Managing Corporate Performance of Development Communities Through "Snapshots" of Their State

Damian A. Tamburri[a], Patricia Lago[a], Hans van Vliet[a], Martin von Weissenberg[b]

[a] *Information Management and Software Engineering group, VU University, Amsterdam, The Netherlands*
[*d.a.tamburri, p.lago, j.c.van.vliet*]*@vu.nl*
[b] *Department of Information Systems, Hanken School of Economics, Helsinki, Finland*
martin.vonweissenberg@hanken.fi

Abstract

Context: Engineering software is a social activity. Consequently, measuring and studying the organisational performance of the development community is key to success. Maintaining organisational performance becomes especially delicate when multiple communities have to join forces (e.g. in global software engineering).
Objective: So far, software engineering discipline lacks systematic methods to study development communities and their performance. In this paper we present ODeSSA a method to "Outline Development Social Structures to Analyse". The method is a refinement of previous work, and is validated on a real-life case-study from a large software development organisation. Applying ODeSSA we show that development communities can be given an intelligible form, which we call *snapshot*.
Method: ODeSSA stems from previous work based on grounded-theory. This paper illustrates ODeSSA and validates the method using a real-life industrial case-study.
Results: In validating ODeSSA through a case-study, we found that *Snapshot*s can have a limited number of forms, each with their own peculiarities. Finally, we found that ODeSSA helps detecting organisational barriers that hinder the developer community performance.
Conclusion: We conclude that ODeSSA determines intelligible forms, or *Snapshots*, for organizations. These forms must be studied, e.g. to determine causes for low organisational performance.

Keywords:
Social Structures; Software Development Organisations; Networked Organisations; Organisational Performance; Organisational Complexity;

1. Introduction

The craft of software is becoming more and more a social activity [1]. Several studies suggest that engineers spend between 70 and 85% of their time working with other people (e.g. end-user focus groups, managers, business sponsors, etc.) [2]. This means that in 70-85% of times, software engineering success is a community effort, rather than a success of individuals. This figure becomes critical with the increase of diversity and size of the development community. For example, Global Software Engineering (GSE) is a development strategy entailing global teams to collaborate together from different locations in different timezones [3]. In this circumstance, distances in time, space and culture, magnify project risks connected to failures in the development social structure [4, 5, 6]. The problem persists since software practitioners lack a systematic method to visualise and manage the organisational state of development communities. In previous work [1] we provided a mechanism to uncover social communities latent emerging during software development. Stemming from previous work in [1], this paper proposes ODeSSA a semi-automatic method for "Outlining Development Social Structures to Analyse". ODeSSA allows the observers of a development community to capture its key information in the form of analysable "*snapshots*". A *snapshot* for a community is obtained mapping answers to a questionnaire onto a community decision-tree [1]. The resulting *snapshot* is a "colouring" [7] of the decision-tree from [1]. Fig. 1 shows an example of a *snapshot*. The *snapshot* represents the characteristics of text-book software project teams [8].

This paper presents the ODeSSA method and validates through a case-study. Applying ODeSSA, we made three key observations: (a) the method allows uncovering inconsistencies on the observed *snapshot*; (b) the community *snapshot* can have a limited number of possible forms, each with its own particular characteristics; (c) the method allows uncovering of organisational barriers that can hinder communication/collaboration across communities during software development.

The rest of the paper is structured as follows: Section 2 explains materials

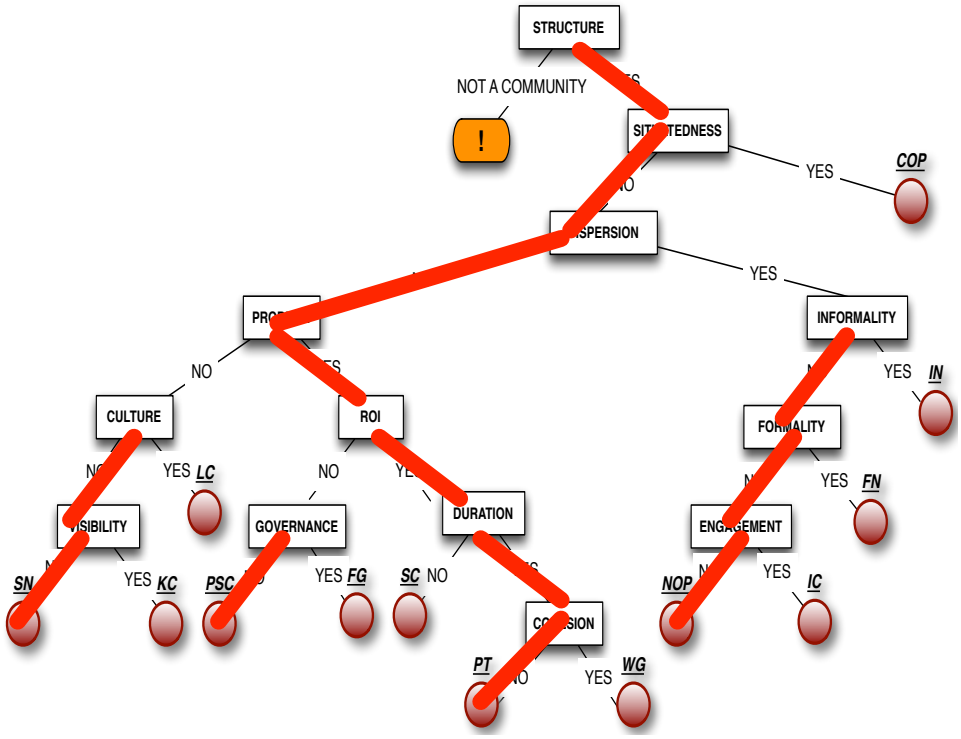

Figure 1: an example snapshot for text-book project teams.

and methods; Section 3 presents the case-study, while Section 4 discusses results. Finally, Section 6 concludes the paper with future work.

1.1. Studying and Supporting Software Development Communities

Literature in GSE already recognises the need to study global communities. More in particular, many works such as [5] have studied empirically the effect of organisational structures on the quality of software. These works motivate further studies to understand, represent and support social structures and communities. In addition, similar works such as [9, 10] study the effect of socio-technical congruence on designing global systems as well as the process of global engineering. Such works suggests that understanding how coordination requirements map to their people counterpart is critical to steer information across global communities successfully. Our work is limited to identifying community types, and finding ways to "format" and study their observable characteristics. Other works such as [11] study the impact of successful communities on socio-technical problems (e.g. coordination, shared

understanding, etc.). These works take a more social and organisational approach, pointing out the paramount importance of supporting explicitly the work of communities rather than allocating tasks to single individuals. In previous work [6, 12], we found that observable *social communities* captured in [11] are a composition of known social community types [12]. The work presented in this paper goes one step further in this direction, by reporting on the results of a large industrial real-life scenario. The results led us to develop a method to "snapshot" software development communities by using a composition of known social community types. Finally, works such as [13, 14] investigate the impact of organizational decision-making on software development. For example, understanding if the current community *snapshot* is performant (or even compatible) with offshoring is essential for the community to "go global" [15]. The method presented in this paper can be used to support organisational decisions, investigating the current and to-be community *snapshots*. Finally, the method enables understanding the negative consequences of change, such as emergent organisational barriers [16].

1.2. Research Questions

This paper reports an empirical study of software development organisations. We pay particular attention on the layout of the organisation. Our work was driven by the following research questions:

What elements of a development community should be made explicit to study its instantaneous state?

Previous research in organisational studies, social-networks analysis (SNA) and software engineering, suggest there are many variables at play within a software development community. In previous research [12, 1], we provided an extensive specification of attributes that define community types. Table 1 provides an overview of community types. These types were obtained through a Systematic Literature Review on organisations and social-networks research [12]. In this work we present information essential to define a *snapshot* for an observed organisation. By examining interviews and other sources of organisational evidence, we found a way to represent a *snapshot* of an observed community, such that the *snapshot* can be further analysed.

What are the possible forms of the development community *snapshot*?

Label	Name	Description
COP	Communities of practice	A CoP consists of groups of people who share a concern, a set of problems, or a passion about a topic or practice. These people deepen their knowledge and expertise in their topic or practice by interacting frequently, face-to-face, collaboratively (helping each other) and constructively (to increase their mutual knowledge). This set of social processes is called situatedness [17, 18].
IN	Informal Networks	INs can be seen as loose networks of ties between individuals that happen to come in contact in the same context. The driving force of the informal network is the strength of these ties between members. Finally, an IN differs from other types since it does not use governance practices but its success is solely based on the cohesion of its members [19].
FN	Formal Networks	Within FNs, members are rigorously selected and prescribed. They are forcibly generated and acknowledged by management of the network itself. Direction is carried out according to corporate strategy and its mission is to follow this strategy [12].
IC	Informal Communities	ICs are usually sets of people part of an organisation, with a common interest, often closely dependent on their practice. They interact informally, usually across unbound distances, frequently on a common history or culture (e.g. shared ideas, experience etc). The main differ- ence they have with all communities (with the exception of NoPs) is that their localization is necessarily dispersed so that the community can reach a wider audience [12].
NOP	Networks of Practice	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. OpenSource forges are an instance of NoP). NoPs have a high geodispersion, i.e. they can span geographical and time distances alike. The high geodispersion increases their visibility and the reachability by members. An unspoken requirement for entry is the expected IT literacy of members [18].
WG	Workgroups	WG are groups of technical experts whose goals span an entire business area or array of organisational factors. WGs are always accompanied by a number of organisational sponsors and are expected to generate benefits as wide as their goals. What's also fundamental, is that a WG is always acknowledged and supported by organisational sponsor(s).
PT	Project-Teams	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 organisation and follow specific strategies or organisational 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 [12].
SC	Strategic Communities	SCs consist of meticulously selected people, experts in certain sectors of interest to a corporation or or a set of organisational partners tied with formal non-disclosure agreements. These try to proactively solve problems within strategic business areas of the organisational sponsor.
FG	Formal Groups	FGs are comprised of people which are 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 interest). Each group has a single organisational goal, called mission (governing boards are groups of executives whose mission is to devise and apply governance practices successfully). In comparison to Formal Networks, they seldom rely on networking technologies, on the contrary, they are local in nature.
PSC	Problem-solving Communities	PSCs can be seen as a specific instance of a Strategic Community focused on a particular problem. One would expect this community to be formal in nature. Contrarily we found that they emerge as informal, since informality aids brainstorming and problem-solving processes.
LC	Learning Communities	LCs provide a space for pure learning and explicit sharing of actionable knowledge (i.e. skills). In a learning community, the leadership is expected to steer the community's practices and membership is subject to approval and tied to the learning objectives given to the member. Each developed or exchanged practice must become part of the organisational culture [20].
KC	Knowledge Communities	KCs are groups of people with a shared passion to create, use, and share knowledge for tangible business purposes (e.g. increased sales, clients profiling, etc.). The main difference with other types is that KCs are expected (by the corporate sponsors) to produce actionable knowledge (knowledge which can be put to immediate action e.g. best-practices, standards, methodologies, approaches, problem-solving-patterns, etc.) into a specific business area [21].
SN	Social Networks	SNs represent the emergent network of social ties spontaneously arising between individuals who share, either willingly or not, a practice or common interest on a problem. SNs act as a gateway to communicating communities [19].

In previous work [6, 1] we found many combinations of communities that can represent software development. In this work we analysed the possible "colourings" of the decision-tree in [1] using data from a big industrial case-study as a reference. We found there are four possible "colourings" that the data can reflect.

2. Materials and Methods

The conclusions in this paper are drawn from a real-life industrial case-study. This section explains what materials we used to carry out the case study and how we examined such materials.

2.1. Materials

The empirical evidence used in our work, stems from empirical research conducted in a big organisation, active in the mobile technologies market. The organisation (called Company X from now on) is multinational that develops both hardware and software components for end-users' mobile-phones and apps to enrich user experience. Company X employs over 130K people in 16 worldwide sites, with sales in over 160 countries. At the time the empirical research was conducted (early 2012), the organisation had a net market sale of 42.4 billion euros.

The focus of our data lies in the organisational structure of a product-chain within Company X's corporate portfolio. Our empirical data describes the overall organisation and some of its software development sites. The data also reports on the collaboration between Company X's Scrum development community and a constellation of open-source communities.

Three key sources of evidence were used:

1. The first key resource (called REF1 from now on) is a research report studying agile and open-source adoption in Company X. The study was conducted using grounded-theory [22]. The study features six interviews with focus groups from Company X. Focus groups had varied expertise, from areas such as project management to agile coaches to code developers. The focus sessions are analysed through grounded theory. The study offers an overview of the organisational structure of Company X as a whole (i.e. as a global community) as well as explaining its local sites. The research focuses on a division of Company X, working on a specific project release, at the time of investigation.

The report is 112 pages long, we use page numbers for the purpose of traceability.

2. The second key resource (called REF2 from now on) is a research report studying three software engineering teams within Company X. The study features 16 interviews harvested through accidental and snowball sampling [23]. Interviewees had varied backgrounds but an average work experience in Company X of about 10 years. The sample consisted of both sexes, however, with a minority of females. As a result, the organisational details of this report focus around software process, software coding methods, development best-practices adopted, problems found, and ultimately, the people's perspective on all the above. The report is 97 pages long, we use page numbers for the purpose of traceability.
3. The third key resource (called AGILE SLIDES from now on) is a set of tutorial slides that agile coaches within Company X used to introduce X's organisational structure/process to newly appointed staff members. The slides provide an overview of the agile process and open-source integration protocols in place within Company X at the time of investigation. The report is 4 pages long, we use page numbers for the purpose of traceability.

All of the above sources are dated around 2012 and are protected by non-disclosure agreements [1]. At the time of investigation, Company X was going through many organisational changes and both studies aimed at understanding the impact of organisational change on teams, their production velocity, community productivity and end-product quality.

2.2. Methods

The ODeSSA method is organised as follows:

1. *Gather Data:* The questionnaire in Fig. 2 must first be used as a checklist to gather necessary organisational details. Once analysed, these details must enable answering the questions in the questionnaire. Missing data can be gathered by directly surveying (e.g. with online forms) members of the community under observation. Alternatively, other quantitative mechanisms can be used, such as Social-Network Analysis (SNA) [25].

[1] available upon signed request

2. *Fill-in Questionnaire:* The questionnaire must then be filled out using the gathered data. The questionnaire is a lightweight mechanism to analysing observable development communities, rather than using complex coding schemes on available evidence, as seen previously in literature [26, 25].
3. *Identify Pivots:* Pivot points are points in the questionnaire that have both YES and NO answers. These points deserve further attention since the indicate one of two things: (a) there is a conflict or inconsistency in the analysed snapshot; (b) there is a sub-community within the analysed community.
4. *Map to Decision-Tree:* Once organisational evidence is ready, the decision-tree can be mapped on the questionnaire to uncover community types. A community *snapshot* can be obtained by combining information from the questionnaire with reflected community types.
5. *Evaluate snapshot form:* The properties of the obtained *snapshot* must be evaluated using the forms presented in Section 4.

To validate the ODeSSA method we conducted a case-study following guidelines in [24]. The case-study objective was to evaluate real-life development communities from a big industrial case using results from previous work [1]. The work in [1] presents a decision-tree that allows to identify community types (summarised on Table 1) by means of key defining attributes. Besides providing validation to the decision framework in [1], the connotation of our case-study was exploratory and descriptive. We set out to define an intelligible form for the observed development community, such that the outlined form could be further studied, e.g. compared to organisational metrics for Company X. These followup quantitative studies are out of the scope of this work.

We used a parallel study and data sources triangulation [27] to ensure the generalizability of the approach. In addition, the use of a parallel study was meant to minimise bias and unreliability of results through observer triangulation, according to [27]. While the two studies were started at the same time, both researchers conducting them were prohibited to share increments of partial results. Weekly meetings were planned to assess progress status only. The supervision of two senior researchers was used throughout the duration of the study.

ODT QUESTIONS	**Y**	**N**	ID	RATIONALE
Can I distinguish a non-trivial structure within the observed people?			1	
Do community members share a situated practice?			2	
Do all community members exhibit a different location?			3	
Do community members use informal communication only?			4	
Do community members exhibit longevity bound to a project?			5	
Do community members actively try to increase visibility of their operations and results?			6	
Do community members actively pursue cohesion?			7	
Are community members expected to demonstrate ROI to business sponsors?			8	
Does the community effectiveness depend on people' engagement?			9	
Do community members accumulate organisational culture?			10	
Do community members require formal status evaluation?			11	
Do community members tackle one explicit organisational problem?			12	
Do community members endure formal governance practices?			13	

Figure 2: a questionnaire to study development communities.

3. Results

To apply the ODeSSA method on our case-study, we analysed available evidence. The evidence available reflected the following scenario:

Company X works with an integrated Scrum model, using many divisions of work, e.g. business units, or product teams. Communication takes place informally only during certain phases. Other phases (such as stakeholder reviews, corporate progress meetings, product release plans, etc.) are regulated by rigid governance policies and protocols. New personnel is formally evaluated and selected to integrate the community. Both Company X and its clients actively encourage and motivate the development community to increase engagement and cohesion. Organisational culture (such as best-practices, recurrent product requirements, collaboration patterns, etc.) is neither collected nor maintained. Rather, the experience of personnel is established during evaluation of background and skills (i.e. during personnel review and initial evaluation before employment). Finally, Company X doesn't employ any mechanism to visualise product development process but leaves it up to Scrum masters and other leadership to disseminate results and increase their visibility.

4. Discussion and Findings

TODO:
We analysed our results and made three key observations. The following text states the observation and provides an interpretation in *italic*.

First, pivot points represent key variation points distinguishing a community from its subcomponents. For example, In our case study we found evidence that led to discovering NoPs. Likewise, we discovered evidence that could lead to discovering WGs. The only point of indecision was resting in the DISPERSION node (see part C of Fig. **??**). Referencing both visits with the evidence available we observed that the path leading to WGs was only visitable if we focused on some particular sites within Company X. Conversely, the path leading to NoPs was only visitable if we considered Company X as a whole, without focusing on any single site or unit within. It follows that the DISPERSION node was a key organisational variation point between Company X as a whole and its various sites. *Pivots can be used as drivers for the analysis of the community snapshot. Their presence indicates to practitioners that there is misalignment between a community as a whole and (some of) its components. This misalignment suggests organisational complexity. Practitioners can use them to investigated further into the complexity, e.g. to verify if complexity is indicator of low organisational performance.*

Second, we found that ODeSSA *snapshots* can assume four possible forms:

1. *simple*: the answers in the questionnaire reflect a single, clear-cut path on the decision tree. The community is well delineated and reflected by a type which is clearly investigated in organisational literature. Practitioners can use results from organisational literature to steer the community or improve its performance.
2. *enriched-community*: the answers in the questionnaire reflect one (or more) path blended with additional "enriching" attributes. The enrichments indicate the need of additional communities. Practitioners can use state-of-the-art organisational design techniques, e.g. to explicitly support the identified community needs. For example, suppose the "visibility" attribute is discovered in mix with the Formal Networks community type. Formality within formal networks might have forced the organisation to adopt mechanisms to increase the visibility of projects, people or artefacts. Refinement and explicit support to such technologies includes the adoption of Information Management or Social Networking technologies [28, 29] to support development community.
3. *complex*: The answers in the questionnaire reflect more than one path, i.e. more than one community. This form reflects the presence of pivots. The snapshot can be used to identify reasons for failing organisational performance. For example, in our case, many interviews suggested performance was hindered by an organisational barrier [16], "lack of visibility of the whole" (e.g. lack of visibility of people, artefacts, processes, etc.). We found the *snapshot* of Company X was in *complex* form but the "VISIBILITY" attribute was set to NO. In these circumstances the creation of a Knowledge Community within Company X is beneficial to increase visibility.
4. *chaotic*: The answers in the questionnaire reflect sparse attributes that don't represent any path but only broken segments. This indicates that the community is in some disarray, since it doesn't show any clear organisational form. The community is not representable with known organisational types. The snapshot form can be used to identify the attributes that remain unsupported (i.e. the broken links). For example, suppose you are observing a set of people working almost as a Workgroup but are not required to show any benefit on investments. This shortcoming can be made explicit and further investigated.

These five forms can be used as a means to measure and analyse community complexity. If the community matches a single path then it's more controllable, since it refers to a single type known to literature. Steering and management complexity increases the more the observed community diverges from a simple form.

Finally, organisational literature defines barriers as mechanisms or circumstances that filter organisational activity [16]. We found that many barriers are present in software development communities but are not necessarily a bad thing. In our case, for example, we found the limited "visibility of the whole" to be an organisational barrier. In literature [12] we have found evidence that this circumstance is often used by organisations to limit the circulation of industrial secrets across communities. This is consistent with the community we observed in Company X. We observed a strong cooperation between Company X and open-source communities. The mechanisms we provide in this paper can aid the discovery or deployment of organisational barriers, e.g. to increase or limit communication across (sub-)communities.

5. Threats to Validity

Based on the taxonomy in [30], there are four potential validity threat areas, namely: external, construct, internal, and conclusion validity.

"External Validity" concerns the applicability of the results in a more general context. The validity of our conclusions could be limited to the investigated organisation. To make sure our results were generalizable we carried out a confirmatory study, focused on confirming our observations with people from Company X. The confirmatory study was also focused on assessing the (re-)applicability of our results in different contexts, based on the experience of the interviewees.

"Construct Validity" and *"Internal Validity"* concern the generalizability of the constructs under study, as well as the methods used to study and analyse data (e.g. the types of bias involved). To mitigate these threats, our methods were tailored to use multiple triangulation as much as possible. Partial results and incremental analysis was conducted to gather constant feedback by two senior researchers, experienced in empirical methods.

"Conclusion Validity" concerns the degree to which our conclusions are reasonable based on our data. Our conclusions were drawn by an analysis of empirical evidence using known and confirmed methods from literature such as gap and taxonomy analysis. The approach and instruments that we

used to gather such evidence were presented and validated in previous work [6, 15, 1].

6. Conclusion and Future Work

the paper introduced a case-study conducted to apply the method from [1] in a real-life scenario. The case-study led us to define ODeSSA a method to Outline Development Social Structures to Analyze. We reported on the case-study findings to answer our research questions.

We answered research question 1 ("What elements of a development community should be made explicit to study its current status?"), defining a development community snapshot by combining: (a) answers to the questionnaire in Fig. 2; (b) visits to the decision-tree previously introduced and (partially) validated in [1]). This was realised by understanding that our case-study was made possible only with the presence of both elements. We answered research question 2 ("What are the possible forms of the development community status, or *snapshot*?"), by further analysing the possible combinations that the snapshot could present. With data from our case-study we understood what these forms meant, so that practitioners can re-apply ODeSSA for their own benefit. We answered research question 3 ("How does organisational change influence the form of a development community *snapshot*"), by sensing the presence of an organisational barrier that inhibited the effectiveness of change within Company X.

All results are based on qualitative research. The validity of all results would benefit greatly by the presence of accompanying quantitative evidence. Much work is still needed to carry out followup quantitative studies. We plan to establish quantitative metrics that map on the organisational complexity framework introduced in this paper. These metrics could be used by practitioners to understand community performance in numbers, e.g. to evaluate the effectiveness of their process model in combination with the resulting community.

Moreover, additional work could be invested in demonstrating the Conway effect [31], e.g. by studying communities of designers and their resulting design, in a vein much similar to [5]. This could be helpful in the management of architectural knowledge on a global scale.

Finally, we plan to integrate quality or process-improvement metrics in software engineering (e.g. CMMI) with metrics and approaches to improve

the development community jointly with its process. These mechanisms could be beneficial to software project management.

Acknowledgment

The authors would like to thank Dr. Ossi Korhonen and Dr. Minna Raisanen for providing empirical evidence in the form of well-defined reports, for us to elaborate the results in this paper. We acknowledge the contribution of all employees in Company X that offered their time to confirm the results contained in this paper.

References

[1] D. A. Tamburri, P. Lago, H. Van Vliet, Uncovering latent social communities in software development, IEEE Software (2012) 1–7.

[2] J. Keyes, Social software engineering :, Taylor & Francis, Auerbach Series, Boca Raton, FL :, 2011.

[3] R. Sangwan, M. Bass, N. Mullick, D. J. Paulish, J. Kazmeier, Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series), Auerbach Publications, Boston, MA, USA, 2006.

[4] J. Herbsleb, A. Mockus, An empirical study of speed and communication in globally distributed software development, IEEE Transactions on Software Engineering 29 (6) (2003) 481–94.

[5] N. Nagappan, B. Murphy, V. Basili, The influence of organizational structure on software quality: an empirical case study, in: International conference on Software engineering, IEEE, Leipzig, Germany, 2008, pp. 521–530.
URL `http://doi.acm.org/10.1145/1368088.1368160`

[6] D. A. Tamburri, E. di Nitto, P. Lago, H. van Vliet, On the nature of the GSE organizational social structure: an empirical study, proceedings of the 7th IEEE International Conference on Global Software Engineering (2012) 1–10.

[7] M. E. J. Newman, Networks: an introduction, Oxford University Press, Oxford; New York, 2010.
URL http://www.amazon.com/Networks-An-Introduction-Mark-Newman/dp/0199206651/ref=sr_1_5?ie=UTF8&qid=1352896678&sr=8-5&keywords=complex+networks

[8] J. C. van Vliet, Software engineering - principles and practice., Wiley, 1993.

[9] L. Northrop, P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, K. Wallnau, Ultra-Large-Scale Systems - The Software Challenge of the Future, Tech. rep., Software Engineering Institute, Carnegie Mellon (June 2006).
URL http://www.sei.cmu.edu/uls/downloads.html

[10] I. Kwan, A. Schroter, D. Damian, Does socio-technical congruence have an effect on software build success? a study of coordination in a software project, IEEE Trans. Softw. Eng. 37 (3) (2011) 307–324. doi:10.1109/TSE.2011.29.
URL http://dx.doi.org/10.1109/TSE.2011.29

[11] E. Davenport, H. Hall, New knowledge and micro-level online organization: 'communities of practice' as a development framework, in: System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on DOI -, 2001, pp. 10 pp.–.

[12] D. A. Tamburri, P. Lago, H. van Vliet, Organizational social structures for software engineering, To appear on ACM Computing Surveys, 2012, pp. 1–35.

[13] J. J. Cusick, A. Prasad, A practical management and engineering approach to offshore collaboration., IEEE Software 23 (5) (2006) 20–29.
URL http://dblp.uni-trier.de/db/journals/software/software23.html#CusickP06

[14] M. Cataldo, S. Nambiar, On the relationship between process maturity and geographic distribution: an empirical analysis of their impact on software quality, in: Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, ESEC/FSE '09, ACM,

New York, NY, USA, 2009, pp. 101–110. doi:10.1145/1595696.1595714.
URL http://doi.acm.org/10.1145/1595696.1595714

[15] D. A. Tamburri, Going global with agile service networks, International Conference on Software Engineering.

[16] A. M. R. Correia, A. Paulos, A. Mesquita, Virtual communities of practice: Investigating motivations and constraints in the processes of knowledge creation and transfer., Electronic Journal of Knowledge Management 8 (1) (2010) 11–20.

[17] S. Gallagher, Introduction: The arts and sciences of the situated body, Janus Head 9 (2) (2006) 1–.

[18] n. Ruikar, n. Koskela, n. Sexton, Communities of practice in construction case study organisations: Questions and insights, Construction Innovation 9 (4) (2009) 434–.
URL http://proquest.umi.com/pqdweb?did=1920022811&Fmt=7&clientId=4574&RQT=309&VName=PQD

[19] R. Cross, J. Liedtka, L. Weiss, A practical guide to social networks, Harvard Business Review (2005) –.

[20] I. Ruuska, M. Vartiainen, Communities and other social structures for knowledge sharing: A case study in an internet consultancy company, Communities and Technologies (2003) 163–183.
URL http://portal.acm.org/citation.cfm?id=966263.966272

[21] J. Erik Andriessen, Archetypes of knowledge communities, Communities and Technologies 2005 (2005) 191–213.
URL http://dx.doi.org/10.1007/1-4020-3591-8_11

[22] J. Corbin, A. Strauss, Grounded theory research: Procedures, canons, and evaluative criteria, Qualitative Sociology 13 (1) (1990) 3–21.

[23] L. A. Goodman, Snowball sampling, The Annals of Mathematical Statistics 32 (1) (1961) pp. 148–170.
URL http://www.jstor.org/stable/2237615

[24] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2) (2009) 131–164.

[25] C. D. Rosso, Comprehend and analyze knowledge networks to improve software evolution, Journal of Software Maintenance and Evolution: Research and Practice 21 (3) (2009) 189–215.
URL http://dx.doi.org/10.1002/smr.408

[26] E. Hustad, Exploring knowledge work practices and evolution in distributed networks of practice., Electronic Journal of Knowledge Management 8 (1) (2010) 69–78.
URL http://search.ebscohost.com/login.aspx?direct=true&db=lxh&AN=48217974&site=ehost-live

[27] R. Stake, The Art of Case Study Research, Sage Publications Inc, 1995.
URL http://books.google.com/books?hl=no&lr=&id=ApGdBx76b9kC&oi=fnd&pg=PR11&dq=case+study&ots=KrNKk2Fnbo&sig=tBVSrcFVI4tzDG3AzErMhF__xe4

[28] R. El-Haddadeh, V. Weerakkody, J. Peng, Social networking services adoption in corporate communication: the case of china., J. Enterprise Inf. Management 25 (6) (2012) 559–575.
URL http://dblp.uni-trier.de/db/journals/jeim/jeim25.html#El-HaddadehWP12

[29] S. Mohammadi, A. Khalili, S. Ashoori, Using an enterprise mashup infrastructure for just-in-time management of situational projects, in: e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on, 2009, pp. 3 –10. doi:10.1109/ICEBE.2009.11.

[30] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering: an introduction, Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[31] M. E. Conway, How do committees invent, Datamation 14 (4) (1968) 28–31.