

# Daedalus or Icarus?

## Experiences on Follow-The-Sun

Jacob Krüger<sup>\*‡</sup>, Stephan Dassow<sup>†</sup>, Karl-Albert Bebber<sup>§</sup>, and Thomas Leich<sup>\*†</sup>

<sup>\*</sup> Harz University of Applied Sciences, Wernigerode, Germany

{jkrueger | tleich}@hs-harz.de

<sup>†</sup> METOP GmbH, Magdeburg, Germany

{stephan.dassow | thomas.leich}@metop.de

<sup>‡</sup> Otto-von-Guericke-University, Magdeburg, Germany

{jkrueger}@ovgu.de

<sup>§</sup> Consiliantibus GmbH, Cologne, Germany

{karl-albert.bebber}@consiliantibus.com

**Abstract**—Follow-the-sun is an approach to develop software by handing off the progress to different time zones as the day passes. Hence, this approach allows companies to work on a project 24 hours a day, potentially reducing its time-to-market. However, several challenges, such as time zone differences or handing off work, are often reported. In this paper, we describe a case study on a follow-the-sun approach that was applied in a German company. During the approach, we did rarely face the aforementioned challenges but experienced different ones. For this reason, the company put its approach on hold but benefited from the learned lessons. Overall, we report on a partly successful follow-the-sun approach and identify five important practices.

**Index Terms**—follow-the-sun; globally acting teams; distributed software engineering; global software engineering; experience report

### I. INTRODUCTION

Global software development promises several benefits, such as reducing costs, utilizing time-zones, or connecting a larger skill pool [7]. *Follow-the-sun* is one approach towards global software development and proposes to hand off work after each workday from one location to another in a different time zone [4, 5, 15]. The main objective of this approach is to reduce the time-to-market for products to achieve a competitive advantage. However, successful implementations are rarely reported [4, 5] due to a variety of challenges in globally distributed software development [7, 8, 12, 14]. Kroll et al. [15] conducted a systematic literature review [18] on follow-the-sun and summarized reported problems, the most prominent being:

- Managing time zone differences
- Handing off work daily
- Overcoming cultural diversity in communication
- Missing synchronous communication
- Resolving cultural and educational differences

In this paper, we describe a case study on a follow-the-sun approach that did not fail for such reasons. We participated as consultants for planning and assessing the approach at a globally acting company listed in the German stock index (DAX). Due to the company's international background, some

issues were solved previously, for instance suitable communication techniques and time-zone management. Still, we faced challenges that are related to other research areas but rarely reported in follow-the-sun research. While the company was able to cope with some of these challenges, others were not to tackle.

More precisely, we contribute the following:

- We report the case study, its implementation, and solved problems. This can help companies to define solutions for their approaches towards global software development in the future.
- We identify and discuss open issues we found during the study, focusing on those that the company did not resolve (i.e., trust, communication, participation, motivation, and tooling). Hence, we raise the awareness among the research community and industry in anticipating problems.

The remaining article is structured as follows. In Section II, we describe the nature of the case study, the company's approach on follow-the-sun, and share our insights. This leads us to the lessons learned in Section III. Finally, we conclude in Section IV.

### II. THE APPROACH

The approach was planned and carried out in an internationally operating German company, which will not be named due to confidentiality. This company's main business is not software engineering, however, it develops and maintains several software systems. Previously, the software division implemented a customizable and globally used information system, in addition to software for different products. Within the company, operational processes, consulting, software development, and maintenance are globally distributed and aligned to national regulations. However, all employees used the aforementioned information system. Thus, we categorize processes integrated into this information system as *core*, which were identical everywhere, or *aligned*, which were customized to a region or nation.

To improve efficiency of software development and maintenance, the company considered to introduce a follow-the-

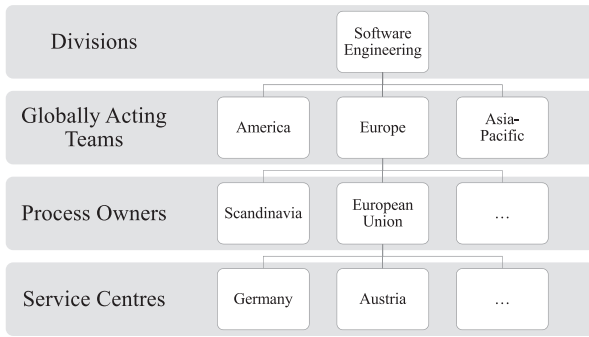


Fig. 1. Conceptual structure of business units in the software engineering division.

sun approach. The main idea was to consolidate software engineering in the centralized system to enable software maintenance and development 24 hours a day. Based on standardized processes the company aimed to reduce response times for customer tickets (i.e., customer requests and bug reports) and achieve faster time-to market for new products on a higher quality (i.e., by applying the two-man rule on team level). A project with several partners had been initiated to implement and assess the follow-the-sun approach.

In the following, we describe the company’s approach towards follow-the-sun. Therefore, we distinguish between an *initial phase*, in which the approach was scoped and implemented, and an *adoption phase*, in which identified problems were addressed.

#### A. Initial Phase

Due to its international background, the company had solved some well-known challenges of global software development [8, 12, 14, 15]. For instance, the employees were familiar in cooperating among different time zones and English was established as default language. The company’s policy, especially on staff education, and previously gained experiences resulted in only few cultural or educational differences. Nevertheless, the company needed suitable infrastructures and processes based on the existing information system. Both were part of the approach’s initial phase.

As a starting point, the company decided to use its German branch’s system. Aligned processes from all over the world were integrated to provide a basis for communication, information, and artefacts. Hence, the approach based on the company’s pre-existing infrastructure and did not follow a specific method, such as FTSProc [10].

While integrating processes into the information system was straightforward, the global distribution and management had to be adopted significantly. Previously, each company abroad had an own business unit for software engineering, resulting in a diversity of processes and development styles. To harmonize processes and enable the follow-the-sun approach, the software engineering division was considerably restructured. We exemplify the resulting structure in Figure 1.

Firstly, the management consolidated software development and maintenance by geographical regions (i.e., in Europe,

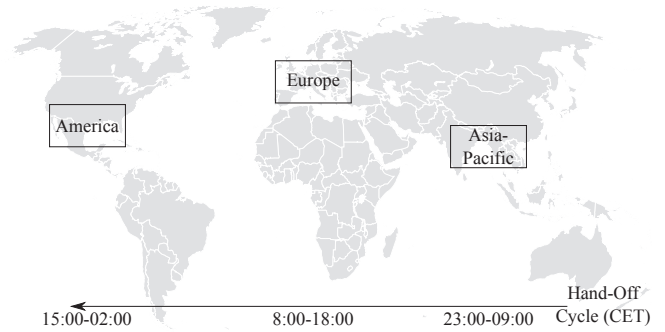


Fig. 2. Localization of globally acting teams.

America, and Asia-Pacific) and located them in suitable areas, as we illustrate in Figure 2. To this point, the company virtually connected distributed teams [7, 8, 9, 13] in these areas to form larger *globally acting teams* in each region. These globally acting teams contributed in maintenance and development projects, especially with urgent deadlines. For each of such projects, an entry in the information system was created, containing corresponding tickets, artifacts, and documents. Any team handed off a project by defining its final status and providing all changes, which represented the starting point for the next team. The workdays between teams overlapped for up to three (see Figure 2) hours, allowing them to communicate and discuss problems.

Secondly, process owners were responsible for aligned processes and located in the corresponding region. For example, they adapted development processes in company branches to regulations of the European Union or Scandinavia. This method was inspired by agile software development [6, 11], which is proposed as a best practice for follow-the-sun approaches [15]. Process owners had, mainly, to coordinate, review, and release changes on their processes.

Finally, the company transformed legacy business units in different countries into service centres. Those handled minor incidents, for instance with workarounds and data correction, allowing them to react faster to customer tickets. Service centres transferred software development or maintenance requests they could not handle to the corresponding globally acting team or process owner. Hence, they interacted with others but were not part of a globally acting team.

This organizational structure seemed to be feasible for the company and allowed to apply a follow-the-sun approach. The management defined clear responsibilities and roles beforehand to reduce conflicts and react fast to customer tickets. In addition, the employees were introduced into follow-the-sun development and received hands-on training.

#### B. Problems

After the initial phase, we interviewed 20 company employees from different teams to receive feedback and to identify problems. 14 of them have degrees in computer science or closely related fields.

The first results of the interviews were promising. We found no language or cultural barriers neither in communication nor interaction of the teams, mainly due to their existing experience in international cooperation. As a result, the participants reported that between 17 and 65% of their tasks were globally distributed. On average, they worked on 3 different projects in parallel, of which 2 were passed on with the follow-the-sun approach.

However, the participants also mentioned that they spend most of their time with previously non-existing and time-consuming activities connected to the follow-the-sun approach. This included searching for information, namely due to the following obstacles:

1) **Lack of information policy and tools.**

While the company aimed to apply a follow-the-sun approach, it promoted neither a common information policy nor appropriate tools. Several of the participants mentioned that they found no guidelines or documentation on how and where to store information. Thus, data were often saved locally on the own computer, printed out, or kept in the e-mail client. The lack of information policy also entailed the other problems.

2) **Constrained knowledge circulation.**

In practice, employees used several ways to share knowledge with co-workers, for instance via mail, dialogue, or intranet. However, the participants reported that they shared information only occasionally and locally. Hence, each team had a different knowledge base, making it difficult to hand off and work on the same project. For this reason, the employees often had to rely on experts or seek advice of their supervisors.

3) **Missing awareness of globally available information.**

The locality of knowledge sharing was also a result of an unawareness of an existing globally available information source. While the company installed the system to manage and store project-related information, only few participants knew this. Thus, the system was rarely updated, wherefore information were sparsely provided, resulting in a vicious cycle. Unfortunately, employees were not able to identify global experts except their usual suspects.

4) **Problematic change management.**

The incident and change management relied on a ticket component of the information system and the responsible process owners. Participants reported two major problems with this approach, resulting in delays in solving customer tickets: Firstly, the German company branch's system provided the basis and, thus, all processes and developers depended on the corresponding knowledge. Due to the lack of globally available information and knowledge, they were not able to answer tickets as fast as possible. Secondly, software changes were often found faulty during reviews, which increased corresponding costs. This was a result of splitting off teams and business units, leading to certain instabilities and reliance.

Despite these problems, the company decided to continue its follow-the-sun approach. In an adoption phase, several systems were added to overcome the challenges reported in these first interviews.

### C. Adoption Phase

During the adoption phase, the company implemented two solutions to overcome the existing problems. Firstly, it established a central knowledge sharing platform in which employees had to store all information. This platform consisted of a wiki, content management, and discussion forum. The whole project management was integrated, wherefore the lack of available information should be overcome. In contrast to the previous situation, all employees were able to:

- access information on running and finished projects,
- find connections between projects, data, and employees,
- identify experts for specific tasks and projects, and
- get information on business politics and guidelines.

With this solution, the company aimed to improve the efficiency and effectiveness of knowledge sharing.

Secondly, the company handled tickets with the follow-the-sun approach and passed them on to globally acting teams to find solutions faster. Also, the company wanted to improve the quality of implementations by including more experts in each participating region than before. To guarantee fast delivery and provide knowledge globally, solutions and the corresponding information were propagated backwards to the ticket's service centre or process owner. Hence, instead of only handing off tasks in one direction (follow-the-sun), solutions were also handed back to ensure that all employees gain knowledge.

The globally common and available information helped to solve the problems identified after the initial phase. However, during further interviews and observations, we found that the follow-the-sun approach was still rarely applied. Mostly, it was used for simple tickets in the incident management and maintenance. Finally, the company decided to put its follow-the-sun approach on hold but integrate the gathered knowledge into its processes. Summarized, the management did not consider the project to be successful but some changes improved its software development and the gained experiences were used in later projects.

## III. DISCUSSION

During our final investigations we received feedback from several developers from all over the world, including the previous 20. We asked them to assess four types of features to identify problems of the company's follow-the-sun approach. In this section, we exemplify the feedback and discuss lessons learned, focusing on the root causes, during this case study.

### A. Community

Within the first category, we wanted to know how well developers felt involved into the follow-the-sun approach and their teams. We received some positive answers, but most of the statements were controversial:

- “A global team does not really exist.”



- *“I don’t really have the possibility to develop personal relationships with other community members.”*

Most of the received answers were similar to those and emphasized missing face-to-face interactions between teams and their members.

In further interviews, we found the main reason for this to be trust issues. Experts, especially external ones and those in more expensive regions, were not motivated to share their knowledge. They feared to lose their work, or at least their standing, if they communicated everything they know to the community. Hence, they avoided the platforms and connected only with each other.

Referring to findings of knowledge management [17] and based on our results, we strongly agree with previous studies on the importance of trust in global software development. Al-Ani and Redmiles [2] as well as Marczak et al. [16] emphasize the importance of trust for diverse and distributed teams. These points were essential in our case study: The teams were located all over the world with diverse roles and knowledge, wherefore they rarely shared knowledge or got familiar with.

In addition, we emphasize that this is a challenge for both sides: Employees in the originating region (i.e., Germany) feared to lose competences to other teams. Contrary, participants in other regions were afraid of processes being centralized and migrated towards Germany.

## B. Tools

Our second category considered the tools that were implemented in the adoption phase to enable discussions and knowledge management. Several employees reported that the situation clearly improved compared to the initial phase but was still problematic:

- *“There are too many different tools. I totally lose track. The functionality of the tools is unclear.”*
- *“It takes too long to get relevant information from colleagues of other regions, a lot of times there is a delay because people are not available.”*

The newly introduced tools added further complexity and effort to the employees’ work, which is a common barrier in knowledge sharing [17]. Most of them needed too much time to familiarize, especially when they were external consultants, and had huge problems to use the tools properly. Additionally, most software developers relied on existing and open-source platforms, for instance `stackoverflow.com`<sup>1</sup>, which they knew well. Hence, the tools were used less frequently than desired and rarely provided solutions or complementary data.

## C. Participation

In a third category, we aimed to identify who participated in the community and why. Several employees criticized that the few experts who contributed were not recognized enough for project decisions, especially during the follow-the-sun adoption:

- *“The key experts should have more influence. They should make the decisions instead of the project lead.”*

<sup>1</sup><http://stackoverflow.com/>, 12.12.2016

- *“I have the feeling that the feedback of the key experts is not valued enough and that they do not really have an influence on further decisions within the project. This fact has to change.”*

We received several responses similar to these two and identified missing integration hindering employees to get more involved.

While experts were rarely willing to share their knowledge, they often proposed and discussed ideas for projects. However, due to the feeling of not being noticed, most of them quickly changed back to their previous work-flows. As a result, and because they feared tracing of their knowledge and performance, most other members of globally acting teams also refused to participate. At the end, the company was not able to gather the required critical mass of developers to provide a solid knowledge base.

## D. Information

The last category focused on availability of information and its distribution between employees. It partly summarized the previous three categories and we aimed to confirm our findings:

- *“Is there a process for modifications? I don’t know where to look for it and on top I don’t know who to ask for problems like this. I have no direct contact person. As an external consultant I feel completely separated.”*
- *“The knowledge transfer from local level to global level is very substandard: If there are changes we do not get any information.”*

Overall, the responses were similar to those for the previous categories.

In particular, we found that external consultants are barely interested into knowledge sharing. They are even less motivated to share their valuable knowledge as it is the reason they were hired. Thus, they rarely participated or provided information. Due to the fact that a huge number of consultants had been involved in the company’s projects, their missing participation represented a huge deficit for knowledge management.

## E. Lessons Learned

As we discussed, we found several aspects that have to be considered for follow-the-sun approaches. Based on our experiences in the case study, we summarize the following five practices as key factors for globally distributed development:

### 1) **Build trust between global teams and their members.**

Many employees feared to lose standing or even their jobs if they share their knowledge with others. Hence, they have to be convinced that such collaborations will be to their benefits and the information are not used to assess their performance. Furthermore, it is important to connect not only team members but also the globally distributed teams themselves.

### 2) **Communication is essential.**

While the company saw communication mainly as the way to pass on information and hand off projects, it was far more important. Most employees missed specific kinds of communication, most prominently face-to-face

meetings, and stated that this hampered team building and relationships. Hence, follow-the-sun approaches require to include different channels for employees to communicate and get to know each other. This way, trust can be build and knowledge sharing improved.

### 3) **Integrate experts into decisions.**

When experts and other employees got the feeling that their opinions were ignored, they quickly rejected the idea of global software engineering. If ideas are not discussed or decisions are made without considering them, employees get the impression that their opinions do not matter. To build trust in the approach and also benefit from the global knowledge, a company should include experts into decisions and communicate results.

### 4) **Motivate knowledge sharing.**

It is often assumed that rewards motivate people to participate in knowledge sharing [1, 17], wherefore a rewarding system was part of the used wiki. However, our case study contradicts this assessment in the context of global software engineering and reinforces corresponding findings, for instance by Bock and Kim [3]. Hence, employees must have intrinsic motivation to cooperate globally and provide their knowledge. This can only be achieved by building trust, showing benefits, and introducing an open community.

### 5) **Use and promote existing tools.**

Tools are essential in follow-the-sun approaches to hand off and manage projects as well as knowledge. Still, companies should not just introduce new tools or adopt those that work for others. Instead, it should focus on providing few, simple, and well-known solutions. Processes could be integrated into existing tools or those already used by the developers, for instance to gain knowledge. Thus, a company can facilitate familiarization and prevent rejections by its employees.

The above described points cannot be considered independently but are closely related. Only based on effective communication and motivation, employees will build trust or can be integrated into decisions and the other way around.

## IV. CONCLUSIONS

Utilizing global software development offers companies several benefits. However, corresponding approaches, such as follow-the-sun, also introduce new challenges for software engineering.

In this paper, we report a case study on follow-the-sun in a German company. We interviewed several employees and identified five key factors for such projects. A company has to build *trust* between its employees for which the most important factor is *communication*. Furthermore, the management should *integrate* experts, *motivate* knowledge sharing, and promote suitable *tooling*.

Our findings support practitioners in assessing and implementing follow-the-sun approaches. For future work, we see the necessity to conduct further studies, focusing on

methods and utilizing reports on best practices and problems. In addition, we found knowledge sharing and trust management to be challenging in global teams, which also requires further research.

## ACKNOWLEDGMENT

This research is supported by DFG grant LE 3382/2-1. We thank Heike Fischbach for commenting drafts of this paper.

## REFERENCES

- [1] A. I. Al-Alawi, N. Y. Al-Marzooqi, and Y. F. Mohammed. Organizational Culture and Knowledge Sharing: Critical Success Factors. *J. Knowl. Management*, 11(2):22–42, 2007.
- [2] B. Al-Ani and D. Redmiles. In Strangers we Trust? Findings of an Empirical Study of Distributed Teams. In *ICGSE*, pages 121–130. IEEE, 2009.
- [3] G.-W. Bock and Y.-G. Kim. Breaking the Myths of Rewards: An Exploratory Study of Attitudes About Knowledge Sharing. *Info. Res. Management J.*, 15(2):14–21, 2002.
- [4] E. Carmel, Y. Dubinsky, and A. J. Espinosa. Follow The Sun Software Development: New Perspectives, Conceptual Foundation, and Exploratory Field Study. In *HICSS*, pages 1–9. IEEE, 2009.
- [5] E. Carmel, A. J. Espinosa, and Y. Dubinsky. "Follow the Sun" Workflow in Global Software Development. *J. Manage. Inform. Syst.*, 27(1):17–38, 2010.
- [6] A. Cockburn and J. Highsmith. Agile Software Development, the People Factor. *Computer*, 34(11):131–133, 2001.
- [7] E. O. Conchúir, P. J. Ágerfalk, H. H. Olsson, and B. Fitzgerald. Global Software Development: Where Are the Benefits? *Commun. ACM*, 52(8):127–131, 2009.
- [8] L. Dubé and G. Paré. Global Virtual Teams. *Commun. ACM*, 44(12):71–73, 2001.
- [9] G. Garrison, R. L. Wakefield, X. Xu, and S. H. Kim. Globally Distributed Teams: The Effect of Diversity on Trust, Cohesion and Individual Performance. *ACM SIGMIS Database*, 41(3): 27–48, 2010.
- [10] E. R. Hess and J. L. N. Audy. FTSproc: A Process to Alleviate the Challenges of Projects that use the Follow-The-Sun Strategy. In *ICGSE*, pages 56–64. IEEE, 2012.
- [11] J. Highsmith and A. Cockburn. Agile Software Development: The Business of Innovation. *Computer*, 34(9):120–127, 2001.
- [12] H. Holmstrom, E. Ó. Conchúir, P. J. Ágerfalk, and B. Fitzgerald. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. In *ICGSE*, pages 3–11. IEEE, 2006.
- [13] J. Kotlarsky and I. Oshri. Social Ties, Knowledge Sharing and Successful Collaboration in Globally Distributed System Development Projects. *Eur. J. Inform. Syst.*, 14(1):37–48, 2005.
- [14] J. Kroll, E. R. Hess, J. L. N. Audy, and R. Prikladnicki. Researching into Follow-The-Sun Software Development: Challenges and Opportunities. In *ICGSE*, pages 60–65. IEEE, 2011.
- [15] J. Kroll, S. I. Hashmi, I. Richardson, and J. L. N. Audy. A Systematic Literature Review of Best Practices and Challenges in Follow-The-Sun Software Development. In *ICGSEW*, pages 18–23. IEEE, 2013.
- [16] S. Marczak, B. Al-Ani, D. Redmiles, and R. Prikladnicki. The Interplay among Trust, Risk, and Reliance in Global Systems Engineering Teams. In *ICGSE*, pages 46–55. IEEE, 2014.
- [17] A. Riege. Three-Dozen Knowledge-Sharing Barriers Managers must Consider. *J. Knowl. Management*, 9(3):18–35, 2005.
- [18] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi. Systematic Literature Reviews in Global Software Development: A Tertiary Study. In *EASE*, pages 2–11. IET, 2012.