Darja Šmite, Blekinge Institute of Technology

Marco Kuhrmann, University of Southern Denmark

Patrick Keil, Keil KTM GmbH

TODAY, WORKING with colleagues in different locations is more common than not, from globally distributed software projects to R&D projects to manifold business processes in companies across all industry sectors. Global software projects, however, have a mixed reputation. Proponents suggest that globalization enables a variety of benefits, such as cost reduction, shorter time to market, access to a skilled labor pool, and increased innovation.[1] Opponents and skeptics, on the other hand, call global software development "the crisis of the decade,"[2] warning that the assumed benefits aren't for everyone and shouldn't be taken for granted.[3]

To understand what makes some global projects synch and others sink, it's essential to emphasize that global projects have different flavors. Not all of them are distributed, for example—nor do they all employ virtual teams. The distinction is in team formation. We might refer to distributed projects as virtual projects because the teams working on them are assembled from people residing in different locations. But these projects might instead have loosely coupled teams in different locations with collocated members. Virtual teams, on the other hand, demonstrate a high level of interdependence and cooperation among team members. Complicating the situation somewhat, globalization can be enabled by offshore outsourcing (subcontracting work to a third party) or offshore insourcing (working with subsidiaries, also known as captive offshoring). This could determine, for example, the amount of control project managers have over teams or team members in remote locations. Figure 1

0740-7459/14/$31.00 © 2014 IEEE

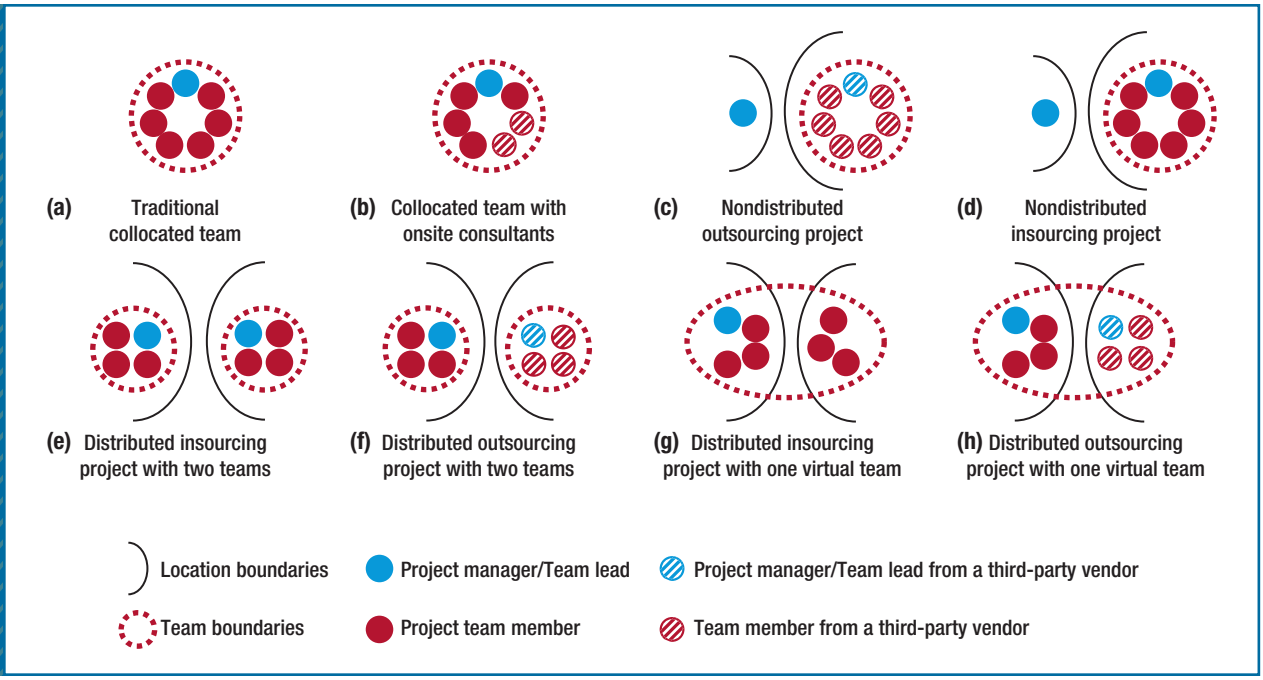

FIGURE 1. Different project setups. Projects with (a) traditional collocated teams, (b) collocated teams with onsite consultants, (c) nondistributed outsourcing projects, and (d) nondistributed insourcing projects aren't distributed, whereas projects with (e) distributed insourcing with two teams, (f) distributed outsourcing with two teams, (g) distributed insourcing with one virtual team, and (h) distributed outsourcing with one virtual team are considered distributed project arrangements. The projects described in (e) and (f) rely on loosely coupled teams, whereas (g) and (h) have dispersed or virtual teams.

illustrates different setups that contrast the teamwork found in collocated, distributed, and dispersed or virtual arrangements.[4]

Globally Dispersed Teams

Given the complexity of global software development as a phenomenon, we chose to focus this special issue on one particular type of global projects, one that uses globally dispersed or virtual teams. A virtual team is a group "of geographically, organizationally and/or time dispersed workers brought together by information and telecommunication technologies to accomplish one or more organizational tasks."[5] Virtual teams differ from traditional collocated teams in that their members remain interdependent when working on their tasks.[6] Virtual teams can operate as a permanent structure as well as on a temporary basis, when team members with specific expertise assemble to accomplish specific deliverables or to fulfill specific customer needs.

Research on virtual teamwork isn't new, but most of it concerns teams in general (studied by sociologists) or in the context of information systems, which can include marketing teams, teams of users and project stakeholders, management teams, requirements engineering teams, maintenance teams, and, among others, software development teams. In 2004, Anne Powell and her colleagues published a comprehensive overview of the state of the art in virtual IS teams.[5] They found that the research available at that time couldn't determine the types of projects suited for virtual teams, the appropriate size and skills composition for virtual teams facing different project types, efficient design characteristics of successful virtual teams, ideal team structures (self-organizing versus managed teams), or the role of managers (virtual team leadership). Moreover, Powell and her colleagues found that more than 90 percent of the research articles they reviewed were based on student projects in controlled environments.[7]

Ten years later, many of the questions that Powell and her colleagues viewed as relevant then are still urgent today, especially for virtual

software teams. Although plenty of empirical research has been conducted on global software development in general—and project management and teaming in particular—contributions that focus on virtual teams in software engineering are scarce. For instance, in a tertiary study on global software development conducted by June M. Verner and her colleagues, virtual teams as a means to organize distributed projects aren't distinguished or even mentioned.[8] Yet the topic of virtual teams continues to evolve in IS research.[9]

Expert Opinions

Deficiency in research, however, doesn't reflect the level of industrial relevance. To set the scene for the special issue, we solicited expert opinions from several industry practitioners working in software development companies or in development groups in industry companies. We inquired whether virtual software development teams are practiced and preferred over loosely coupled teams, and what it takes to succeed with virtual teamwork in software projects.

Sameer Deans, from ThoughtWorks in South Africa, is a clear proponent of virtual teamwork. In his commentary, he recommends that companies move from old-fashioned, disciplined teams residing in different locations to more cross-functional virtual teams, in which development, testing, and system management competence is spread across all locations of a globally distributed project.

Our second expert is Lars-Ola Damm, from Ericsson in Sweden. In his commentary, he discusses the pros and cons of relying on cross-functional virtual teams versus loosely coupled component-based teams, and gives a context-dependent recommendation. Unlike the experience at ThoughtWorks, where integration of disciplined knowledge is crucial, the main concern at Ericsson is related to the integration of knowledge from different functional areas of a system. Large-scale software projects seem to call for a more modularized approach, thus our expert argues against virtual teams. However, he agrees that when the architecture is unstable or has unclear

VIRTUAL TEAMS HELP REDUCE HANDOVERS AND SILOS

SAMEER DEANS, THOUGHTWORKS

Based on what I've seen, I feel the strategy that can work for most teams with team members in multiple locations is to try and have a truly distributed team. This means having every location perform as many team functions as possible, instead of having all the testing or analysis done in one place and the development in another. Such division leads to silos, which affect team dynamics and performance.

It might not be possible with every single position at ThoughtWorks, but as far as it's feasible, we try to plan for a full complement of roles in each of our locations. In essence, we've become a cross-functional feature team with team members from each location operating as a virtual team. Making the virtual-team strategy work has prerequisites:

- Try to kick off with a face-to-face initiation and planning workshop. This helps set context for the team and the project.
- While large-scale travel might not be financially or personally feasible, having key team members meet at regular intervals helps bring the team together from both a cultural and context-sharing point of view. Teams that don't meet their business representatives or product owners often suffer from misalignment with business goals, so it's critical to arrange frequent meetings with these folks.
- Finally, nothing beats seeing your teammates, albeit virtually. Most organizations that employ virtual teams use collaboration tools such as phones, instant messaging, and conference lines, but adding a video link makes the communication more visible and personal. This added dimension helps overcome the impersonal aspect of remote teams.

With more than 17 years of industry experience, Sameer Deans has worked with all kinds of teams, including distributed ones. His experience is based on 16 projects, which are diverse in size (7 to 80 developers per project, and up to 30 developers on one team) and in geography (partners based in the US, UK, Brazil, Hong Kong, India, and Australia). His position at ThoughtWorks is strongly related to running agile software development, also with virtual teams.

IS IT REALLY A GOOD IDEA TO WORK IN VIRTUAL TEAMS?

LARS-OLA DAMM, ERICSSON

Practical experience from different formations of large-scale distributed development has convinced me that distributed development of any kind inevitably results in a significant overhead. In particular, it has a negative impact on communication, especially in inadequate or delayed communication between development teams. The largest impact for a company with tough market demands is in my experience on the time to market, especially if there is a time-zone difference involved. Just consider this scenario: the average response time to a query between developers across locations is one day instead of one hour. Any of those queries that require a response before development can have a direct negative impact on the time to market. The impact becomes even greater if there are, for example, cultural differences or site competition forces that hinder good cooperation across locations.

So why are companies still doing it? In my experience, it's most often a consequence of where the available engineers happen to be located, for example, in low-cost countries or where acquired companies have development. Some might wonder, then, what exactly is the preferred setup for distributed competence?

I think it's largely a trade-off between two options: having virtual teams where each group suffers from communication overhead within the team but not so much between teams, or having collocated subsystem teams where the communication overhead is between teams. The virtual option has a strong advantage in that it can work for an agile cross-functional team taking end-to-end responsibility on a customer feature. However, in a large-scale environment, it's hard for such a team to have enough knowledge to handle the code changes needed across the whole system. I've experienced this with teams who complain that they "just barely knew a little bit about everything and couldn't be good at anything," which both increased development time and decreased quality since more mistakes were made.

So, then, how do you make this trade-off? At least part of the answer comes from determining if customer requirements and the system architecture are stable. The more unstable, the harder it is to define clear boundaries between subsystems and thereby divide the work between teams. In this scenario, a more cross-functional setup is preferable. A middle-ground approach is to have collocated teams developing, testing, and releasing their own subsystems, but when a customer feature requires more communication across the system, temporary cross-functional virtual teams are created to develop that feature. Another approach that's gaining more momentum at Ericsson is to apply open source practices when developing large distributed systems; that is, teams that have responsibility for one subsystem can contribute with patches to components that other teams are responsible for, utilizing the power of open source review and submission rules to collaborate with each other more effectively than through documentation. It doesn't solve the problem, but it decreases the level of overhead.

Lars-Ola Damm received a PhD in software engineering and has worked with large-scale and distributed software development at Ericsson for 13 years. He currently manages several agile development teams that are part of the development of a telecom system involving about 1,000 people distributed across eight locations on three continents. In this system, most development teams are collocated cross-functional teams focusing on one subsystem, but in some areas, virtual teams have been formed as well.

boundaries, a cross-component virtual teamwork is preferred.

Our third expert, Marco Nock from Knorr-Bremse in Germany, heads a department consisting of different virtual project engineering and product lifecycle support teams. His staff works in different locations across the globe on product development. He emphasizes the challenges of creating a team spirit and avoiding destructive competition between locations. According to Nock, when projects have multiple teams, virtual teamwork has certain advantages compared to a network of loosely coupled teams:

- It's easier to create a common spirit and a shared goal across locations.
- The network of virtual teams is more stable, which increases constancy.
- Know-how is shared and transferred across locations as a part of teamwork that reduces risk,

especially for long-term projects.
- The risks regarding immoderate competition or blockade situations are lower.

Note that all three experts in one way or another underline the importance of specialization in each location. They agree that this creates value through knowledge transfer and provides the needed feelings of contribution and ownership.

In This Issue

In addition to the commentaries from industry, we also called for research contributions. We targeted submissions that could unite research originality with industrial relevance. In particular, we wanted to know which aspects must be considered in virtual teamwork, such as team setup, management and control, tool support for team coordination, information sharing, communication, and team motivation, by addressing cultural differences and other human factors. Despite the popularity of global software development, soliciting submissions dedicated to virtual software teams appeared to be a challenge. We received 16 submissions in total, and after a thorough peer review process, selected two articles for the special issue. This again highlights the scarcity of research on virtual software teams and the importance of fostering research in this area.

The two articles presented here meet our goals of providing insights into challenges and opportunities of virtual teams, and, furthermore, lay the foundation for further investigation.

In "Collaboration Spaces for Virtual Software Teams," Kevin Dullemond, Ben van Gameren, and Rini van Solingen illustrate the importance

VIRTUAL TEAMS HAVE BOTH PREREQUISITES AND BENEFITS

MARCO NOCK, KNORR-BREMSE SYSTEME FÜR SCHIENENFAHRZEUGE

My experience from over 10 years of engineering and software development projects, as well as academic and R&D projects, shows that, despite the manifold merits of modern communication and collaboration tools, the efficiency and effectiveness of virtual teams is still highly dependent on trust and team spirit. In virtual teams, each and every member needs to work with team members in other locations as intensely as with colleagues in the same room. Therefore, it's important not to underestimate the importance of fostering the power of a shared vision and the desire to reach a goal together. Any success of any virtual team should be celebrated together, independent of who had the lead, the initial idea, or the biggest workload.

At the same time, it's important that each location has the responsibility for some well-defined topics, modules, or products (depending on the business you're in) such that they never feel that they're in a "far off" location that doesn't have anything to be proud of. This might sound like a contradiction—shared vision versus individual responsibilities—but it's crucial to seek the organization's equilibration.

The ultimate goal in setting up a virtual team is to choose team members and allocate tasks in a way that individual competencies and regional as well as cultural competences amplify each other.

From a management perspective, it is important to monitor the performance of a virtual team and to identify blocking points early enough. Key performance indicators shall create the required transparency to manage virtual teams properly.

Virtual teams form dense networks. Compared to loosely coupled teams, escalation paths can be established more easily and are more transparent. Also, virtual teams have advantages regarding interfaces to and contacts with your customers, especially if they also have multiple locations worldwide. What I would call "working networks" require regular workshops or longer stays at other locations to create personal relationships. Only then can the differences in basic skills, mindsets, or cultural background transform from risks into opportunities for a project.

Marco Nock received a PhD in mechanical engineering from Universität Erlangen-Nürnberg and has had many years of management experience in the transportation industry. At Knorr-Bremse, a manufacturer of braking and onboard systems for rail vehicles, he heads the rail services engineering department, covering the development of products and integrated systems for the service business, advanced and systems engineering for modernization projects, and product care worldwide.

ABOUT THE AUTHORS

DARJA ŠMITE is an associate professor at the Blekinge Institute of Technology, where she leads the research efforts related to the effects of offshoring for Swedish software companies. Her other research interests are related to global software engineering, large-scale agile software development, and software process improvement. Šmite is also affiliated with the University of Latvia, where she holds a visiting professor position in the area of software engineering. Contact her at darja.smite@bth.se.

MARCO KUHRMANN is an associate professor at the University of Southern Denmark, where he works on software process and lifecycle management. He has served on international programs and organized several international academic conferences and workshops. In addition to his academic work, Kuhrmann participates in various industry projects regarding process improvement, tools, and methods in different distributed software development settings. Contact him at kuhrmann@mmmi.sdu.dk.

PATRICK KEIL is managing director of Keil KTM GmbH, a consulting and services company focusing on software development and application maintenance. His research interests include research and industry projects on offshoring and distributed development, IT trend research, and process improvement. Keil received a diploma in economics from Ludwig-Maximilians-Universität München. Contact him at patrick.keil@keil-ktm.com.

of virtual collaboration spaces to simplify and support daily teamwork. Based on their empirical studies in two software companies, they conclude that tailored virtual-team support can ensure a better collaboration than even a collocated setting.

In "Onboarding in Open Source Projects," Fabian Fagerholm, Alejandro Sanchez Guinea, Jay Borenstein, and Jürgen Münch focus on open source software (OSS) projects as representatives of virtual teamwork. Open source practices have recently become a subject of increasing interest for commercial use in software companies. The authors of this article investigate the process of onboarding new developers into OSS projects and present experiences and findings based on collaborations between Facebook and the University of Helsinki.

Over the past decades, today, and in the future, business contexts in software organizations and the common ways of developing software are changing dramatically. Formation of teams in distributed environments, virtual or not, calls for new ways of working across geographic, temporal, and cultural boundaries. This, however, also requires effective leadership approaches enabled through systems, processes, technology, and people.[10] We hope that this special issue provides some ideas and strategies for practitioners and open questions for researchers.

References

1. R. Sangwan, N. Mullick, and D.J. Paulish, *Global Software Development Handbook*, Auerbach Publishers, 2006.
2. D. Parnas, "Agile Methods and GSD: The Wrong Solution to an Old but Real Problem," *Comm. ACM*, vol. 49, no. 10, 2006, pp. 26–34.
3. Ó Conchúir et al., "Global Software Development: Where Are the Benefits?," *Comm. ACM*, vol. 52, no. 8, 2009, pp. 127–131.
4. D. Šmite, "Distributed Project Management: Ten Misconceptions That Might Kill Your Distributed Project," *Software Project Management in a Changing World*, G. Ruhe and C. Wohlin, eds., Springer Verlag, 2014, pp. 301–320.
5. A. Powell, G. Piccoli, and B. Ives, "Virtual Teams: A Review of Current Literature and Directions for Future Research," *The Database for Advances in Information Systems*, vol. 35, no. 1, 2004, pp. 6–36.
6. V. Casey, "Developing Trust in Virtual Software Development Teams," *J. Theoretical and Applied Electronic Commerce Research*, vol. 5, no. 2, 2010, pp. 41–58.
7. H.K. Edwards and V. Sridhar, "Analysis of the Effectiveness of Global Virtual Teams in Software Engineering Projects," *Proc. 36th Hawaii Int'l Conf. System Sciences*, 2003; www.computer.org/csdl/proceedings/hicss/2003/1874/01/187410019b.pdf.
8. J.M. Verner et al., "Systematic Literature Reviews in Global Software Development: A Tertiary Study," *Proc. 16th Int'l Conf. Evaluation & Assessment in Software Eng.* (IET 12), 2012, pp. 2–11.
9. A.A. Ebrahim, S. Ahmed, and Z. Taha, "Virtual Teams: A Literature Review," *Australian J. Basic and Applied Sciences*, vol. 3, no. 3, 2009, pp. 2653–2669.
10. L.A. Hambley, T.A. O'Neil, and T.J.B. Kline, "Virtual Team Leadership: The Effects of Leadership Styles and Communication Medium on Team Interaction Styles and Outcomes," *Organisation Behaviour and Human Decision Processes*, vol. 103, 2007, pp. 1–20.

See www.computer.org/software-multimedia for multimedia content related to this article.