2013 IEEE 8th International Conference on Global Software Engineering Workshops

A Systematic Literature Review of Best Practices and Challenges in Follow-the-Sun Software Development

Josiane Kroll[1], Sajid Ibrahim Hashmi[2], Ita Richardson[2], Jorge L. N. Audy[1]
Computer Science School, Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil[1]
Lero - the Irish Software Engineering Research Centre, University of Limerick, Ireland[2]
{josiane.kroll@acad.pucrs.br, audy@pucrs.br} [1]
{Sajid. Hashmi, Ita. Richardson}@lero. ie[2]

Abstract— **Follow-the-sun (FTS) software development is a strategy used to reduce the length of software projects that are developed across globally distributed locations. However, due to communication and collaboration challenges, software companies find it difficult to adopt this development strategy during task allocation and daily project handovers. In this study, we present results from a Systematic Literature Review (SLR) performed on papers published between 1990 and 2012. Our goal was to identify best practices and challenges for FTS implementation. We found 36 best practices and 17 challenges for FTS. These results are discussed in this paper in order to indicate opportunities for future research and make our results useful for the project managers.**

Keywords—Global software development; Follow-the-sun (FTS); best practice; challenges.

I. Introduction

FTS is a subset of GSD (Global Software Development) where software development is distributed over 24 work hours per day in order to reduce the overall development time [1]. In FTS, team members are spread across different time zones to achieve a single project outcome [2]. Many companies have tried to implement FTS strategy, but have abandoned it after some point because of the difficulty of putting it into practice [1].

For this reason, our study aims to investigate best practices and challenges for putting FTS into practice. It extends the study published by [3], and provides new information about FTS best practices and challenges. We substantially extend the empirical evaluation of FTS which was conducted in that previous study.

In this paper, we present 36 best practices and 17 challenges in FTS implementation. We then discuss these results indicating opportunities for future research. We also discuss our results in order to make it useful for the project managers concerned.

The paper is organized as follows. In the next section, we describe the follow-the-sun concept. In section 3, we present the research method. In the section 4, we present the results obtained. In the section 6, we discuss the results. Finally, in the section 7, we draw our conclusions.

II. Follow-the-Sun Software Development

Follow-the-sun (FTS) is a software development strategy used in the GSD context in order to take advantage of the temporal distance between several production sites located in different time zones [4] [1]. Its main purpose is the reduction of the software development life cycle duration or time-to-market in order to obtain a competitive business advantage [1].

When a working team finishes its regular working hours, another team located in another location and time zone starts its workday. Unfinished tasks are handed from one team to another by the end of each working day [5].

The transition of tasks between the teams is called handoff [6]. At each location, handoffs are conducted on a daily basis, at the end of each site shift [7]. The concept of handoffs, with the segregation of tasks, enables software development teams to work on a continuous basis on the project [8]. According to [10], FTS is a special case of GSD where there is a handoff of unfinished work every day. FTS efficiency is determined by the quality of knowledge transfer and the duration it involves [7].

III. Research Method

We conducted a Systematic Literature Review (SLR), following the guidelines defined by Kitchenham and Charters [9]. The first step to perform an SLR is to define a research protocol, which is described in the next.

A. Research Questions

We defined two research questions (RQ) for this study:

RQ1: *What FTS challenges are reported in the literature?*

RQ2: *What are the best practices recommended for FTS?*

B. Data Sources

We searched published studies in seven digital libraries as shown in Table 1. For each digital library, query strings were created according to the search tool. We targeted literature published between 1990 to 2012 because studies on GSD began in the early 1990's [10].

C. Search String

In the literature, sometimes FTS is also referenced as 24-hour development model, 24-Hour Knowledge Factory Paradigm (24HrKF), round-the-clock and shift work. We included these terms as part of our search string in order to identify as many relevant papers as possible. The search was conducted using the boolean search expression as follows:

978-0-7695-5055-8/13 $26.00 © 2013 IEEE
DOI 10.1109/ICGSEW.2013.10

IEEE computer society

(("Follow-the-sun" <OR> "round-the-clock"<OR> "24-hour development" <OR> "24-Hour Knowledge Factory Paradigm" <OR) "shift work") <AND> "software")

D. Selection Process

After an extensive data search, we came up with 773 studies. To select papers, one of the authors read the title followed by the abstract. We excluded posters, panels, abstracts, presentation and summaries studies. At this point, one author read the full paper. Repeated studies, those that did not specifically focus on FTS and the ones that did not belong to software engineering were excluded. The number of studies was reduced to 27. The numbers of studies found for each resource are listed in Table 1.

TABLE I. STUDIES SELECTION

Digital library	**Total**	**Excluded**	**Included**
IEEE Xplore	106	91	15
ACM Digital Library	251	244	7
Wiley Inter Science Journal Finder	81	79	2
Elsevier Science Direct	33	32	1
Spring Link	155	154	1
ISI Web of Knowledge	54	53	1
Engineering Village	93	93	0
Total	**773**	**746**	**27**

E. Data Extraction Process

We created a data extraction form using MS Excel. Meta-data such as author, title, year and publication source were collected with descriptive data fields such as topic, challenges and proposed best practices. To identify best practices, we followed the definition given by Williams [11]: *"A best practice is a software development practice that, through experience and research, has proven to reliably lead to a desired result and is considered to be prudent and advisable to do in a variety of contexts."*

F. Validity of the Process

The main threats to the validity of the process are the study selection, inaccuracy in data extraction, incorrect classification of studies, research methods and types, and potential author bias. In order to ensure that process of selection and inaccuracy in data extraction was unbiased, we followed Kitchenham and Charters [9] recommendations. In relation to concepts used in the search, we assume there is no incorrect definition for FTS, because the research area is not consolidated as of yet.

Regarding the study's classification and findings, at least two researchers discussed each paper. In case of disagreement, the issue was discussed until a consensus. Therefore, there is a possibility that the extraction process may have resulted in removing some papers which should be included.

IV. RESULTS

This section presents the results from the research questions we defined for the SLR.

A. Challenges Reported in the Literature for FTS (RQ1)

To answer RQ1, we mapped the challenges in these three categories. We also calculated the frequencies of challenges in different studies (Column 3). These findings are listed in Table 2.

TABLE II. FTS CHALLENGES

No	**Challenge (CH)**	**Freq.**	**Reference**
COORDINATION			
01	Time zone differences	9	[9][12] [13] [14] [15] [16] [17] [18] [19]
02	Daily handoff cycles or handing off work-in progress (unfinished objects)	9	[1] [6] [7] [8] [13] [15] [18] [20] [21]
03	Geographic dispersion	3	[6] [15] [22]
04	Cost estimation	3	[17] [21] [23]
05	Loss of teamness	2	[6] [24]
06	Number of sites	1	[13]
07	Coordination breakdown	1	[6]
08	Managerial difficulties	1	[25]
09	Technical platforms	1	[16]
COMMUNICATION			
10	Communication difficulties (socio cultural diversity)	8	[1] [6] [7] [13] [15] [16] [22] [25]
11	Synchronous communication	5	[4] [7] [8] [17] [24]
12	Language differences	3	[15] [18] [22]
13	Loss of communication richness	2	[6] [23]
14	Technical difficulties	1	[25]
15	Manage religious or national holidays	1	[26]
CULTURE			
16	Cultural differences (increase in number of development sites, lack of synchronous communication)	7	[1] [6] [13] [15] [18] [16] [22]
17	Different technical backgrounds	3	[16] [17] [19]

B. Best Practices Recommended for FTS (RQ2)

Table 3 shows the outcomes from RQ2. Each best practice identified is described in the next.

TABLE III. BEST PRACTICES FOR FTS

No.	**Best Practices (BP)**	**Freq.**	**Reference**
01	Agile methods	6	[1] [6] [16] [17] [23] [26]
02	Use of technology for knowledge sharing	6	[9] [20] [26] [27] [28] [29]
03	Process documentation	3	[17] [27] [30]
04	Use of an FTP Server (or data repository) to exchange code and documents	3	[6] [27] [28]
05	Time window	3	[4] [3] [31]
06	TDD (Test-driven development)	2	[7] [32]
07	Application of FTS for testing	2	[1] [6]

08	Overlap of one hour between distributed teams	2	[3] [33]
09	Calendar of handoff sessions should be clearly defined	2	[3] [33]
10	Backup teams	2	[3] [33]
11	CPro concept	2	[23] [32]
12	Implementation of 'tracking system'	2	[3] [33]
13	Use of proper communication technologies or tools	2	[4] [25]
14	Scrum stand-up meetings	2	[23] [34]
15	Face-to-face communication	2	[18] [31]
16	Time zone management	2	[3] [12]
17	Pair programming	1	[27]
18	Application of FTS for testing and development phases	1	[21]
19	Daily exchange of the project status by technologies	1	[28]
20	Daily handoffs of 30 minutes duration with each development site	1	[7]
21	Screen sharing	1	[4]
22	Clean handoff and sticky handoff interactions	1	[35]
23	Wikis and online forums to share knowledge between team members	1	[23]
24	Low task granularity	1	[24]
25	Task distribution by sequencing or dependency	1	[27]
26	Emails out-of-hours	1	[31]
27	Informal, unplanned and ad hoc communication	1	[15]
28	Corporate technologies	1	[4]
29	Models of emails and electronic messages	1	[36]
30	Opt out for development sites where team members could speak the same language	1	[31]
31	At least one hour of overlap between two production sites	1	[33]
32	Teams distribution across two or three sites	1	[37]
33	Meetings between team members for building trust	1	[15]
34	Team members with same culture	1	[18]
35	Cultural awareness training	1	[17]
36	Similar code patterns	1	[27]

- *BP01- Agile methods*: agile methods or adaptive approaches aim to adapt quickly to software development environments. Agile methods also emphasize communication and collaboration in an iterative software development process [26].

- *BP02 - Use of technology for knowledge sharing*: many technologies are available to make knowledge sharing easier between the teams. Tang et al. [4] and Gupta et al. [20] recommend technologies such as, webcams and instant messaging software to improve communication between the team members distributed across multiple sites.

- *BP03- Process documentation*: implementation of this practice ensures availability of technical documentation. It also can be used to maintain a history of FTS implementation, which would subsequently improve the decision making process.

- *BP04 - Use of an FTP Server (or data repository) to exchange code and documents:* this practice consists of the use of a common data repository to exchange code and documents between team members. Project files and code can be stored in this data repository. All team members should have full access to this data repository [6] [27].

- *BP05 - Time window*: this practice is used by the teams to minimize collaboration conflicts between sites. It provides opportunities for synchronous interactions without prior schedule definition [31].

- *BP06 – Test Driven Development (TDD):* this practice provides an approach for incremental software development, in which software units are developed in small pieces. This approach does not require initial design details as software units are incrementally developed following test-before-code stile [38].

- *BP07 - Application of FTS for testing*: testing is the best software development phase to implement FTS [21]. In this phase, small and low complexity tasks can be handled regularly between production sites separated by different time zones.

- *BP08 - Overlap of one hour between distributed teams:* to perform handoffs at the beginning and at the end of each working day are necessary to ensure an overlap of one hour between the distributed teams, in order to provide opportunities for synchronous communication [3] [33].

- *BP09 - Calendar of handoff sessions should be clearly defined*: this practice is used to provide better communication between teams. It allows the teams to interact daily according to the same timetable [33].

- *BP10 - Backup teams*: this practice is used to give 24/7 support during holidays and weekends. Implementation of *Backup teams* ensures that information is not lost due to a probable communication channel breakdown during the national holidays and weekends [33] recommends that at least 10% of the teams must be available to implement this practice.

- *BP11 - CPro concept*: CPro is an agile software process that improves the CP (Composite Persona) performance. It also assigns workloads to the different members of a CP, in a way that maximizes productivity [32].

- *BP12 - Implementation of 'tracking system'*: 'tracking system' is implemented to check teams' performance in GSD environments. This practice aims to plan and control events that can result in delays for projects [33].

- *BP13 - Use of proper communication technologies or tools*: communication between FTS teams can be carried out using proper communication technologies or tools [28] such as, telephone calls, emails and IM. Furthermore, many communication technologies and tools are available to support communication between distributed teams.

- *BP15 - Face-to-face communication*: in FTS context, the end-product quality may suffer due to lacking of options available for synchronous communication [18]. Rich communications media like face-to-face tend to be more efficient than media such as telephone or email [31].

- *BP16- Time zone management:* time management is necessary to fit the teams' working hours for a good overlap [12]. However, choosing sites for a good overlap is not always possible. Time zone differences became manageable when is possible to negotiate teams working hours.

- *BP18 - Application of FTS for testing and development phases*: evidence from studies conducted on software industry shows that FTS is effective for testing as well as development phases. These phases can work well in FTS because handoffs are structured and granulate [1].

- *BP19- Daily exchange of the project status by technologies:* this practice recommends the use of technologies such as, telephone calls, video conferences or emails for the daily exchange of the project status. Telephone calls and video conferences provide synchronous communication for real time interactions [28]. These technologies may be used in conjunction with others.

- *BP20 - Daily handoff of 30 minutes duration with each development site*: Hess and Audy [7] recommend that handoff sessions should be of 30 minutes duration between the two sites. According to these authors, 30 minutes are sufficient to transfer tasks and discuss task details.

- *BP21 - Screen sharing:* screen sharing contributes to transfer knowledge between team members [4]. Its use makes easy to understanding the information that is been discussed.

- *BP22 - Clean handoff and sticky hands-off interactions:* this practice discusses punctual questions related to the project. On the other hand, sticky hands-off interactions are more intense, but can be used effectively [35].

- *BP23 - Wikis and online forums to share knowledge between team members*: this practice consists on creating an internal wiki and online forums as a knowledge base in order to share problems and solutions. Both of these provide informal knowledge in a structured format.

- *BP24 - Low task granularity*: FTS can be effective for software development in context to low task granularity, such as, bug correction or call center activities; i.e. technical support [24].

- *BP25 - Task distribution by sequencing or dependency:* in the sequencing or dependency distribution, one task is divided between two or more members who are distributed across different time zones. One member would transfer the task to another member localized in a different site. This member would take up the task and would continue from the point since the preceding team's member made the last change. This practice allows for 24 hours working development [27].

- *BP26 - Out-of-hours emails*: time zone difference between the development sites may invariably make team members to perform part of their work at home. Out-of-hour emails help to reduce potential delays between sites. This practice can be implemented by providing free internet access and laptops for all teams involved on the project [31].

- *BP27 - Informal, unplanned and ad hoc communication:* BP27 is important to support collaboration between the teams. It can be implemented through discussion pairs [15].

- *BP28 - Corporate technologies*: BP28 recommends technologies such as, video conferencing, screen sharing and other corporate resources for the teams attending meetings from their homes. This practice provides more flexible interaction windows to increase connectivity between the teams [4].

- *BP29 - Models of email and electronic messages:* a unique message template could be used to assign specific meaning to a message, for example, technical and non-technical requests could be distinguished by using different message templates. These templates should describe the essential information with fields that could facilitate in recalling information typically included in the actual message.

- *BP30 - Opt out for development sites where team members could speak the same language*: many problems occur due to language issues. Choosing offshore teams with the same language is advantageous for FTS [31].

- *BP31 - At least one hour of overlap between two production sites:* Management of time overlaps between sites reduces communication and coordination problems during handoff sessions [26]. Moreover, effective management of overlaps helps to promote 27/4 support.

- *BP32 - Teams distribution across two or three sites:* this BP defines the number of sites for FTS, which must be at least two sites [37]. More than three sites may result in coordination problems.

- *BP33 - Meetings between team members for building trust:* meetings are used to establish or reestablish trust, increase in the number of project meetings would definitely help to increase the level of trust among the team members; whereas, reduction in it would definitely hamper the cause [15].

- *BP34 - Team members with the same culture*: team members who share the same culture develop trust more quickly than those who come from different culture [18]. Furthermore, team members from the same culture are more inclined to establish trust than the team members from different culture.

- *BP35 - Cultural awareness training:* BP35 aims to develop cultural awareness among team members. This practice should be implemented at the beginning to educate team members on each others culture.

- *BP36- Similar code patterns:* similar code patterns allow team members to understand and identify changes made in the code since the last handoff session. Furthermore, similar code patterns can avoid reworking [27].

V. Discussion

We investigated the findings from 27 relevant studies that were published since 1990. As a result, we obtained as results 17 challenges and 36 best practices for FTS implementation.

Related to the challenges identified, our analysis focuses on the frequencies of them. This makes possible to see which categories have been emphasized in past research and thus to identify gaps and possibilities for future research.

In the Coordination category, a great number of studies report time zone differences and daily handoff cycles as challenges for FTS implementation. This result makes sense, considering that time zone differences are the main characteristic of FTS projects and daily handoff cycles are used two or three times a day to transfer tasks between sites [1]. In addition, within Coordination, seven more challenges were found in three studies. Although these findings point to lower frequencies, not identifying the challenges can lead to negative consequences for FTS projects. Unfortunately, we found very few successful cases of FTS. One of the reasons for this may be that companies do not deal effectively with coordination challenges.

Eight studies reported Communication difficulties, often related to the socio-cultural diversity of teams [15]. We also found five studies reporting synchronous communication as a challenge. According to [18] [31] the lacking of face-to-face communication in GSD projects is a main obstacle to communication. In FTS, making opportunities for spontaneous interaction can result in a large amount of communication overhead introduced during task handoffs [8].

Language differences and loss of communication richness is mentioned as a challenge caused by socio-culture distance [15]. Technical difficulties are related to the disparity in infrastructure whereas the management of religious or national holidays poses yet another challenge, as they do not coincide with those holidays in western locations.

In the Culture category, we found two challenges cited by ten studies. Cultural differences arise due to circumstances such as increased numbers of development sites, lack of synchronous communication and differing languages. Different technical backgrounds can be caused by different skills and competencies. Both are determined mainly by social, ethnic and religious aspects [15].

Related to best practices identified, we have observed that *BP01 - Agile methods* and *BP02 - Use of technology for knowledge sharing* were the most cited in studies (six studies each). BP01 recommends agile methods for FTS. Agile methods have high acceptance in the software industry. XP and Scrum are the most indicated to implement FTS [7] [23].

BP02 recommends using technologies to develop FTS activities. Technologies like conference video, telephone calls and email are low cost strategies and may be utilized by companies to perform synchronous and asynchronous communication between teams.

BP03 – Process documentation, BP04 - Use of a FTS server (or data repository) to exchange code and documents and *BP05 - Time window* has three studies each. BP04 and BP05 are used to perform handoffs. This result makes sense, considering daily handoff cycles as challenges for FTS implementation. On the other hand, BP03 is not a usual practice adopted by agile methods. However, BP03 provide advantages, such as, product and quality service improvement, cost reduction and using of resources in the best way.

Other best practices identified have two studies each. It was observed that 51% of best practices report communication aspects, 40% coordination aspects and only 8% cultural aspects. These findings show a lower percentage of studies discussing cultural aspects. However, cultural aspects are not less relevant for FTS. Cultural diversity is discussed as a barrier for FTS teams and it can negatively affect on understanding level, task development and team effort [33].

VI. Conclusions and Future Work

GSD organizations aim to use FTS in order to gain a competitive advantage. However, FTS challenges make this strategy difficult to implement in a Global environment. Moreover, since the existing literature on the area does not fully address any concrete approach to successfully implement FTS, there remains a big research gap. In order to fill this gap, we performed a SLR and came up with certain FTS practices as well as challenges that were reported by different existing studies.

Challenges identified are focused on the main FTS characteristics. It appears an immature research area. There are many opportunities for future studies related to coordination, communication and culture aspects.

The analysis of best practices revels that the mostly best practices identified are generic best-practices for software development. Companies can use its knowledge to adapt its own practices to develop FTS. Future studies will aim to identify GSD practices associate to FTS challenges, in order to minimize potential problems that involve key aspects of FTS.

Acknowledgment

This research has received funding from Science Foundation Ireland grant 10/CE/I1855 to Lero (www.lero.ie), a Higher Education Authority grant PRTLI 4 to the Lero Graduate School in Software Engineering. The first author is supported by CAPES (Brazilian Ministry of Education).

References

[1] E. Carmel, J. A. Espinosa and Y. Dubinsky, "Follow the Sun Workflow in Global Software Development", Journal of Management Information Systems Vol. 27 No. 1, pp. 17 – 38, 2010.

[2] A. Cameron, "A Novel Approach to Distributed Concurrent Software Development using a Follow-the- Sun Technique", Unpublished EDS working paper, 2004.

[3] Kroll, J. and Audy, J. L. N., "Mapping Global Software Development Practices for Follow-the-Sun Process," Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , pp. 164-168, 27-30 Aug. 2012.

[4] J. C. Tang, C. Zhao, X. Cao and K. Inkpen, "Your time zone or mine?: a study of globally time zone-shifted collaboration," In Proceedings of the ACM 2011 conference on Computer supported cooperative work (CSCW '11). ACM, New York, NY, 2011, USA, 235-244.

[5] E. Carmel, Y. Dubinsky and J. Johnston, "Follow the sun workflow in global software development: Theory, modeling and quasi-experiment to

explore its feasibility", In Proceedings of the Third Global Sourcing Workshop: The Impacts of Global IS Sourcing on Engi- neering, Technology and Innovation Management, Key- stone, CO, USA, March 2009.

[6] E. Carmel, A. Espinosa and Y. Dubinsky, "Follow The Sun Software Development: New Perspectives, Conceptual Foundation, and Exploratory Field Study". 42nd Hawaii International Conference on System Sciences, Proceedings, 2009.

[7] E. Hess and J. L. N. Audy, "FTSProc: A Process to Alleviate the Challenges of Projects that Use the Follow-the-Sun Strategy," Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , vol., no., pp. 56-64, 27-30 Aug. 2012.

[8] A. Gupta, I. Crk, and R. Bondade, "Leveraging temporal and spatial separations with the 24-hour knowledge factory paradigm," Information Systems Frontiers 13, 3 (July 2011), 397-405.

[9] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering (version 2.3). Technical report, Keele University and University of Durham, 2007.

[10] R. Prikladnicki, J. L. N. Audy and F. Shull, "Patterns in Effective Distributed Software Development." IEEE Software, v. 27, p. 12-15, 2010.

[11] L. Williams, A (partial) Introduction to Software Engineering Practices and Methods, 2008-2009 (Fifth) Edition.

[12] H. Holmstrom, E. O. Conchuir, P. J., Agerfalk and B. Fitzgerald, "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance". Proceedings of the IEEE international conference on Global Software Engineering (ICGSE '06). IEEE Computer Society, Washington, DC, USA, pp. 3-11, 2006.

[13] J. Kroll, E. Hess, J. L. N. Audy, and R. Prikladnicki, "Researching into Follow-the-Sun Software Development: Challenges and Opportunities," In: 6th International conference on Global Software Engineering (ICGSE), 2011, Helsinki, Finland.

[14] E. Carmel and J. A. Espinosa, I'm Working While They're Sleeping: Time Zone Separation Challenges and Solutions, Kindle Edition, 2011, 188 p.

[15] S. Setamanit, W. Wakeland and D. Raffo, "Improving Global Software Development Project Performance Using Simulation". Management of Engineering and Technology, Portland International Center, 2007, 2458-2466.

[16] M. Yap, "Follow the sun: distributed extreme programming development," Agile Conference Proceedings, 218- 224, 2005.

[17] J. J. Treinen and S. L. Miller-Frost, "Following the Sun: Case Studies in Global Software Development," IBM Systems Journal, 45 (4), October 2006.

[18] S. Setamanit; W. Wakeland; D. Raffo, "Planning and improving global software development process using simulation", Proceedings of the international workshop on Global software development, Shanghai (China), 2006, pp. 8–14.

[19] I. Gorton, I. Hawryszkiewycz and K. Ragoonaden, "Colaborative tools and processes to support software engineering shift work". BT Technology Journal, (1997), 189–198.

[20] A. Gupta, E. Mattarelli, Seshasai S. and J. Broschak, "Use of collaborative technologies and knowledge sharing in co-located and distributed teams: Towards the 24-h knowledge factory," The Journal of Strategic Information Systems, Volume 18, 147-161, 2009.

[21] E. Carmel, Building your Information Systems From the Other Side of the World: How Infosys manages time differences, MIS Quarterly Executive, 5 (1), 2006.

[22] A. R. Santos, A. Sales and P. Fernandes, "Setting Up a Stochastic Model for Teams Working in a Follow-the-Sun Environment," Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , vol., no., pp. 179, 27-30 Aug. 2012.

[23] A. Gupta, L. Hu, T. Hedberg, C. Prendergast and I. Crk, "Creating the 24-Hour Knowledge Factory" (February 13, 2012). Disponível em: http://ssrn.com/abstract=2004791.

[24] J. A. Espinosa and E. Carmel, "Modelingcoordinationcosts due to time separation in global software teams", Global Software Development Workshop, International Conference on Software Engineering (ICSE), 2003. IEEE: Portland, OR, 64–68.

[25] P. Jalote and G. Jain, "Assigning tasks in a 24-h software development model," Journal of Systems and Software. 79, 7 (2004), 904-911.

[26] J. Kroll, A. R. Santos, R. Prikladnicki, E. R. Hess, R. Glanzner, A. Sales, J. L. N. Audy and P. Fernandes, "Follow-the-Sun Software Development: A Controlled Experiment to Evaluate the Benefits of Adaptive and Prescriptive Approaches," Proceedings of the 24th International Conference on Software Engineering & Knowledge (SEKE 2012), 551-556.

[27] A. Taweel and P. Brereton, "Modeling Software Development across Time Zones," Information and Software Technology, 48, 1 (January, 2006), 1-11.

[28] V. Ramesh and A. Dennis, "The object oriented team: Lessons for virtual teams from global software development." In HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences, volume 1. IEEE Computer Society, 2002.

[29] I. Gorton and S. Motwani, "Issues in co-operative software engineering using globally distributed teams". Information and Software Technology, Volume 38, Issue 10, p. 647-655, 1996.

[30] G. Avram, "Knowledge Work Practices in Global Software Development," Electronic Journal of Knowledge Management Volume 5 Issue 4, p. 347 – 356, 2007.

[31] B. Lings, B. Lundell, P. J. Ågerfalk and B. Fitzgerald, "A reference model for successful Distributed Development of Software Systems". ICGSE 2007. IEEE International Conference on, pp. 130-139, 2007.

[32] N. Denny, I. Crk, R. S. Nadella, and A. Gupta, "Agile Software Processes for the 24-Hour Knowledge Factory Environment", February 27, 2009.

[33] S. Deshpande and I. Richardson, "Management at the Outsourcing Destination - Global Software Development in India," In Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09). IEEE Computer Society, Washington, DC, USA, 217-225, 2009.

[34] A. Gupta, R. Bondade and N. Denny, "Software Development Using the 24-Hour Knowledge Factory Paradigm," (April 29, 2008). Disponível em: http://ssrn.com/abstract=1130062.

[35] C. Visser and R. V. Solingen, "Selecting Locations for Follow-the-Sun Software Development: Towards A Routing Model", Fourth IEEE International Conference on Global Software Engineering, 2009.

[36] I. Gorton, I. Hawryszkiewycz and L. Fung, "Enabling software shift work with groupware: a case study". System Sciences, Proceedings of the Twenty-Ninth Hawaii International Conference on, Vol. 3, 72-81, 1996.

[37] V. R. Solingen and M. Valkema, "The Impact of Number of Sites in a Follow the Sun Setting on the Actual and Perceived Working Speed and Accuracy: A Controlled Experiment", Global Software Engineering (ICGSE), 5th IEEE International Conference, 165- 174, 2010.

[38] S. Deshpande, I. Richardson, V. Casey and S. Beecham, "Culture in Global Software Development - A Weakness or Strength?," Global Software Engineering (ICGSE), 2010 5th IEEE International Conference pp. 67-76, 23-26 Aug. 2010.