	Team A, B and C need to realize round-the-clock productivity with three consecutive, 8-hour shifts. Information continuity must be allowed and transfer of information needs to be planned.
Defined Roles / Responsibilities	Person "a" in team A is leader and made responsible for timely delivery. If "a" turns over, then person "a.1" should take over.
Team Selection	Task 1 will be carried out by team A, which is composed of people a+b+c; Task 2 will be outsourced to partner X; Task 3 will be carried out by team B from project Y.
Motivation	Manager 1 in site X adopts informal leadership approaches [2] to motivate Teams A and B while they work within a large GSE project.
Technical Support	Teams X and Y will be available 24/7 to provide network-wide support on workspaces, IDEs, codebase versioning and (in case of emergency) back-ups
Coordination	Team 1 will finish working on Task A at day D+11; Team 2 will wait day D+12 then make sure the release of task A is complete and start integrating Task A and B.
Cooperation	Teams 1 and 2 must pool resources on task A, since it requires their combined set of skills.
Teamness	Developer X of team A often leaves daily stand-up meetings beforehand or is uncooperative towards women in the same team, he cannot work properly in team A (50% women).
Visibility	Teams A and B should render their progress visible to all the development network since their timely delivery is critical for the good-health of the whole project.
Trust	After every stand-up meeting, developer X in off-shore team A phones the headquarters to verify instructions received; developer X doesn't trust leadership in team A.
Fear	developers X and Y are often heard talking about moving to India since their current site will be closed due to out-sourcing; as a result, the rest of the developers at their site are developing uncooperatively towards outsourcing attempts.
Project Management	**Deliverable A.1 is late of three days; deliverable A.2 depends on A.1 but is more critical. Deliverable A.2 should be started from partial results of A.1 and adjusted live.**
Effective Partitioning	**Teams A, B and C are in three different timezones. Only two of these timezones are contiguous in shift (Teams A and B). Teams A and B should work on the same work package. Team C should work on a work package as independent as possible to Team B's.**
Risk Management	**Technological Gateway at site A, is developer X. X is constitutes a single point of failure. Developer Y should be instructed to follow X and take-over as needed.**
Language Selection	**language at all sites should be homogenized to english. All sites should select english-certified developers so that international collaboration is possible.**
Tools	**technical space of each JAVA developer should be eclipse-centric; technical space of each designer and modeler should make extensive use of UML technologies and standards based on it.**
Information	**three types of information should be supported within the technical space of each developer at each site: models; documents; codebase. All information should be exchanged through secured emails and traceable.**
True Cost	**underlying costs (over-times, holiday, latencies and idle-times, etc.) must be calculated at each site and final figures should be summed up at every monthly meeting.**
Culture	**Teams 1 and 2 can work hand in hand since they are part of the same nation and are not limited by different national holidays or shift-times.**
Reporting	**BIRT should be available at every workstation; every developer should comment every artifact produced or retouched; all should be aware of their gateway (human or technological) towards other sites.**
Process	**Scrum sprints will be used to develop the project.**

Table IV
25 ORGANIZATIONAL FACTORS IN GSE.