FTS-SPM: A Software Process Model for Follow the Sun Development

Preliminary Results

Josiane Kroll[1], Ita Richardson[2], and Jorge L. N. Audy[1]
[1]Pontifical University Catholic of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
[2]Lero - The Irish Software Engineering Research Centre, University of Limerick (UL), Limerick, Ireland
josi.unc@gmail.com, ita.richardson@ul.ie, audy@pucrs.br

Abstract— Previous work reports that some software companies have tried to implement FTS, but have failed to realize the anticipated outcomes. In our research, we propose a software process model to support FTS implementation, named FTS-SPM (Follow the Sun Software Process Model). The proposed FTS-SPM comprises six sub-processes and twenty-five best practices. In this paper, we present the preliminary results from an expert panel conducted with 20 participants to validate the FTS-SPM software process model. Specific questions were asked to a group of GSD (Global Software Development) experts to uncover the usefulness and relevance of each best practice mentioned in FTS-SPM. The initial findings from interviews show that not all twenty-five best practices are perceived as high value practices for FTS projects.

Keywords—follow the sun; global software engineering; virtual teams; time zone management; software process model.

I. INTRODUCTION

Follow the Sun (FTS) development is a special case of Global Software Development (GSD) [1]. It is applied in the context of GSD projects to take advantage of time zone differences between development sites. As team members are distributed across multiple time zones, organizations can develop software twenty-four hours continuously [2].

FTS is uniquely focused on speed of development. It is applied to software projects when a software product needs to be developed quickly and the cost is irrelevant to the client [3].

FTS is seen as a potential software development strategy for organizations. However, while the FTS concept looks promising in theory, it appears to be difficult in practice. Many software organizations have tried to implement FTS, but have abandoned it after a while because of the difficulty to put it into practice. Consequently, there are only a few documented success cases in the software industry. The lack of software practices and processes to close the gap between theory and practice is observed as the main barrier to the FTS evolution in Software Engineering and in the software industry.

In our research, we propose a software process model for FTS implementation in GSD environments. We named it as FTS-SPM (Follow the Sun Software Process Model). We combined best practices from the literature [4] and lessons learned from a case study [5] to build a software process model for FTS. In order to validate the FTS-SPM, we conducted an expert panel with 20 experts. In this paper, our goal is to present and discuss the initial findings.

The remainder of this paper is organized as follows. In the next section, we give details of the FTS-SPM, best practices and sub-processes. In section 3, we present the research methodology followed in this study. In section 4, we present the initial findings and future work. We draw our conclusions in section 5.

II. THE FTS-SPM: FOLLOW THE SUN SOFTWARE PROCESS MODEL

We proposed the initial software process model for FTS, called FTS-SPM (Follow the Sun Software Process Model), based on the results of the literature review [4] and a case study [5]. Subsequently, we built a second and third version of the FTS-SPM, with a view to improving the initial software process model design [6]. We present the FTS-SPM in Figure 1.

A. Structure of the FTS-SPM

The FTS-SPM comprises six sub-processes: *SP01: Team Setup, SP02: Project Planning, SP03: Communication Protocol, SP04: Cultural Training, SP05: Task Allocation,* and *SP06: Handoff Sessions.* The sequence flow (arrows) between sub-processes show in which sequence each sub-process is developed (shown in Figure 1).

The FTS-SPM has an initial and final state. The initial state causes the process to start with Team Setup (SP01). The final state is the end of the process when all tasks were finished, at which point there is a software delivery. SP01, Team Setup, starts the process. It aims to identify available sites and allocates human resources for the project.

SP02, where project planning is defined, is started following SP01. SP01 provides information to develop the project plans, and these are developed by the project manager.

SP03, SP04 and SP05 are started in parallel following SP02. SP03 defines communication resources and the schedule for synchronous communication between sites. The project manager can suggest technologies or tools already used in other projects. SP04 develops cultural training sessions

978-1-4799-5206-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICGSEW.2014.14

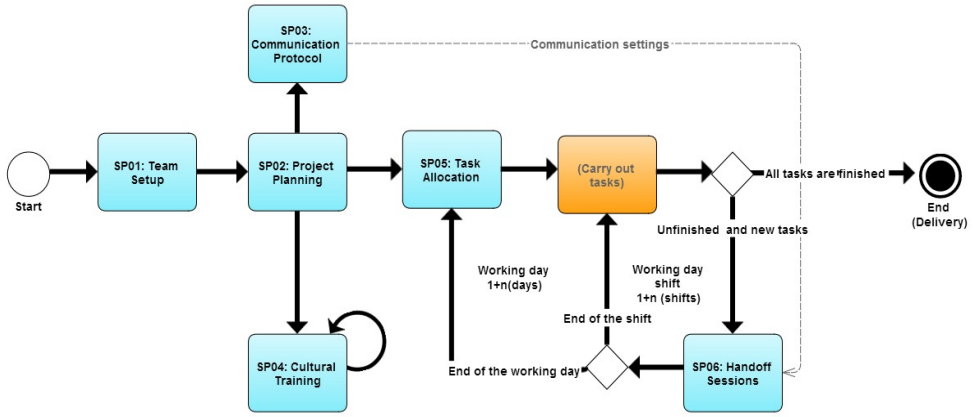

Fig. 1. The FTS-SPM (Follow the Sun Software Process Model).

in order to establish trust between team members. SP04 may be developed many times during the project to re-establish the trust between team members (loop arrow).

At the beginning of each working day, SP05 is undertaken, as it provides tasks for the day. A software project may have many working days. Within SP05, the sequence and dependency relationships between tasks must be identified. All details about tasks sequence and dependency should be described during the project planning.

SP06 is started following SP05. SP06 aims to receive and to transfer tasks in progress, new tasks and project updates. At the beginning and at the end of each working day shift, SP06 is undertaken. One working day may have at least two working day shifts. The process finishes when at the end of a working day shift, there are no more tasks to develop.

'Carry out tasks' is an internal sub-process of the organization. Each organization defines how it should be executed. We show this sub-process in our FTS-SPM to represent how it is related to other sub-processes.

In the first diamond, the process can finish if all tasks are finished or can start SP06, if there are unfinished or new tasks to transfer to another site. In the second diamond, a new working day shift starts if the end of the shift is or else, if it is the end of the working day, SP05 starts.

Arrows in the FTS-SPM show the sequence flows between sub-processes. An additional arrow is included between SP03 to SP06 indicating the relationship between those sub-processes. The communication settings defined in SP03 are used in SP06.

B. *Best Practices*

Sub-processes are developed based on best practices. Best practices were included in the sub-processes based on literature and lessons learned from a case study. Twenty-five best practices comprise the FTS-SPM as shown in Figure 2. Next, we describe each best practice.

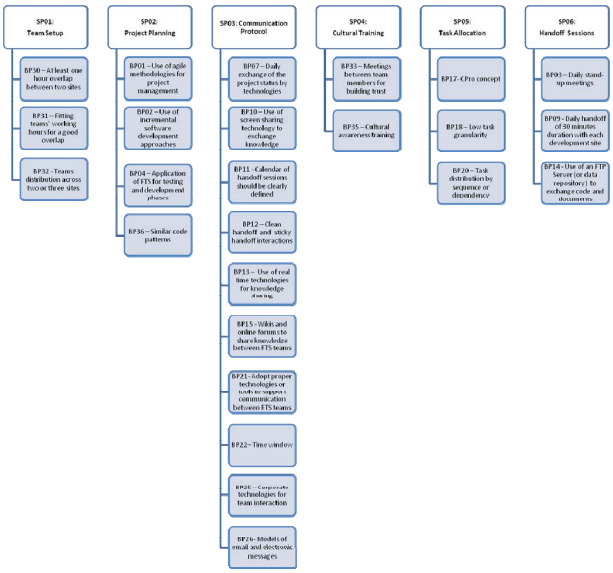

Fig. 2. Best practices included in the sub-processes.

SP01: Team Setup

- *BP30 - At least one hour overlap between two sites:* management of time overlaps between sites reduces communication and coordination problems during handoff sessions [7]. Moreover, effective management of overlaps helps to promote 27/4 support.

- *BP31 - Fitting teams' working hours for a good overlap:* time management is necessary to fit the teams' working hours for a good overlap [8]. However, choosing sites for a good overlap is not always possible. Time zone differences became manageable when is possible to negotiate teams working hours.

- *BP32 - Teams distribution across two or three sites:* this BP defines the number of sites for FTS. This must be at least two sites [2]. More than three sites may result in coordination problems.

SP02: Project Planning

- *BP01 - Use of agile methodologies for project management*: 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 [9].

- *BP02 - Use of incremental software development approaches:* 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 [10].

- *BP04 - 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 [11].

- *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 [12].

SP03: Communication Protocol

- *BP07 - 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 [13]. These technologies may be used in conjunction with others.

- *BP10 - Use of screen sharing technology to exchange knowledge:* screen sharing contributes to transfer knowledge between team members [14]. Its use makes it easier to understand the information that is being discussed.

- *BP11 - 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 [15].

- *BP12 - Clean handoff and stocky handoff 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 [16].

- *BP13 - Use of real time technologies for knowledge sharing:* many technologies are available to make knowledge sharing easier between the teams. Tang et al. [14]) and Gupta et al. [17] recommend technologies such as web cams and instant messaging software to improve communication between the team members distributed across multiple sites.

- *BP15 - Wikis and online forums to share knowledge between FTS teams:* this practice consists of 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.

- *BP21 - Adopt proper technologies or tools to support communication between FTS teams:* communication between FTS teams can be carried out using proper communication technologies or tools [15] such as, telephone calls, emails and IM. Furthermore, many communication technologies and tools are available to support communication between distributed teams.

- *BP22 - 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 [18].

- *BP25 - Corporate technologies for team interaction:* 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 [14].

- *BP26 - Models of emails 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.

SP04: Cultural Training

- *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 [19].

- *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.

SP05: Task Allocation

- *BP17 - 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 Deny et al. [20].

- *BP18 - Low task granularity:* FTS can be effective for software development in context to low task granularity, such as, bug correction or call center activities; e.g. Technical support [21].

- *BP20 - Task distribution by sequence 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 [12].

SP06: Handoff Sessions

- *BP03 - Daily stand-up meetings:* stand-up meetings came up from Scrum methodology. It is a daily team meeting that helps to provide a status update to the team members [22].

- *BP09 - Daily handoff of 30 minutes duration with each development site:* Hess and Audy [10] 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.

- *BP14 - Use of an FTP Sever (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 [11] [12].

III. RESEARCH METHODOLOGY

In this section, we present details of how the expert panel approach was used to validate the FTS-SPM. The value of expert knowledge is also recognized in a recent evaluation of software quality that suggests methods to formally capture expert judgment [23]. The adoption of the expert panel method in this study follows the main goals:

- To gather the view of experts about the applicability of best practices included in the sub-processes for FTS projects;
- To gain an understanding of how best practices included in the sub-processes can support FTS projects.

Experts in the evaluation process will help to refine the FTS-SPM to make it applicable in the software industry. We selected 20 experts to validate the FTS-SPM. In Table 1, we present the information about participants who were interviewed to validate the FTS-SPM.

TABLE I. PARTICIPANTS' INFORMATION

Expert	**Job title**	**Type of experience** (Academic or Industry or Both)	**Location**	**Involved in GSD**
Exp1	Senior Member of Technical Staff	Both	USA	9 years
Exp2	Postdoctoral researcher	Academic	Italy	+10 years
Exp3	Professor/ Researcher	Academic	New Zealand	+4 years
Exp4	Managing Director	Both	Germany	15 years
Exp5	Software Engineer	Both	USA	17 years
Exp6	Project Manager	Industry	Romania	7 years
Exp7	Professor/ Researcher	Both	New Zealand	+20 years
Exp8	Project Manager	Both	Brazil	7 years
Exp9	Project Manager / Researcher	Both	India	10 years
Exp10	Head Marketing	Industry	India	10 years
Exp11	Professor	Academic	USA	+20 years
Exp12	Researcher	Both	Ireland	4 years
Exp13	CTO, Professor	Both	Netherlands	20 years
Exp14	Project Manager	Both	Brazil	7 years
Exp15	Researcher	Both	Ireland	5 years
Exp16	Professor	Academic	Italy	13 years
Exp17	Senior Researcher	Academic	Finland	14 years
Exp18	Professor/ Researcher	Academic	Spain	10 years
Exp19	IT Senior Manager	Both	Brazil	13 years
Exp20	Project Manager	Both	Poland	3 years

A. The Evaluation Process

The participants were asked to provide evidence of experience in practice to support why these best practices or sub-processes should be part of the FTS-SPM. The responses from the interviews are mapped with some recommendations for FTS development. The validation will prove helpful in highlighting the strengths and weaknesses in the FTS-SPM and providing further directions for improving the model.

Two questions were written based on each best practice, each sub-process and considering the whole model. Thus, the questionnaire includes 64 questions. We also created a plan to distribute questions between experts. Each question was answered three times by different experts. Each expert answered 8 or 10 questions. At the end, we collected 192 answers to validate the FTS-SPM.

IV. INITIAL FINDINGS AND FUTURE WORK

In this section, we present our initial findings from the expert panel. As follows, we describe the next steps of this research.

A. Initial Findings

We analyzed 10 hours of recording interviews data. For each best practice, we have two questions and three answers for each question. As per the answers given by the experts, all best practices are analyzed and classified as VALID, PARTIALLY VALID, CONTEXT SPECIFIC or INCONCLUSIVE in the validation process.

- VALID: experts are in full agreement with the best practice.
- PARTIALLY VALID: experts are either fully or partially in agreement with the best practice.
- CONTEXT SPECIFIC: these best practices are applicable in a particular context.
- INCONCLUSIVE: experts are in disagreement with the best practice.

We identified 12 best practices (48%) as Valid, 11 best practices (44%) as Partially valid, and 2 best practices as Context specific (8%). We did not identify best practices as Inconclusive. Table 2 summarizes the results of the validation process for the FTS-SPM.

The results show that not all twenty-five best practices are perceived as high value practices for FTS projects. For those best practices classified as VALID, experts are in full agreement with its inclusion in the FTS-SPM. For those best practices classified as PARTIALLY VALID, we found a mix

of answers for the same question. For those best practices classified as CONTEXT SPECIFIC, a particular context is described by the experts.

TABLE II. INITIAL RESULTS

Sub-process (SP)	Best practice (BP) Nº	Best Practice (BP) title	Evaluation by Experts
SP01: Team Setup	BP30	At least one hour overlap between two sites	PARTIALLY VALID
	BP31	Fitting teams' working hours for a good overlap	VALID
	BP32	Teams distribution across two or three sites	PARTIALLY VALID
SP02: Project Planning	BP01	Use of agile methodologies for project management	PARTIALLY VALID
	BP02	Use of incremental software development approaches	VALID
	BP04	Application of FTS for testing and development phases	VALID
	BP36	Similar code patterns	PARTIALLY VALID
SP03: Communication Protocol	BP07	Daily exchange of the project status by technologies	VALID
	BP10	Use of screen sharing technology to exchange knowledge	VALID
	BP11	Calendar of handoff sessions should be clearly defined	PARTIALLY VALID
	BP12	Clean handoff and stocky handoff interactions	CONTEXT SPECIFIC
	BP13	Use of real time technologies for knowledge sharing	VALID
	BP15	Wikis and online forums to share knowledge between FTS teams	CONTEXT SPECIFIC
	BP21	Adopt proper technologies or tools to support communication between FTS teams	PARTIALLY VALID
	BP22	Time window	PARTIALLY VALID
	BP25	Corporate technologies for team interaction	VALID
	BP26	Models of emails and electronic messages	VALID
SP04: Cultural Training	BP33	Meetings between team members for building trust	VALID
	BP35	Cultural awareness training	PARTIALLY VALID
SP05: Task Allocation	BP17	CPro concept	PARTIALLY VALID
	BP18	Low task granularity	PARTIALLY VALID
	BP20	Task distribution by sequence or dependency	PARTIALLY VALID
SP06: Handoff Sessions	BP03	Daily stand-up meetings	VALID
	BP09	Daily handoff of 30 minutes duration with each development site	VALID
	BP14	Use of an FTP Sever (or data repository) to exchange code and documents	VALID

The opinion of experts about best practices took into consideration mainly its benefits for increasing the productivity and quality and at least two experts are in full agreement with a particular best practice.

Some experts have reported their own experience adopting a particular best practice in GSD projects. Thus, it will be possible to collect some recommendations to improve best practices supporting FTS characteristics.

It is interesting to observe that some best practices frequently adopted in GSD projects were identified as Partially valid, e.g. BP22 - Time window. For one expert, BP22 can be not applied to all cultures as a benefit. This is because sometimes team members don't make themselves available for a simple chat without prior scheduling.

The majority of the respondents from industry stated that their organizations have an interest in implementing FTS projects and in some situations the FTS concept is successfully applied. For example, an expert gave us the following answer about BP20 - Task distribution by sequence or dependency: "*When tasks cannot be divided and there are two people in different locations, then we do round-the-clock and follow the sun. Somebody starts and then another continues. There are situations where we work with Scrum, so they can choose the tasks, discuss issues, negotiate. And there are other situations where have a big task, then we do follow the sun. One starts, another continues and if the task is not finished yet, send back to the first one".*

These initial findings are interesting to observe that not all twenty-five best practices are perceived as high value practices for FTS projects. The use an expert panel to seek the opinion of experts about the FTS-SPM show perceived benefits more in some best practices than others.

B. Limitations and Future Work

This study has some limitations, which exist in any research project. In this study, we report the initial findings from a panel expert method. As with any research method, the author's bias is the main limitation. Thus, in order to reduce author's bias, at least two researchers reviewed the interpretation given to the collected data. In case of disagreement between reviewers and the main researcher (author 1), disagreements were reviewed until consensus was achieved.

It is also important to highlight the knowledge obtained from interviews (expert panel). This knowledge will contribute to get a better understanding about best practices contributing to make improvements in the FTS-SPM. Additionally, this knowledge will contribute to support decisions and future decisions about the next steps of this research.

Future steps are being planned in order to extend the classification approach for sub-processes and the sequence flow between sub-processes. Then, we will map recommendations given by experts in order to improve the model structure, as well the definition of best practices for

each sub-process. Future reports will encompass the complete FTS-SPM description, as well as the feedback from the practitioners.

V. CONCLUSION

In this paper, we report the preliminary results from an expert panel conducted with 20 experts to validate the FTS-SPM. We collected 10 hours of data recording interviews. Based on these data, we analyzed and classified all best practices. 48% of the best practices were classified as Valid, 44% as Partially valid, and 8% as Context specific. In contrast to previous works, our preliminary results show that not all twenty-five best practices are perceived as high value practices for FTS projects.

Our work has practical implications for organizations that want to implement FTS. Organizations will benefit by having access to the list of most relevant best practices for FTS, as well as a set of recommendations provided by experts to support FTS projects. This will provide opportunities for companies with the objective of achieving faster development succeed with FTS.

ACKNOWLEDGMENT

The first author is supported by the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91) and the second author is partially supported by the Science Foundation Ireland grant 10/CE/I1855 to Lero (www.lero.ie).

REFERENCES

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

[2] 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", 5th IEEE International Conference on Global Software Engineering (ICGSE), pp.165-174, 2010.

[3] 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.

[4] 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 Global Software Engineering Workshops (ICGSEW), pp. 18-23, 2013.

[5] J. Kroll, R. Prikladnicki, Audy, J. L. N., Carmel, E. and Fernandez, J. "A Feasibility Study of Follow-the-sun Software Development for GSD Projects". In: International Conference on Software Engineering (SEKE), 2013, Boston, USA.

[6] J. Kroll, I. Richardson, and J. L. N. Audy, "Proposing a Software Process Model for Follow the Sun Development", 26th International Conference on Software Engineering & Knowledge (SEKE), 2014, Vancouver, CA.

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

[8] 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", IEEE International Conference on Global Software Engineering (ICGSE '06), pp. 3-11, 2006.

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

[10] E. Hess and J. L. N. Audy, "FTSProc: A Process to Alleviate the Challenges of Projects that Use the Follow-the-Sun Strategy", IEEE Seventh International Conference on Global Software Engineering (ICGSE), pp. 56-64, 2012.

[11] 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, 2009.

[12] A. Taweel and P. Brereton, "Modelling software development across time zones", Inf. Softw. Technol. 48, 2006, pp. 1-11.

[13] 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.

[14] J. C. Tang, C. Zhao, X. Cao, and K. Inkpen, "Your time zone or mine?: A study of a global 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.

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

[16] 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 (ICGSE), 2009.

[17] 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, Vol. 18, pp.147-161, 2009.

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

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

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

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

[22] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams", International Conference on System Sciences, HICSS 2007. p. 274, 2007.

[23] S. Beecham, T. Hall, C. Britton, M. Cottee, and A. Rainer, "Using an expert panel to validate a requirements process improvement model", Journal of Systems and Software, Vol. 76, Issue 3, 2005, pp. 251-275.