Managing Knowledge in Global Software Development Projects

Torgeir Dingsøyr, SINTEF, Norwegian University of Science and Technology, Norway
Darja Šmite, Blekinge Institute of Technology, Sweden

How should knowledge be managed in global software development projects? To answer this question, the authors draw on established software engineering research and study three focus groups in two global companies, discussing which knowledge management approaches are appropriate.

Software is increasingly being developed by global teams.\(^1,2\) In companies of all sizes, projects are being set up across several development sites, separated by distance, time zones, and cultural differences. In the past, companies focused on outsourcing certain software projects to low-cost countries. Today, companies often instead choose to establish their own sites in different countries, enabling access to sufficient personnel and technology knowledge while ensuring the necessary control over staff turnover and protection of intellectual capital.\(^3\) Recent advancements in version-control systems and the availability of both low-cost and high-end communication technology have made global collaboration easier. However, many companies are still striving to increase the effectiveness of global software development projects.

As Pär Ågerfalk and Brian Fitzgerald\(^4\) describe, global software development projects are challenging, owing to a lack of informal interactions between team members, which can reduce their awareness of others’ work tasks. Furthermore, cultural differences can create misunderstandings, and inconsistent work practices can impinge effective coordination.

Knowledge management,\(^5\) which is “a method that simplifies the process of sharing, distributing, creating, capturing, and understanding of a company’s knowledge,”\(^6\) helps ensure the development of a product that has the right features and a sufficient level of quality. Studies of
teamwork show how shared knowledge improves team effectiveness, because sharing knowledge about the development process and what’s to be developed helps teams avoid costly misunderstandings. Also, having team members who can perform other members’ tasks during times of high workloads helps ensure continued progress, because all available resources can be used on high-priority tasks. Furthermore, a study of software development performance shows how better integration of domain and technical knowledge leads to increased software development effectiveness and efficiency.¹

Here, we examine strategies for knowledge management and, through a focus group study of three global collaborations, we describe current practices to address the question of how knowledge should be managed in global software development projects.

**Knowledge Management Approaches**

Researchers have been paying closer attention to knowledge management in global software development projects. For example, Kevin Desouza and his colleagues have argued for new methods and models as companies increase global development,⁸ and Ilan Oshri and his colleagues have described practical methods for managing knowledge in global projects.⁹ Although the number of documented empirical studies devoted to knowledge management in global software engineering is small,¹⁰ a review of knowledge management in software engineering in general revealed 29 empirical studies,¹¹ so we wondered if we could apply some of the lessons learned from general knowledge management to the particular setting of global software development.

As Ågerfalk and Fitzgerald point out, there are three particular challenges in global software development: temporal, geographic, and sociocultural distances.⁴ In the following, we briefly present findings from the general studies of knowledge management in software engineering and discuss how they might apply to global software development. We use Michael Earl’s framework of knowledge management schools.¹² This widely used framework, which has been applied in software engineering,¹¹ divides knowledge management into five approaches.

The systems school focuses on applying technology for knowledge management by storing knowledge in repositories. Studies in software engineering found that repositories have been used to share different types of knowledge, such as risk assessment and software design experience. One study shows broad application of an easy-to-use repository.¹³ Because knowledge is codified, this approach works well with geographic and temporal distance. However, there might be sociocultural challenges—for example, different preferences regarding which knowledge to maintain, how to communicate the knowledge, and how much detail should be described.

The cartographic school focuses on knowledge maps and creating knowledge directories. A study of a skills management system found that this tool was in use for a variety of purposes, from allocating resources, to searching for competence and identifying project opportunities, to upgrading skills.¹⁴ The tool enabled learning both at the individual and company level. Knowledge maps and directories on company intranets could be beneficial when the company is distributed geographically. However, this is most effective when the temporal distance is small and knowledge can be transferred orally. This school might also be vulnerable to sociocultural distance—for example, there might be variations in terms of what kind of knowledge is mapped in a skills management system and how the skill levels are evaluated.

The engineering school focuses on processes and knowledge flows in organizations. A number of studies in software engineering describe this approach, focusing primarily on processes for mapping knowledge, conducting project retrospectives, mentoring programs, and describing work processes, such as in the Capability Maturity Model.¹⁵ This approach relies on explicit knowledge and thus isn’t affected by geographic or temporal distance. However, knowledge about processes might be interpreted differently in diverse sociocultural settings.

The organizational school focuses on networks for sharing or pooling knowledge. Many companies have applied this approach by establishing internal communities of practice—that is, groups who interact regularly to share knowledge about a common topic of interest. The systematic review¹¹ refers to one study, which claims that communities built on existing networks are more likely to succeed. In a community, both
tacit and explicit knowledge can be exchanged, but typically, the explicit knowledge exchange is less formal than in a knowledge repository. Lessons learned reports and templates are typical examples. Knowledge is usually communicated orally in physical or virtual meetings. Thus, this school can suffer from challenges related both to geographic, temporal, and sociocultural distance.

The spatial school focuses on how the design of an office space can facilitate knowledge management. This can range from setting up whiteboards close to water coolers or coffee machines to making use of open-plan offices. A popular approach in agile development is to establish taskboards, which provide visible information regarding the project status to team members and other stakeholders during formal and informal meetings. The systematic review didn’t identify any studies of this knowledge management approach in software engineering. Notably, this approach depends on physical colocation, and from studies of agile software development, it seems to work well for small teams.

Thus, when applying such knowledge management approaches in global settings, temporal and geographic distances affect the ability to access and share knowledge, while sociocultural differences introduce challenges in terms of aligning how knowledge is shared and maintained (see Table 1). We would expect traditional global projects to rely on systems and engineering schools, because codified information sharing is less vulnerable given the distances. Agile development, on the contrary, implies the dominance of spatial and organizational schools, with its focus on sharing tacit knowledge. The cartographic school can provide a cost-effective means for managing knowledge globally for both traditional and agile projects.

**Knowledge Sharing: Focus Groups**

To explore how knowledge is shared in global projects, we organized three focus group workshops in two organizations. Both organizations are large international companies developing complex embedded software solutions using product-oriented processes.

“Alpha” is headquartered in Sweden and is rapidly extending its operations into Asia. We held two workshops focusing on collaborations involving sites in Sweden, China, and India. “Beta” is headquartered in the US. Our study focused on collaborations involving sites in Sweden and Russia. Organizations were selected on the basis of accessibility and their interest in taking part in this research. Both companies represent typical global companies in terms of their knowledge management—neither had an explicit initiative related to knowledge management.

Alpha had a mature agile development environment, while Beta had recently started using agile development. The time difference between the sites was seven hours in workshop 1 and 3.5 hours in workshop 2. Participants from workshop 3 had fully overlapping work hours, after an adjustment in workshop 1. We used the focus group research method to capture employee perceptions, opinions, beliefs, and attitudes regarding knowledge management. This method helped us quickly obtain information on emerging phenomena through structured, moderated discussions with groups of practitioners. In a workshop lasting four hours, we explored knowledge management strategies, challenges, and potential improvements, asking the following key questions:

- What knowledge is important for efficient completion of daily work?

---

**Table 1. Knowledge management approaches and their application in global projects.**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Focus</th>
<th>Aim</th>
<th>Potential challenges in global software development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems</td>
<td>Technology</td>
<td>Knowledge bases</td>
<td>Sociocultural differences</td>
</tr>
<tr>
<td>Cartographic</td>
<td>Maps</td>
<td>Knowledge directories</td>
<td>Sociocultural differences</td>
</tr>
<tr>
<td>Engineering</td>
<td>Processes</td>
<td>Knowledge flows</td>
<td>Sociocultural differences</td>
</tr>
<tr>
<td>Organizational</td>
<td>Networks</td>
<td>Knowledge pooling</td>
<td>Geographic and temporal distance</td>
</tr>
<tr>
<td>Spatial</td>
<td>Space</td>
<td>Knowledge exchange</td>
<td>Geographic and temporal distance</td>
</tr>
</tbody>
</table>

---

*January/February 2014*
Which knowledge resides locally, and which is shared globally?
How is knowledge shared and maintained?
What knowledge is easy to share, and what introduces challenges?

The moderator instructed the participants, who engaged in the workshop activities for individual brainstorms and group discussions. In the colocated workshops, we used whiteboards and flip charts to structure results. Participants used post-it notes and markers to document answers to questions posed by the moderator.

In the video-conference workshop involving two sites, one of the moderators documented the results using a mind-map software tool. The results were shown on a shared screen to all participants. Meeting minutes were sent to all participants for validation, and then the description of knowledge management practices from the minutes was coded into knowledge management schools—for example, “use of knowledge repositories” was coded as the systems school. The results were presented to the companies for verification and feedback.

Global Knowledge Management
Many knowledge management schools were in use locally and globally, as shown in Figure 1. Beta seemed to struggle with managing important knowledge both globally and locally, while Alpha was able to easily manage most knowledge.

One possible explanation for this difference is related to the nature of work conducted in each case. In Alpha, we explored two collaborations in which the product components shared a platform. The majority of the work was conducted solely in one location, and only the shared platform part required joint coordination. In Beta, we explored an offshore site, which was represented by participants from different projects. Most of these projects involved maintaining systems previously developed in Sweden. This could affect the complexity of the necessary knowledge management.

Here, we present findings from each workshop in relation to each school (see Figure 1).

Systems School
In all three workshops, participants described extensive use of knowledge repositories available in corporate databases, intranets, and local project file servers. Figure 1 shows that this approach was mainly used to manage knowledge globally, and the two most mature agile projects viewed this knowledge as easy to manage, while the project in Beta found much of this knowledge difficult to manage.

A common challenge with this school is that repositories easily become information graveyards, where knowledge is stored but not retrieved. Also, codifying knowledge is usually more expensive than transferring it orally. Participants from all three projects complained about the search functions in the repositories. During Alpha’s first
workshop, we identified a lack of awareness of how to search effectively. During Beta’s workshop, we learned that some knowledge repositories contained limited knowledge but were easy to search, while others contained much more information but were difficult to search.

The challenge in this approach is thus primarily related to how the knowledge repository is developed, which supports the theoretical argument that this school is robust to temporal and geographic distance. Sociocultural differences didn’t seem to influence use of the systems school. This might be because the sites had a long history of collaboration—the systems school might be more problematic for new projects.

We also observed that progress and future plans weren’t reported across the sites. Participants of the first Alpha workshop explained that each site managed the knowledge locally in its own repository, making the knowledge inaccessible for remote colleagues.

**Cartographic School**

Interestingly, we found only one example of the cartographic school. It was a globally maintained wiki page containing indexed links to project information sources in one of the projects in Beta. An overview of “what is where” in global projects is important, because each site might use local knowledge repositories or many repositories for different types of knowledge.

Gaps in understanding “who knows what” were stressed in all three workshops. Although colocated team members knew each other well, familiarity with colleagues from other teams (in large projects) and from remote sites was challenging. Formal meetings usually provided individuals with knowledge about where they could obtain further knowledge. However, if such meetings were held only locally, the cross-site links weren’t created. The knowledge of who has decision power was missing in both projects in Alpha.

Although cartographic knowledge management solutions seem easy in theory, they require commitment and joint effort from all collaborating parties and thus are difficult to put into practice. There’s a great need to introduce easy-to-use cartographic knowledge management strategies in global projects, which could help increase awareness and improve coordination.
Engineering School
Distance introduces coordination challenges, but defining and standardizing work processes across sites can help, as outlined in the engineering school. This school was implemented as a global effort in Alpha, while Beta had a mixture of locally and globally supported processes. Such knowledge was mainly viewed as easy to manage, but from the Alpha projects, we learned that the ways of working differ across sites, and even within a site, when a large project is divided into teams. This corresponds with previous findings regarding the challenges of standardizing work practices through role and process descriptions on intranets.18

Some processes were missing, thus prohibiting knowledge sharing across project teams. In Alpha, feedback from team retrospectives was stored in a corporate repository but not shared across teams. Only a few participants in Sweden were aware of a process improvement initiative, which targeted cross-team learning, presenting their findings once a year. Learning from each other was even more challenging across remote sites. Furthermore, some processes were hard to describe. In Alpha, we learned that sharing knowledge about the notion of good versus bad code quality and test quality was challenging due to sociocultural differences.

Organizational School
This approach to knowledge management develops communities, where members keep in touch through video conferences, physical meetings, and telephone and email conversations. These communities can be within and across sites. Most of the knowledge managed through these means was local in Alpha, and evenly local and global in Beta. The Alpha participants of the second workshop found this knowledge to be the most hard to manage, while it was moderately hard in Beta.

With this school, much of the product and project knowledge remains tacit and is accumulated in the heads of experts. Distance introduces new challenges for sharing this knowledge. While knowledge locally is shared through personal contact networks and during formal and informal meetings, reliance on an organizational knowledge management school in global projects becomes more challenging.

Participants from the Alpha project explained that certain knowledge—such as code anatomy, product architecture, implementation, and dependencies—wasn’t shared globally. This could explain why the knowledge managed with organizational school approaches wasn’t seen as hard in this project. Live meetings necessary for sharing such information were challenged by time-zone differences. In Beta, developing a cross-site community required “bridgeheads,” a person staying at the other site to get acquainted with the remote organization and learn from experts or a person from the other site who can facilitate this. Participants from Beta admitted that it was difficult to create a community with remote colleagues. Building personal relationships without face-to-face interaction takes a long time.

Spatial School
The Alpha projects and one of the projects in Beta used taskboards to visually communicate the team’s work status. The knowledge managed through this approach was seen as easy to share, especially because it was maintained locally. We haven’t found sufficient evidence that this school could work effectively over long distances.

In Alpha, the teams that use agile methods locally share knowledge on taskboards and innovation boards to capture ideas that are unavailable across sites. One project in Beta recently started using a video camera to show a taskboard to the other site; however, it’s unlikely that a temporary video-feed of the board will create the same form of awareness of tasks. In fact, the local team in Beta was also forced to maintain the board in a separate meeting room instead of having it close to the team’s working location. Although advanced tools for video-walls ensuring real-time projections from remote offices are available, none were applied in the companies studied.

This study shows that knowledge management is challenging both at local and global levels, and companies need to focus on both. Rather than selecting one knowledge management school, companies should carefully consider several. Approaches can be perceived as easy or hard to implement, depending on the context. Developers working on global projects need sufficient information to create a shared knowledge of the task and team and to increase task awareness, thereby improving project coordination.19
In particular, companies looking to manage global software projects should consider the following.

First, identify the main global challenges in the project. Are there barriers for knowledge management due to geographic, temporal, or sociocultural distance?

Second, define what should be shared locally and globally. Discuss different strategies for managing knowledge in the project as early as possible. Knowledge sharing needs can vary between different sites, so focus group workshops (such as the ones used for this study) are useful in facilitating an awareness of the challenges.

Third, for local knowledge management, use approaches that require fewer resources first, such as the spatial, organizational, and cartographic schools. Use visual boards, build on existing networks, and establish easy-to-use overviews of employee knowledge. Finding the right balance of local and global knowledge management can be difficult, as our focus groups showed. For global knowledge management, select an approach after evaluating the challenges inherent in your global software development project. Remember that cross-site collaboration must be fostered over time. This can be done through collaborative workshops on team tasks but also supported through indexes and profile pages that help remote teams learn who knows what and where to find information. This offers an inexpensive way of sharing knowledge globally.

Fourth, avoid known knowledge management pitfalls. Companies often rely on codified knowledge and process standardization efforts in global projects. We recommend that companies learn from successful cases in the systems school when using knowledge repositories. Avoid repositories that create impediments to knowledge sharing or that are only used in certain sites or not used at all. We also stress exploiting approaches that promote sharing tacit knowledge, unless there’s a large potential for reusing explicit knowledge.

Finally, apply strategies to improve work practice as identified in studies in the engineering school. General findings and our focus group study indicate that describing work methods on an intranet won’t lead to changes in work processes. 

Acknowledgments

We are grateful to Claes Wohlin at the Blekinge Institute of Technology and Tore Dybå and Tor Erlend Fagri at SINTEF for comments on an earlier version of this article.

References


Torgeir Dingsøyr is a senior scientist at SINTEF Information and Communication Technology and an adjunct associate professor in the Department of Computer and Information Science, Norwegian University of Science and Technology. His research interests include knowledge management, software process improvement and agile development methods. Dingsøyr received his doctoral degree in software engineering from the Norwegian University of Science and Technology. Contact him at torgeir.dingsoyr@sintef.no.

Darja Smite is an associate professor in Blekinge Institute of Technology in Sweden and a professor at University of Latvia. Her research interests include various aspects of global software engineering, as well as agile software development. Smite received her doctoral degree in computer science from University of Latvia and her habilitation in software engineering from Blekinge Institute of Technology. Contact her at darja.smite@bth.se.