Title:
Networking in a Large-Scale Distributed Agile Project

Author:
Nils Brede Moe, Darja Šmite, Aivars Š blis, Anne-Lie Börjesson, Pia Andréasson

Conference Name:
International Symposium on Empirical Software Engineering and Measurement

Conference Year:
2014

Conference Location:
Turin

Networking in a Large-Scale Distributed Agile Project

Nils Brede Moe
SINTEF | Blekinge Institute of Technology
Norway | Sweden
nils.b.moe@sintef.no

Darja Šmite
Blekinge Institute of Technology
Sweden
darja.smite@bth.se

Aivars Šāblis
Blekinge Institute of Technology
Sweden
aivars.sablis@gmail.com

Anne-Lie Börjesson
Ericsson AB
Sweden
anne-lie.borjesson@ericsson.com

Pia Andréasson
Ericsson AB
Sweden
pia.andreasson@ericsson.com

ABSTRACT

Context: In large-scale distributed software projects the expertise may be scattered across multiple locations.

Goal: We describe and discuss a large-scale distributed agile project at Ericsson, a multinational telecommunications company headquartered in Sweden. The project is distributed across four development locations (one in Sweden, one in Korea and two in China) and employs 17 teams. In such a large scale environment the challenge is to have as few dependences between teams as possible, which is one reason why Ericsson introduced cross-functional feature teams – teams that are capable of taking the full responsibility for implementing one entire feature. To support such teams when solving problems, ensure knowledge sharing within the project and safeguard the quality Ericsson introduced a new role – Technical Area Responsible (TAR).

Method: We conducted extensive fieldwork for 9 months at two Ericsson sites in Sweden and China. We interviewed representatives from different roles in the organization, in addition to focus groups and a survey with seven teams.

Results: We describe the TAR role, and how the TARs communicate, coordinate and support the teams. Also architects support the teams, however not as closely as the TARs. We found that the TAR is usually a senior developer working halftime or fulltime in the role. We also present measures of the actual knowledge network of three Chinese and three Swedish teams and the TARs position in it.

Conclusions: TARs are central in the knowledge network and act as the boundary spanners between the teams and between the sites. We learned that availability of the TARs across sites is lower than that with local TARs. We also found that the size of a team's knowledge network depends on how long the team members have been working in the company. Finally we discuss the advantages and the challenges of introducing experts in key roles in large scale distributed agile development.

ESEM'14, September 18–19, 2014, Torino, Italy.

Copyright 2014 ACM 978-1-4503-2774-9/14/09...$15.00

Categories and Subject Descriptors

K.6.1 [**Management of computing and information systems**]: Project and People Management – *Staffing, Systems development*

K.6.3 [**Management of computing and information systems**]: Software Management – *Software development, Software process*

General Terms

Management, Measurement, Human Factors

Keywords

Boundary spanners, global software development, offshoring, insourcing, social network analysis, distributed agile, large-scale agile, technical area responsibility.

1. INTRODUCTION

Modern software organizations working on large-scale evolvable systems in a distributed context are exposed to a rapid increase in complexity related to a growing number of customers, increasing complexity of software products, and increasing size of the distributed development organization. Ericsson is such an organization, and this paper reports empirical evidence from organizing the development process of a project that is part of a product consisting of 10-12 million lines of code, 30 subsystems partly developed in Sweden and China.

Existing research suggests that continuous software evolution requires access to an enormous amount of knowledge and skills [3-5]. The term "access" refers to receiving a valuable piece of knowledge and knowing how to use it. A software developer and a software team need to rely on knowledge resources embedded within, available through, and derived from a network of relationships, also known as social capital. Social capital is both the network and the assets that may be mobilized through that network [1]. In other words, "who you know" affects "what you know".

The importance of social networks grows when a single developer or even a development team in a large evolving system development project does not possess the expertise needed in their daily work. The expertise needed in complex systems development includes technical skills (programming languages, and methodologies), domain knowledge (how are the customers using the product and the features), and also the knowledge of the product being developed or maintained (its source code structure and architecture). Expertise that software developers and teams accumulate grows significantly and becomes even more critical as the products evolve [4]. When changes occur on the organizational level (e.g. introducing new team members, adding new teams, assigning new tasks), or on more technical level (adding new dependencies, components, or performing major refactoring) effective access to people's knowledge and information becomes crucial. In distributed development environment the needed expertise may be scattered across multiple locations, which adds to the complexity of the network of relationships.

For Ericsson to succeed in a large-scale distributed setup high-performing teams are crucial. One step towards performance increase was transition to feature teams. Since such teams are working in all parts of the code, they often need to have access to the knowledge outside of their own team. To understand the role of the knowledge networks at Ericsson we will investigate the social networks of the teams in the distributed large-scale project, the position of experts in these networks and the impact of distribution.

The rest of the paper is organized as follows. In Section 2 we present the background. In Section 3 the project under study is described in more detail. In Section 4 we briefly introduce the methodology of the study. Section 5 contains our findings, while Section 6 concludes the paper.

2. BACKGROUND

2.1 Introducing Cross-Functional Feature Teams at Ericsson

Ericsson is a large international company headquartered in Sweden. The company relies on agile methods (Scrum), and develops a wide range of products and solutions, including generic software products offered to an open market and complex compound systems with customised versions. Due to the growing complexity of their software products more resources were needed. However it was problematic to get access to enough resources in Sweden. At the same time there was a need for reducing development cost and to be closer to important markets, therefore Ericsson has distributed development of a number of software products across its sites in different locations. In this paper we describe experiences from two sites of Ericsson, one in Sweden and one in China.

In 2007 the Swedish site of Ericsson realized they had too little competence in certain areas of the product reported in this paper, and experienced problems implementing new features in the large-scale system. The strategy then was to move from competence-based teams only developing part of a feature, to cross-functional feature-based teams. As one manager explained: *"Teams should broaden their competence so they can work on all necessary subsystems"*. The organization then invested in building teams that could implement a whole feature.

Earlier Ericsson had node architects responsible for entire subsystems. However, after growing the number of teams and introducing cross-functional feature teams working in all parts of the code, the need for access to expertise about certain type of functionality supported from several subsystems emerged. In response to this, the role of a Technical Area Responsible (TAR) was introduced.

2.2 Introducing the TAR role at Ericsson

The TAR is responsible for supporting teams and answering technical questions about their subsystems. They help the teams to get the right design and code structure. A TAR should also act as a mentor for teams that need to do changes in the TAR's area when the teams have little experience in that area. Additionally TARs are responsible for reviewing and prioritizing trouble reports. TARs are essential for the cross-functional teams to operate, ensure the quality and evolvability of the system. Notably, TARs are not architects. This role is assigned to the most skilled and senior developers. More details about the TARs can be found in the next section.

The subsystem reported in this paper is developed both in Sweden and in China. In the beginning all senior developers that could be assigned responsibility for a technical area were located in Sweden, because the Chinese site was yet immature. This caused problems. One manager said: *"The load got so heavy on those four TARs, we had to appoint persons in China to be co-TARs... they started to learn more about the subsystem and could take responsibility of reviewing…"* In 2011, TARs were introduced in the Chinese site.

3. PROJECT AND PROCESS OVERVIEW

The investigated subsystem is distributed across 4 sites – Sweden, China (Ericsson site) and China (consultant site), and Korea.

Teams. The number of developers working on the subsystem grew from 8 developers in 2007 to 30 developers in 2009, and scaled up to around 60 developers by 2013. In early 2014 the subsystem was developed by 5 teams in Sweden, 8 in-house teams in China, 2 Chinese homeshore consultant teams and 2 Korean teams. Ericsson relies on cross-functional teams (XFT), which are teams that have all core competences needed for the development and release of a feature. At Ericsson the XFT includes roles like system manager, system designer, function tester, system tester, and sometimes a TAR working parttime in the team. In addition, each XFT has a scrum master, and collaborates with an Operative Product Owner (OPO). Other roles, such as architects, system owners, design owners, product owners, configuration management personnel, integration leaders, line managers, agile coaches, etc. support the teams. Each of these XFTs consists of 5-9 team members. Team members might have several roles in their team, and some team members are associated with several units in different roles. The teams are usually situated in an open plan office. The focus of our investigation is on the cooperation between the teams at the two Ericsson sites in Sweden and China that develop one specific subsystem.

Features. Different feature assignments have different domain specificity. This puts certain demands on the knowledge in the team or accessible for the team. While technical knowledge is said to be easy to learn, it takes years to learn the domain. Teams are assigned features that can require implementation of changes in different parts of the system. Although teams are working on features that can require changes in the same code, the dependencies on both subsystem and task levels are minimized by implementing plans that depict dependencies. The features are allocated to teams based on competence and previous experience. Noteworthy, very large features may be assigned to several teams. In that case, physical proximity between the teams is also taken into consideration. Besides new feature development, teams are involved in fixing bug reports.

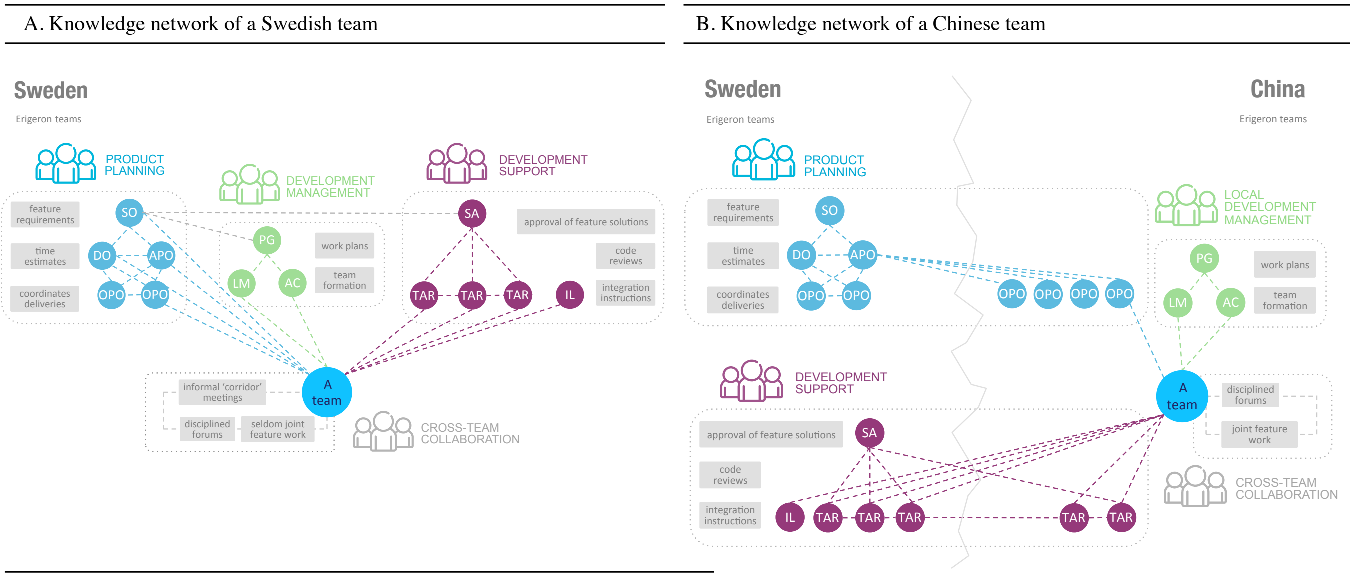
TAR. The Technical Area Responsible (TAR) is the contact person for technical discussions from teams to protect teams from intrerrupts. The TAR usually works 50% in a team and 50% as a TAR. He or she support the teams with design proposals, code and design reviews. In particular, the TARs support teams in early phases if needed, and mentor teams when they are working in a new area. The TARs also give input to OPOs regarding product improvements, and help OPOs to make early investigations of possible work packages for teams. TARs at Ericsson have the following mandates:

- Prioritization of trouble reports;
- Responsibility for guidelines (coding, quality assurance, trouble reports, deliveries);
- Responsibility for internal interfaces towards other subsystems;
- Mandate to reject insufficient design proposals;
- Deciding mandate if conflicting opinions regarding the code structure emerge after code reviews;
- Responsibility for code reviews;
- Responsibility for raising quality issues/improvements to the OPOs and providing suggestions on actions.

Sprint zero. The teams have end-to-end responsibility for a feature. Sprint zero is a phase, in which the team develops a common understanding of the feature requirement and proposes a concrete solution, which is further reviewed by the stakeholders. This is the stage where the team really commits to develop the feature. Therefore all team members are needed with their various viewpoints.

From the system level to the subsystem level. Since the system grew over time and accounts tens of subsystems, the product ownership and governance became more hierarchical as well. System Owner (SO) is one person who has the overall responsibility for the system and its quality. This person overlooks all subsystems in the product and guides the system's evolution. The system owner is tightly integrated in the development network and uses the expertise from the senior developers and technically skilled personnel when taking critical decisions for the product. The system architects (SA), are responsible for the system architecture. Further the product ownership is divided upon the areas. There is an Area Product Owner (APO) per subsystem. The APO works closely together with the OPOs (Operative Product Owners), forming a hierarchy. APOs are responsible for defining what to implement in overview perspective, while OPOs are an essential part of the teams' social networks. It is worth mentioning that there are other roles supporting the teams in continuous integration and different levels of testing. However, these are not a part of this paper, since our focus is on the task allocation and development.

Notably, distance matters when it comes to coordination. The studied system's owner confessed *"There is always difficulty with distributed sites when it comes to feedback. In many cases it is always easier to go and talk to people here in the same time zone."* Because of cross-site coordination challenges, a few other roles were implemented in close proximity to the teams in each location (discussed in the next section, see also Fig. 1).


A. Knowledge network of a Swedish team

B. Knowledge network of a Chinese team

Abbreviations

PG – program manager
LM – line manager
AC – agile coach

DO – design owner
SO – system owner
APO – area product owner
OPO – operative product owner

IL – integration leader
SA – system architect
TAR – technical area responsible

Fig. 1 – Swedish and Chinese teams' connections to different roles

4. BASIS FOR THE OBSERVATIONS

4.1 Data Collection

Our investigation started with a set of initial interviews that aimed at a better understanding of the implementation of the distributed cross-functional feature teams and the TAR role. We interviewed representatives from different roles at both Ericsson sites in Sweden and China.

After initial interviews we organized focus groups with development teams from both sites. The sampling strategy was guided by the Ericsson representatives (fourth and fifth authors of the paper) and included selecting new as well as mature teams in both sites. During the focus group we acquired about the presence of the skills needed for solving team's tasks, teamwork practices, interaction with other teams, roles and communities; usefulness of the externalized knowledge; teams' reliance on different types of knowledge and skills in their daily work; and team's perception of their performance.

We also conducted a survey with the seven teams participating in the focus groups. The survey was web-based for the Swedish site and paper printouts handed out by the first and second author at the Chinese site. We asked respondents to identify in "free-recall" format project-related knowledge contacts that the respondents pull the knowledge from or transfer to. Respondents where also asked to describe the knowledge received and transferred, as well as evaluate the frequency of knowledge-sharing and accessibility of their contacts (using a 5-point Likert scale). Our survey is a partial replication of a survey conducted by Manteli et al. [2]. However, we extend it by obtaining a directed knowledge network and acquiring detailed information about each knowledge-sharing relation. In contrast to Manteli et al., who applied the survey on the project and unit level, we trace our observations to the team level.

In total, 40 people from 7 teams completed the questionnaire. However, one of the teams was removed from the analysis due to a very low response rate (33%). The response rate for the remaining 6 teams in total was 90% and for individual teams did not fall below 71%. The final sample consisted of 35 project members and 336 dyadic ties in the social network.

After the survey, we asked the respective team responsible managers (two in China and one in Sweden) to evaluate each team and it´s performance.

4.2 Data Analysis

The interview and focus group data was used to profile each of the teams and the role of TARs, and later to find explanations from the observations based on the social networks. The survey data was analyzed using basic statistics and visualized with the help of open-source social network visualization software called Gephi (https://gephi.github.io).

In this paper, we first visualize detailed social networks of three different teams, and connect our observations with the characteristics of the teams. Next we explore the role of TARs by visualizing the focused inter-team network that includes the TARs and excludes other roles. The same network is used to explore the inter-team collaboration. Finally, TAR availability is calculated based on the values provided by the respondents and supplemented by our findings from the interviews and focus groups.

5. RESULTS

In this section we present our results that focus on the role of the social capital of the teams and knowledge networks in the globally distributed organization.

5.1 The social network of teams

When analyzing the social network we tried to gain an understanding why certain teams are performing better than the others. This is why our analysis focused on comparing the differences between high-performing and low-performing teams and between newly formed teams and highly experienced teams. In the following we exemplify our findings using two high-performing teams (one from Sweden and one from China) and one team from China that has performance challenges.

The Swedish team S3 in Fig. 2 is classified as a high-performing team. The team is constantly given new types of features to solve, features that require the team to work in several subsystems at the same time. The figure illustrates how the team members (colored nodes) are connected with other roles (grey nodes). Closeness of each of these connections outside the team indicates the frequency of collaboration. Those that collaborate daily are the closest. Working on high complex tasks involving many subsystems is probably the main reason for why this team reports using input from a lot of people outside of their team.

Additionally, we noted a few other characteristics that seemed to be important. This team has been stable for 2-3 years and consists of team members that have on average around 10 years of experience in Ericsson. Being in Ericsson for a long while they probably know whom to get input from.

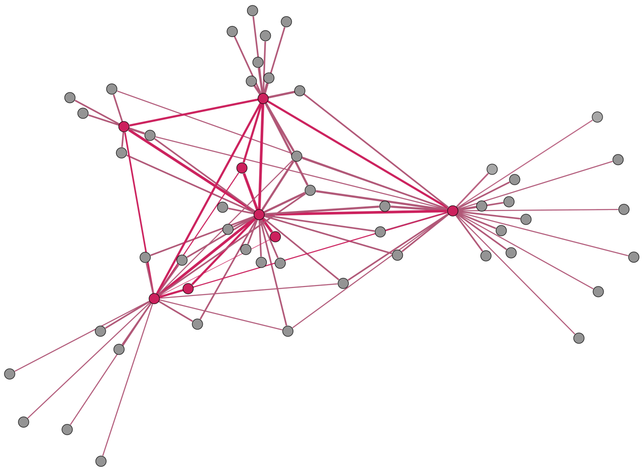


Fig. 2 – Social network of a S3, a well-performing Swedish team

Fig. 3 illustrates a well-performing Chinese team. In contrast to the Swedish team (in Fig. 2), the Chinese team's social network is smaller. They have knowledge exchange with less people outside their team and the contact is less frequent. When comparing the team characteristics, we notice that although the team is working together for about 3 years, in contrast to the Swedish team, the team members have lower experience and affiliation with Ericsson. From this we hypothesize that the differences in knowledge connections outside the team could be explained by the experience gathered by individuals in an organization. The social network of an individual developer grows over the years, and is used irrespective of the affiliation with a specific team.

Another plausible explanation of the difference between the two teams could be connected to the distribution of the project. The Swedish teams are collocated with a lot of experts, as illustrated in Fig. 1, while the number of supporting roles present in the Chinese site is considerably lower.

One final explanation could be related to the differences in the types of tasks and culture. It could be that the types of tasks solved by the Chinese team requires less interaction with others, or that the culture in the Chinese site encourages less interaction (e.g. asking for help can be related to loosing face or it might be seen as disturbing others). However, we have not collected any data that supports these hypotheses.

During the focus group we learned, the Chinese team seems to put a lot of effort into strengthening the team internal network. They constantly try to improve teamwork, each day they go out and eat lunch together and every month they engage in team-building activities out of their personal budget. Notably, the Swedish teams did not report on such activities.

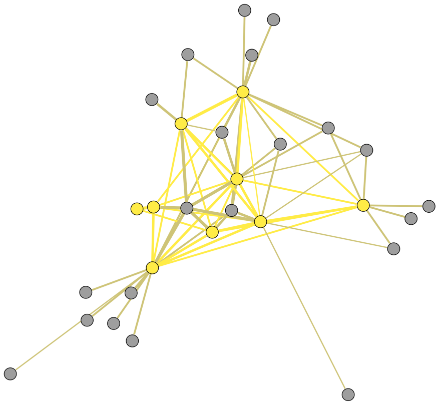


Fig. 3 – Social network of C3, a well-performing Chinese team

To contrast our results we have also looked at a network of a newly formed Chinese team (see Fig. 4), which at the moment of investigation naturally is not performing very well. Notably, social network analysis illustrates a completely different picture. The team has fewer connections and consequently interacts with fewer experts. Since they have a lot of newcomers they naturally have a smaller network. Also they are given easier tasks because they have been newly formed.

This example is illustrative to highlight the problem of newly formed teams. They have less experience and fewer contacts and thus knowledge connections outside the team. Which again affects their performance. In other words, team members need to be connected with those who possess the knowledge. Obviously, if not formally managed such connections take a considerable amount of time to develop. This means that the roles such as of TARs could be especially important for new teams. In fact, in Fig. 5 we see that the team C1 interacts closely with the TARs especially with the local TAR in China but also with one of the Swedish TARs, which suggests that the formal knowledge-sharing mechanism established at Ericsson through TARs is being exploited.

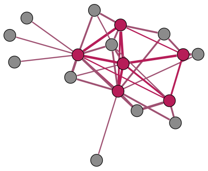


Fig. 4 – Social network of C1, a newly established Chinese team

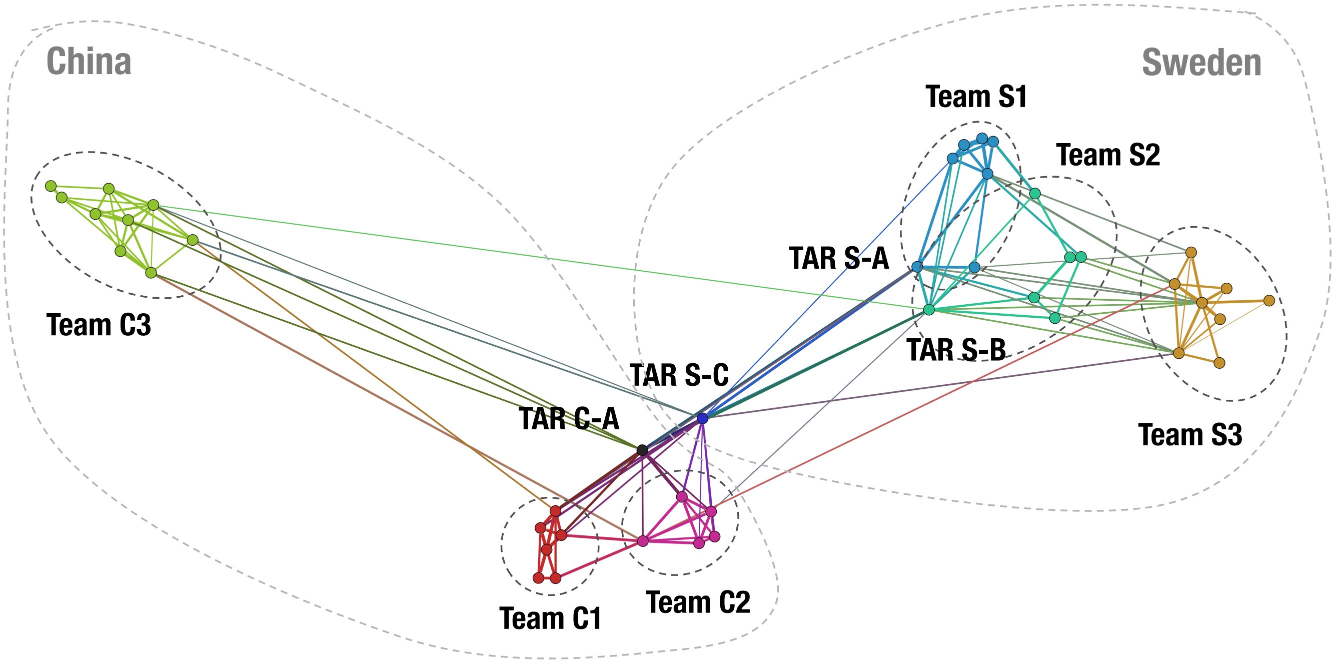


Fig. 5 – TARs in the social network of the six teams

5.2 TARs in the Social Network

When we integrate all the teams and the TARs in the same social network (see Fig. 5), we can see that the TARs occupy a very central role by having connections to all teams. First we notice that the sites are distinctly clustered. We then see that the four TARs (three in Sweden and one in China) are positioned in the center. However their position is different. We found that the Swedish teams connect only to the Swedish TARs, while Chinese teams interact primarily with the local TAR and one of the Swedish TARs, who unexpectedly appears in the Chinese site's cluster, meaning that this person collaborates mostly with people at the remote site. These two TARs have an overlapping technical area responsibility, i.e. domain.

Which TAR the team collaborates with also depends on the nature of feature tasks received by the teams in the two locations. Certain tasks require interaction with specific TARs, and not other TARs who are responsible for other areas.

We continue our analysis by exploring the role of each individual TAR in the social network, and highlight the three teams discussed in Section 5.1 (see Table 1). We see that the Chinese teams have more connections to the TARs than the Swedish teams (22 over 15 connections for all six teams). The level of experience in the Chinese teams can explain the higher number of connections. The difference in the number of connections is also visible when comparing the number of connections to TARs from the teams (S3 – a well-performing Swedish team, C3 – a well-performing Chinese team, and C1 – a newly formed Chinese team). The newly formed team with performance challenges has the highest number of connections to TARs.

Table 1: TARs in the social network

TARs	Total connections*			Team S3	Team C3	Team C1
	With six teams	Locally	Remotely			
C-A	10	10	0	0	3	3
S-A	6	5	1	2	0	1
S-B	10	8	2	2	1	0
S-C	11	2	9	1	2	3
Total:	**37**	**22 with Chinese teams** **15 with Swedish teams**		**5**	**6**	**7**

* We have excluded connections between the TARs from the statistics.

5.3 Availability of TARs

We further explore the availability of TARs as perceived by the survey respondents. In this analysis we are especially interested in understanding the cross-site connections. We asked our respondents to state whether they agree that a TAR is available and received the scores on the 5-point Likert scale (1 – strongly disagree, 2 – disagree, 3 – neutral, 4 – agree, 5 – strongly agree).

From Table 2 we can clearly see that the Swedish TARs are perceived as less accessible to the Chinese team compared to the local Chinese TAR. One explanation that we have heard during the focus groups is related to the time difference and language barriers. The Swedish TARs are available only a few hours every day, and thus communication abilities are limited. The questions asked by the Chinese teams might get delayed, especially if further clarifications are needed.

Another explanation is that the Swedish TARs are only partially working as TARs, being allocated to a team as developers 50%, while the Chinese TAR at the moment of our investigation was dedicated to work as a TAR 100%. This could explain the lowest total availability score of the TAR S-C, who has the highest number of connections, most of which being remote (as seen in Table 1).

Table 2: TAR availability scores in the social network

TARs	Perceived by all six teams	Perceived by local teams	Perceived by remote teams
C-A	4.1	4.1	–
S-A	4.3	4.8	2
S-B	4.2	4.5	3
S-C	3.2	4.5	2.9

* We have excluded connections between the TARs from the statistics.

5.4 Inter-team collaboration

When comparing the inter-team collaboration (see Fig. 5) it is evident that the Swedish teams are more integrated with each other than the Chinese teams. By integrated we mean that the team members are connected with team members from other teams. In fact, the social network shows that the Chinese teams are distinctly clustered, while the boundaries of the Swedish team clusters are positioned closely and partially overlap (see S1 and S2 teams). This means that the Swedish teams get input from each other, while there is less interaction between the teams in China. One possible explanation for this could be related to the TAR role implementation in each site. TARs are a part of a team in Sweden. However, there is a number of connections beyond those among and with the TARs. When we removed the TAR connections and visualize the resulting networks, it was evident that the remaining Swedish team members communicated a lot.

Another possible explanation is related to the affiliation with the company. The Swedish developers, as discussed in Section 5.1 have a longer history of working at Ericsson and with each other (most of them stayed at Ericsson for about 10 years). Team constellations change and we assume that some of the developers keep their previous team members close in the social network. In contrast, the Chinese teams employ developers relatively new to the company, which might be the reason for a low number of connections among the teams.

Another interesting finding emerges from comparing the sites. The highest ranked Chinese team, C3, operates more isolated than the other teams, while team S3 (a well performing Swedish team) seems to be more integrated with the other teams. We assume that it could be explained by the nature of assigned tasks. Team S3 is constantly challenged. New tasks outside of the area of expertise of the team require getting input from other teams and experts when solving tasks and coordinating work. At the same time, C3 is a well performing team with a relatively stable task profile, thus the team gains less from interacting with other, perhaps, even less performing teams. However, one would expect that the less performing teams should have a greater need to cooperate with well performing teams, which is not seen from the emerging network.

During the focus groups we also learned that both Swedish and Chinese teams have similar types of forums (such as communities of practice), including a testing forum, an integration forum, a developers' forum, and a scrum masters' forum. However, the Chinese teams confessed that they seldom had the time to join the forums. Lack of networking through these forums also results in not knowing who knows what in other teams. This could be another explanation for why there is less interaction between the Chinese teams compared to the Swedish teams.

6. LESSONS LEARNED

From studying the role of the TAR, the teams and their social networks at Ericsson we found that there is a difference in the social networks and the role of the TAR between the sites and the teams. We believe that these differences can be the result of the culture, the number of years of experience in the organization, the nature of the tasks assigned to a team, and the geographic and time differences between the sites that influence remote collaboration. Our lessons learned are thus summarized in the following:

- Large-scale software development projects are often associated with inability for everybody to know everything. When the knowledge is not present in the team, connections become evidently important. "Who you know" affects "what you know" or in other words the amount of knowledge available for solving the tasks. The need for establishing knowledge retrieval ties is also important for newly employed individuals, newly formed teams and teams working with tasks outside of the current area of expertise.
- Formal mechanisms for knowledge access are helpful. This is especially true when the tasks demand diverse knowledge that is not always present in the team. The importance of formal mechanisms emerges especially in the sites with high employee turnover rates or relatively new sites, such as the Chinese site in our case. Accumulating the social capital and establishing connections takes considerable time. Therefore newly employed team members can only rely on the experts in formally assigned roles. Another formal mechanism is documentation, i.e. organizational capital. However, all teams reported that documentation was less important than the social network, partly because the documentation was not always updated and was often scattered across multiple databases.
- Alternatively, the newly formed teams could be assigned less challenging work that does not require combination of knowledge from different domains or areas of expertise. As such, the teams might not need to interact with too many experts. A practical implication of this is that organizations need to consider gradually increasing the complexity of the tasks assigned to teams formed from recently employed developers.
- Despite many beliefs that experts are emergent, formally assigned TARs in Ericsson appeared to occupy the central position in the network of a large-scale distributed agile project. This might be because the TAR role is assigned to the most experienced and well-respected developers. Due to their central position we also see them as the bridges for cross-team and cross-site knowledge and information exchange, while the feature teams themselves work in relative isolation from each other. This is especially important for establishing joint quality standards, and coordinating the common ways of working across a large-scale organization.
- Implementing the TAR role is evidently not an easy task. Availability scores suggest that some TARs might be overloaded. Chinese and Swedish sites address this situation differently. The Chinese site has chosen to create a fulltime position for a TAR, while the Swedish site allocates the TARs halftime to development in a team, and halftime as the TAR. This is done because the TARs are usually the most senior developers, and their contribution is important in the development too. And also because the involvement in the actual development might have a positive impact on maintaining the high level of expertise.
- Communities of practice and different disciplined forums encourage networking among members of different teams and thus create opportunities for learning who knows what. At Ericsson networking is commonly organized within one site. We learned that Swedish teams network more often than the Chinese teams, who claimed not having enough time for attending the forums. We thus emphasize that implementing communities of practice requires management support and organizational culture that supports networking.

7. CONCLUSIONS

In this paper we described practical experiences from a large-scale distributed agile project at Ericsson that employs feature teams in Sweden and China. Such teams require a lot of diverse knowledge that often takes years to accumulate. This is why Ericsson has implemented a formal knowledge sharing mechanism – the role of a Technical Area Responsible (TAR), in addition to the System Architects (SA), a System Owner (SO), a Product Owner (PO) and general System Managers. Notably, TARs are not architects. This role is commonly assigned to the most experienced senior developers who are well-respected in the network.

To understand the TAR role at Ericsson we investigated a social network of six teams in two locations. We have found that TARs are central in the teams' network and often act as the boundary spanners both between the teams and more importantly between the sites. This is interesting, since the common belief suggests that central experts are emergent and not formally assigned. We also learned that the availability of TARs across sites is lower than that with local TARs. We also found that the size of a team's knowledge network depends on how long the team members have been working in the company. This is why when scaling, forming new teams, as well as in the new sites or sites with high employee turnover, as often are the offshore sites, the role of a TAR is crucial. Finally, we believe that the TAR role could be found useful in similar large-scale projects employing cross-functional feature teams outside Ericsson in general and in other projects at Ericsson in particular (the TAR role is not a widely used practice).

As the future work we plan to perform detailed data analysis. We will first add other subsystem TARs in the network, and explore their importance, frequency of communication and availability. We will also add the remaining teams into the detailed analysis. We plan to calculate measures related to network clusters and cores, and performing analysis of the knowledge flows by studying each incoming (knowledge retrieval) and outgoing (knowledge transfer) connections between the dyads in the network. We hope to gain a better understanding of what knowledge is required from outside of the teams, and connect this to the factors that already emerged from our preliminary analysis – levels of experience and affiliation with the company.

8. ACKNOWLEDGMENTS

This research is supported by Ericsson AB and the Swedish Knowledge Foundation under the KK-Hög grant 2012/0200, and by the Smiglo project and the Research Council of Norway under the grant 235359/O30.

9. REFERENCES

[1] Bourdieu, P. The forms of capital, *Handbook of Theory and Research for the Sociology of Education*, Richardson, J., Ed. New York: Greenwood, 1986.

[2] Manteli, C., van den Hooff, B., and van Vliet, H. The Effect of Governance on Global Software Development: An Empirical Research in Transactive Memory Systems. *Information and Software Technology*, available online, 2014

[3] Parnas, D.L. Software aging, in *Proceedings of the 16th international conference on Software engineering*. Sorrento, Italy: IEEE Computer Society Press, 1994, pp. 279-287.

[4] Rajlich V.T., Bennett, K.H. A staged model for the software life cycle, *Computer*, 2000, pp. 66-71.

[5] van de Laar, P., Douglas, A.U., America, P. Researching Evolvability, *Views on Evolvability of Embedded Systems:* Springer Verlag, 2010, pp. 1.